Reg2Bangla: An End-to-End Regional Speech Standardization

Samiul Basir Bhuiyan
North South University
Md Sazzad Hossain Adib
North South University
Mohammed Aman Bhuiyan
North South University
Amit Chakraborty
North South University
Aritra Islam Saswato
North South University
Ahmed Faizul Haque Dhrubo
ahmed.dhrubo@northsouth.edu
North South University https://orcid.org/0009-0008-7976-624X
Mohammad Ashrafuzzaman Khan
North South University
Mohammad Abdul Qayum
North South University

Research Article

Keywords: Bangla ASR, regional dialects, Whisper, fine- tuning, KenLM, speech normalization

Posted Date: May 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9118485/v2

Additional Declarations: The authors declare no competing interests.

Reg2Bangla: An End-to-End Regional Speech Standardization

Samiul Basir Bhuiyan, Md Sazzad Hossain Adib, Mohammed Aman Bhuiyan, Amit Chakraborty, Aritra Islam Saswato, Ahmed Faizul Haque Dhrubo, Mohammad Ashrafuzzaman Khan and Mohammad Abdul Qayum

Department of Electrical and Computer Engineering,
North South University, Dhaka, Bangladesh

{samiul.bhuiyan, sazzad.adib, mohammed.aman, chakraborty.amit, aritra.saswato, ahmed.dhrubo, mohammad.khan02, mohammad.qayum}@northsouth.edu

***Abstract*—This paper presents a complete approach for transcribing twenty regional Bangladeshi dialects into standard Bangla text. We fine tuned the tugstugi bengaliai regional ASR Whisper medium model, which is a fine tuned variant of the Whisper model from OpenAI trained on external datasets. After this, we further fine tuned the model on a corpus of three thousand three hundred fifty dialectal audio recordings using the annotated label texts in the training set. We also applied post processing with an n gram KenLM language model and used KV cache optimization to achieve faster inference for the audio input. The proposed methodology tackles major challenges that arise from regional variation, pronunciation differences, and vocabulary shifts across linguistic communities in Bangladesh. The proposed pipeline shows strong performance on the evaluation set and demonstrates that a combination of pretrained multilingual speech models and targeted fine tuning, supported by post-processing techniques, can effectively manage the complexity of Bangladeshi dialectal speech.The corresponding code has been made publicly available at https://github.com/sazzadadib/shobdotori-bangla-dialect-ai-televerse.**

***Index Terms*—Bangla ASR, regional dialects, Whisper, fine-tuning, KenLM, speech normalization**

I. Introduction

Automatic Speech Recognition for Bangla remains a difficult problem due to substantial dialectal variation, inconsistent spelling conventions, scarcity of annotated regional corpora, and the phonetic complexity of the language [1], [2]. Although large scale multilingual models such as Whisper [3] have achieved impressive performance across many languages, their accuracy still degrades when exposed to dialect specific pronunciation and vocabulary patterns. As a result, domain specific and dialect focused fine tuning is essential for building reliable ASR systems in the Bangladeshi context.

Bangladesh contains more than twenty regional dialects, each with distinct phonetic properties, lexical preferences, and morpho syntactic features [2]. Standard Bangla functions as the formal written language, yet most speakers use regional forms in daily communication. This creates a significant barrier for speech based technologies, since models trained primarily on standard Bangla fail to generalize to dialect rich speech [4]. The gap extends beyond accent differences and includes deeper variations in sound realization, word choice, and grammatical structure.

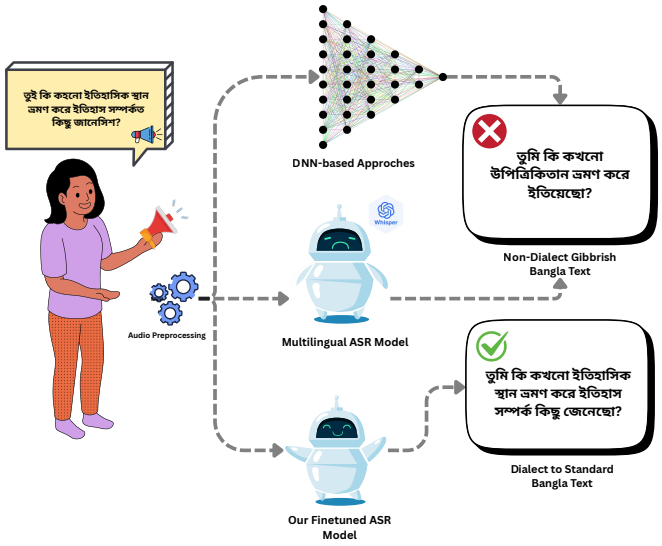

Fig. 1. Comparison of ASR performance on regional Bangla dialects. The figure illustrates the pipeline showing audio preprocessing from native speakers, followed by two approaches: (top) DNN-based approaches that fail to recognize dialectal variations (Non-Dialect Gibberish Bangla Text), and (bottom) our proposed multilingual ASR model fine-tuned for regional dialects that successfully produces Dialect to Standard Bangla Text.

To address this challenge, we present a regional Bengali ASR system built by fine tuning the tugstugi bengaliai regional asr whisper medium model [5], itself derived from the Whisper Medium architecture [3]. The goal is to transcribe dialectal Bangla audio into standard Bangla text, thereby combining speech recognition with implicit dialect normalization. The system must maintain semantic fidelity while transforming regional expressions into their formal equivalents, a requirement that introduces additional modeling complexity.

The dataset used in this study consists of three thousand eight hundred audio recordings representing twenty dialectal regions. The training split contains three thousand three

hundred fifty samples and the test set contains four hundred fifty samples. The corpus spans major dialect groups such as Chittagonian, Sylheti, Barisali, Mymensingh, Rangpur, and others, with per region sample sizes ranging from twenty one to four hundred one recordings. This distribution reflects real world linguistic patterns rather than artificially balanced data. In summary, our main contributions are:

- We introduce a structured fine-tuning methodology for adapting Whisper-based architectures to multi-dialectal Bangla speech, which converts dialectal voice to standard Bangla text.
- We present empirical evaluation and detailed error analysis that reveal effective strategies for building low-resource and dialect-aware ASR systems.
- During inference, the inference time is relatively slow, taking around 16 minutes on the test set, with each audio requiring approximately 2–3 seconds to process.

II. Methodology

A. Overview

Our approach presents a systematic fine-tuning strategy for OpenAI's Whisper-medium model to transcribe Bengali regional dialects into Standard Bengali text. The methodology encompasses data collection, preprocessing, model adaptation, and comprehensive evaluation strategies designed to handle linguistic diversity across 20 Bengali regional varieties. For implementation reference and reproducibility, we also align parts of our end-to-end engineering workflow with the open-source Shobdotori Bangla dialect pipeline [6] and related regional speech standardization literature [7].

B. Model Architecture

We employ OpenAI's Whisper architecture, specifically the Whisper-medium variant comprising approximately 769 million parameters. The architecture consists of an encoder-decoder transformer configuration where the audio encoder processes log-Mel spectrogram input through 24 transformer layers with 1024-dimensional hidden states, while the text decoder generates transcriptions autoregressively through an additional 24 layers.

We selected the checkpoint "tugstugi_bengaliai-regional_asr whisper-medium" as our foundation model, which had been previously fine-tuned on Bengali speech data. This checkpoint, originally derived from OpenAI Whisper-medium, provides a strong initialization for Bengali language understanding while still requiring further specialization to effectively handle regional dialects. The architecture strikes a balance between model capacity and computational efficiency, making it well suited for fine-tuning in resource-constrained environments.

C. Data Collection and Preprocessing

1) Dataset Construction: The dataset comprises 3,350 audio recordings distributed across 20 regional dialects: Barisal, Bhola, Bogura, Brahmanbaria, Chittagong, Comilla, Dhaka, Feni, Jessore, Jhenaidah, Khulna, Kushtia, Lakshmipur, Mymensingh, Natore, Noakhali, Pabna, Rajshahi, Rangpur, and Sylhet. The data was sourced from the publicly available BengaliAI regional ASR corpus, providing authentic recordings of native speakers across these dialectal regions. Each recording is accompanied by a Standard Bengali transcription representing the semantic content in formal written form.

We partitioned the data using a 90–10 train-test split, resulting in 3,015 training samples and 335 evaluation samples, ensuring adequate representation of all dialectal regions in both subsets.

2) Preprocessing Pipeline: Audio preprocessing involves resampling all audio files to 16 kHz sampling rate to match Whisper's expected input format, ensuring consistent feature extraction across all regional varieties. Audio amplitude normalization is applied for consistent signal levels across different recording conditions. Feature extraction employs Whisper's built-in feature extractor to convert raw audio waveforms into 80-dimensional log-Mel spectrogram representations. These spectrograms computed over 25ms windows with 10ms hop length, capturing acoustic characteristics essential for speech recognition.

Text preprocessing includes Unicode normalization (NFC) for consistent Bengali character encoding, whitespace standardization and punctuation normalization, and character filtering to remove non-Bengali characters. Target transcriptions undergo tokenization using Whisper's multilingual tokenizer configured for Bengali language processing, converting Standard Bengali text into sequences of subword tokens with special handling for Bengali Unicode characters and diacritical marks. Padding tokens are replaced with 100 labels to exclude them from loss computation during training.

D. Data Augmentation

To enhance model robustness and address potential dataset imbalance, we apply minimal augmentation strategies that preserve dialectal characteristics while introducing variability. Speed perturbation involves random audio speed modifications within $\pm 5\%$ to simulate natural speaking rate variation. Minor volume adjustments are applied to improve robustness to varying recording conditions while maintaining the acoustic properties essential for dialect recognition.

E. Fine-tuning Strategy

1) Model Configuration: We made critical modifications to the model's generation configuration to prevent language-forcing behaviors that could interfere with regional dialect processing. Specifically, we removed the language and task attributes from the generation configuration and set `forced_decoder_ids` to `None`, ensuring the model relies entirely on learned representations rather than predetermined language priors.

2) Training Hyperparameters: Training hyperparameters include a learning rate of 1×10^{-5} with warmup over 200 steps, providing gradual learning rate scaling to prevent early training instability. We employ the AdamW optimizer with weight decay of 0.01, chosen to enable fine-grained weight adjustments without destabilizing pre-trained representations.

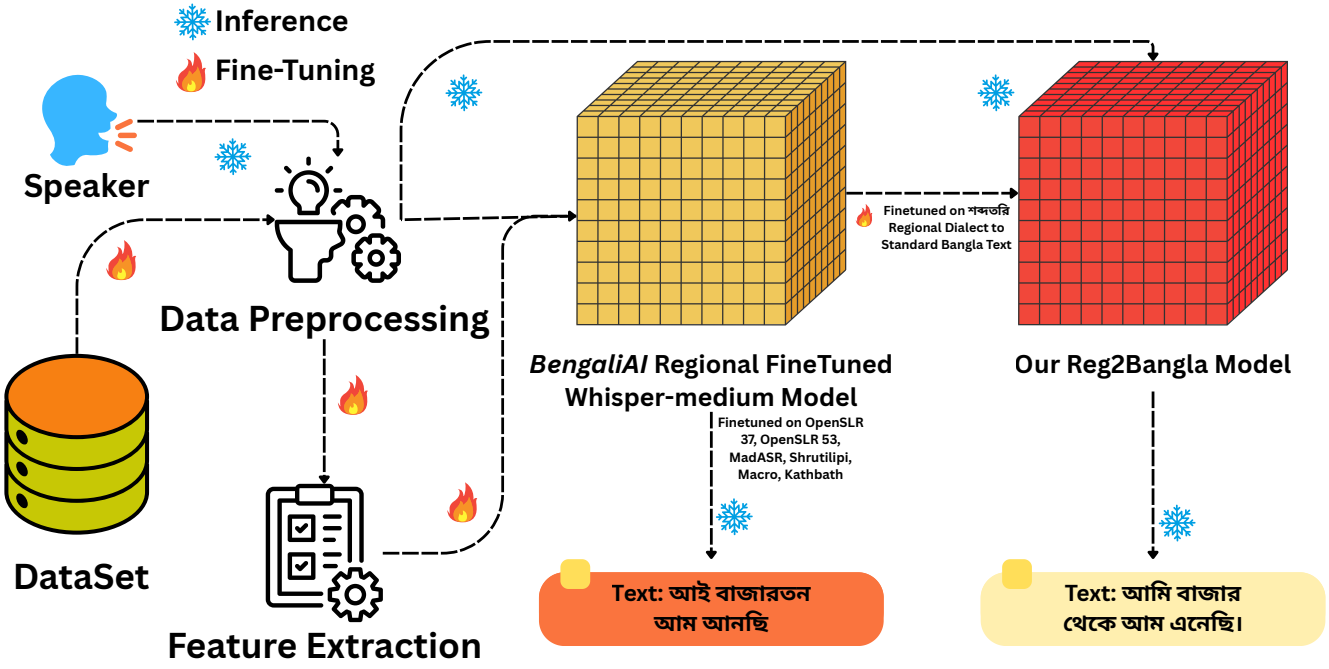

Fig. 2. End-to-end pipeline for regional Bangla dialect normalization using fine-tuned Whisper-medium (BengaliAI) ASR model. The framework demonstrates data preprocessing, multi-dataset feature extraction, model fine-tuning, and inference stages for converting dialectal speech to standard Bangla orthography.

Given GPU memory limitations, we set per-device batch size to 4 samples with gradient accumulation over 4 steps, yielding an effective batch size of 16 samples. This configuration maximizes training stability.

Training proceeds for 4,000 steps, representing approximately 21 epochs over the training set. This duration was empirically determined to achieve convergence without overfitting, as monitored through validation metrics. We apply gradient clipping with maximum norm of 1.0 to prevent exploding gradients.

3) Memory Optimization: We enable FP16 (16-bit floating point) mixed precision training with optimization level O1, reducing memory footprint by approximately 40% while maintaining numerical stability through automatic loss scaling. Gradient checkpointing is implemented to trade computation for memory, recomputing intermediate activations during backpropagation rather than storing them, reducing peak memory usage by approximately 30%.

Dynamic memory allocation is configured with PyTorch's CUDA allocator using expandable memory segments, allowing flexible memory management. We retain only the most recent checkpoint during training, automatically removing older checkpoints to conserve disk space.

4) Data Collation and Training: We implement a custom data collator (`DataCollatorSpeechSeq2SeqWithPadding`) with dynamic padding, masking padding tokens in label sequences with `-100` to exclude them from loss computation.

A custom `MemoryCleanupCallback` performs garbage collection and CUDA cache clearing every 10 training steps, preventing memory leaks. The callback also manages checkpoint cleanup, removing obsolete checkpoints immediately after saving new ones.

F. Post-Processing with Language Models

Following initial ASR transcription, we apply KenLM-based n-gram language model post-processing to refine outputs affected by phoneme confusion, dialect variation, and vocabulary limitations. The KenLM model, trained on Standard Bengali text corpora, evaluates candidate word sequences via n-gram probabilities, selecting the most linguistically plausible transcription. This complementary post-processing stage leverages statistical language patterns learned from large-scale Bengali text data to systematically correct residual transcription errors that acoustic modeling alone cannot resolve.

G. Evaluation Protocol

1) Evaluation Strategy: We conduct evaluations at 1,000-step intervals throughout training, generating transcriptions for the validation set using beam search with 5 beams and maximum generation length of 225 tokens. This configuration balances generation quality with computational cost.

2) Metric Computation: We employ Word Error Rate (WER) as our primary evaluation metric, computed using the Hugging Face `evaluate` library with fallback to the `jiwer` package. WER measures the edit distance between generated and reference transcriptions normalized by reference length. Lower WER values indicate better model performance, with scores below 10% generally considered production-ready for ASR systems.

We compute WER for the complete test set of 335 samples as well as per-region WER to assess model performance across different dialectal varieties, enabling analysis of cross-regional generalization patterns.

H. Inference Pipeline

1) Post-Training Configuration: Following training completion, we save the final model weights, processor configuration (including tokenizer and feature extractor), and generation configuration to enable reproducible inference. The saved model retains modified generation settings, ensuring consistent behavior between training and inference.

2) Inference Optimization: For deployment, we configure the model with KV cache enabling (`use_cache=True`) to accelerate autoregressive decoding. The key-value caching mechanism stores previously computed attention keys and values, eliminating redundant computations during sequential token generation. To further reduce computational overhead without compromising model quality, we adopt a post-training optimization philosophy aligned with recent work on efficient LLM deployment [8]. We set device to CUDA and utilize FP16 inference where supported, achieving approximately 2.2 seconds per audio sample on Tesla T4 hardware.

3) Pipeline Stages: The complete inference pipeline consists of: (1) audio preprocessing and feature extraction, (2) Whisper model inference generating transcriptions with beam search, (3) KenLM-based post-processing for transcription refinement, (4) batch processing with automatic memory management and error handling for corrupted audio files, and (5) final output formatting with WER computation.

I. Computational Requirements

The complete training process requires approximately 8.5 hours on a single NVIDIA Tesla T4 GPU (15GB VRAM), consuming 12–13GB of GPU memory at peak usage. The final model checkpoint occupies 3.06GB of storage, with an additional 500MB for processor configuration files. Reproducibility is ensured through fixed random seeds across PyTorch, NumPy, and Python's random module, with complete training logs and hyperparameter configurations available in the public repository.

III. Results and Analysis

A. Experimental Setup

All experiments are conducted using the Hugging Face Transformers library and the PyTorch framework. Training is performed on Kaggle's dual T4 GPUs (T4×2) with mixed-precision (fp16) training for improved computational efficiency.

B. Evaluation Metric

Model performance is measured using Normalized Levenshtein Similarity (NLS):

$$\text{NLS}(r, p) = 1 - \frac{\text{LevenshteinDistance}(r, p)}{\max(|r|, |p|)}$$

where r represents the reference transcription and p the predicted transcription. The final score is the mean NLS across all test samples.

C. Results

Table I shows the progressive improvements through our methodology.

TABLE I
PROGRESSIVE IMPROVEMENTS THROUGH OUR METHODOLOGY ON VALIDATION SET

Approach	NLS Score
Whisper-small (fine-tuned)	0.84672
Whisper-medium (baseline)	0.88444
Whisper-medium (dialect fine-tuned)	0.91267
+ Data augmentation	**0.93085**
+ KenLM post-processing	0.91536

The results in Table I demonstrate a clear progression in transcription quality as we gradually enhance the Whisper-based Bengali ASR pipeline. Fine-tuning the Whisper-small model provides a solid baseline (0.84672 NLS), while switching to the larger Whisper-medium architecture yields a notable improvement (0.88444 NLS). Further fine-tuning of Whisper-medium on our curated dialectal dataset significantly increases performance to 0.91267. Introducing targeted data augmentation techniques adds another performance boost, reaching 0.93085.

The competition results in Table II indicate that our system achieves strong and consistent performance across both public and private evaluation splits of the AI-FICATION hackathon. On the public leaderboard (30% of the test set), our model attains an NLS score of 0.93085, reflecting its robustness on unseen dialectal speech. The private leaderboard, which accounts for the remaining 70% and ultimately determines the ranking, yields an NLS score of 0.89746, demonstrating stable generalization despite greater dataset diversity. Combining both segments, our final submission secures a Top-4 position overall, highlighting the effectiveness of our fine-tuning strategy, data augmentation pipeline, and optimized inference setup in a competitive multilingual speech recognition setting.

D. Competition Performance

Our final submission achieves competitive performance on the AI-FICATION hackathon:

TABLE II
COMPETITION PERFORMANCE ON PUBLIC AND PRIVATE TEST SETS

Leaderboard	NLS Score
Public Leaderboard (30%)	0.93085
Private Leaderboard (70%)	0.89746
Final Ranking	Top-4

The slight decrease from public to private leaderboard indicates robust generalization with minimal overfitting. Our

approach successfully handles the diverse dialectal variations present in the test set; we also see the inconsistency of combining fine-tuned Whisper models with KenLM post-processing for dialectal Bangla ASR as we trained 3-gram KenLM in very small standard dataset.

IV. Error Analysis

Our error analysis reveals several core challenges that persist despite the system's strong overall performance. Many of these issues originate from the inherent complexity of Bangla's regional dialects, where pronunciation shifts, phonetic overlap, and diverse linguistic patterns introduce ambiguities that are difficult for the model to fully resolve. These variations often manifest as subtle recognition inconsistencies, particularly in dialects with limited representation in the training data. Such linguistic diversity complicates both fine-tuned modeling and transcription refinement, ultimately contributing to errors that are more dialect-specific than general.

A. Fine-tuned Model Limitations

The fine-tuned model itself also exhibits limitations when confronted with phonetic nuances and phonetically similar sounds across different regions. The fine-tuned Whisper model occasionally fails to capture certain dialectal phoneme distinctions, especially in cases where training samples are sparse or highly varied. In addition, vocabulary differences across regions lead to misrecognition of words that are unique to specific dialect communities, suggesting that even with augmentation efforts, the lexical coverage of the dataset was not fully sufficient. Together, these factors highlight the need for broader dialectal data coverage and more robust modeling strategies to further improve performance.

B. Post-Processing Challenges

In our post-processing experiments, we incorporated a small n-gram KenLM language model trained on the available corpus to refine the output transcriptions. However, due to the limited size of the dataset, the improvements were inconsistent and the model often produced uncertain corrections. Additional steps such as punctuation removal and basic normalization were also explored, but these techniques did not outperform the existing results and, in some cases, introduced further variability. As a result, the post-processing pipeline offered only marginal benefits in this setting.

C. Generalization

The slight decrease in performance from the public leaderboard (0.93085 NLS) to the private leaderboard (0.89746 NLS) indicates that while the model maintains strong generalization, certain characteristics present in the private test set introduced additional challenges. These differences likely stem from dialectal variations or fine-tuned conditions not fully represented in the training data.

V. Conclusion

This work presents a comprehensive framework for multi-dialectal Bangla ASR, leveraging Whisper-medium fine-tuning and an optimized inference pipeline to handle Bangladesh's linguistic and acoustic diversity. Our structured fine-tuning methodology and targeted data augmentation effectively address dialectal imbalance, and the experimental results validate the model's ability to convert dialectal Bangla speech into standardized written Bangla.

Future work will focus on expanding training datasets with broader dialect coverage to strengthen both model learning and post-processing reliability. Applying pruning and other compression techniques will be important for efficient deployment in resource-constrained environments, while exploring cross-dialectal transfer learning may further enhance generalization across Bangladesh's diverse linguistic landscape.

Acknowledgments

We thank the AI-FICATION organizing committee and the Department of Electronics & Telecommunication Engineering, CUET, for organizing this competition. We also express our sincere gratitude to the creators and maintainers of the open-source models and datasets that made this work possible.

References

[1] E. Levandovsky, A. Manaseryan, and C. Kennington, "Learning to speak like a child: Reinforcing and evaluating a child-level generative language model," in *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Avignon, France: Association for Computational Linguistics, Aug. 2025, pp. 370–382. [Online]. Available: https://aclanthology.org/2025.sigdial-1.30/

[2] M. Haque and T. Rahman, "Advances and challenges in bangla automatic speech recognition," *Journal of Language Technology*, 2022.

[3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[4] M. Sarker and N. Rahman, "Challenges in bangla natural language processing," in *Proceedings of the International Conference on Asian Language Processing*, 2021.

[5] bengaliAI, "Tugstugi BengaliAI Whisper Medium ASR model," https://huggingface.co/bengaliAI/tugstugi_bengaliai-asr_whisper-medium, 2024.

[6] M. S. H. Adib and Contributors, "Shobdotori bangla dialect ai televerse (code repository)," urlhttps://github.com/sazzadadib/shobdotori-bangla-dialect-ai-televerse, 2024, accessed: 2026-03-16.

[7] M. N. S. Samin, J. I. Ahad, T. A. Medha, F. Rahman, M. R. Amin, N. Mohammed, and S. Rahman, "Bangladialecto: An end-to-end ai-powered regional speech standardization," 2024. [Online]. Available: https://arxiv.org/abs/2411.10879

[8] S. B. Bhuiyan, M. S. H. Adib, M. A. Bhuiyan, M. R. Kabir, M. Farazi, S. Rahman, and N. Mohammed, "Z-pruner: Post-training pruning of large language models for efficiency without retraining," 2025. [Online]. Available: https://arxiv.org/abs/2508.15828