Team Vocab Vanguard

Bhashika: Dialect Aware TTS Model for Indic Languages

Presented by Sarthak Sharma, Drishti Singh, Abhishek Dutt, N. Deepika, Aindrila Majumdar

Abstract

We propose Bhashika, a dialect-aware TTS model for Indic languages and regional accents. Unlike standard models, which lack phonetic considerations for dialect-based speech generation, Bhashika addresses this gap. It utilizes the Vaani dataset from IISC Bangalore and ARTPARK, which includes various dialects and code-mixing of Hindi and English. The pre trained model is finetuned and then improves upon previous approaches by adopting a multi-encoder, single-decoder architecture. Evaluation will be conducted using MOS and MCD scores to assess its naturalness and accuracy.

Overview

01 Introduction

02 Related Work

03 Novelty

04 Implementation

05 Architecture Proposal

06 Results

Introduction

We are building a dialect-aware Text-to-Speech (TTS) system for Indic languages that generates natural-sounding speech, reflecting regional pronunciations, intonation, and prosody.

Deep Learning in TTS Model
Advancements in deep learning have revolutionized text-to-speech (TTS) models, moving beyond traditional methods like formant and concatenative synthesis. Neural TTS models generate natural, human-like speech with minimal manual feature engineering, making them ideal for diverse real-world applications.

Motivation
India's linguistic diversity presents unique challenges in TTS development. Dialects and prosody vary significantly across regions, but current models rely heavily on manual annotation and custom phoneme dictionaries, limiting scalability.

What do we want to accomplish from this proposed idea?
Our goal is to build a dialect-aware TTS system for Indic languages that captures regional pronunciation, intonation, and prosody variations, enabling scalable speech synthesis without the need for manual annotations. The system will multiple dialects and Hinglish code-switching, ensuring it preserves regional linguistic identity in synthetic speech.

Expected Outcomes

1. Pronunciation Variations: Adapts to dialect-specific pronunciations (e.g., खाना as "khana" or "kana").
2. Intonation & Stress: Reflects unique patterns (e.g., {म vs. मैं usage).
3. Hinglish Adaptation: Retains code-switching nuances (e.g., "blue" remains ब्लू)
4. Tone & Pitch: Captures regional tonal variations (e.g., rising intonation in Delhi Hindi for “तुम कैसे हो?”).

Related Work

IndicTTS-based Models:

Research on TTS for Indic languages is limited, but advancements like deep learning-based systems for 13 Dravidian and Indo-Aryan languages have been made, with FastPitch and HiFi-GAN V1 showing superior results.

Dialect-based TTS Works

Several studies focus on dialectal speech generation, such as fine-tuning TTS models for Nepali and Arabic dialects using techniques like FastPitch and dialect tokens to improve accuracy.

Multi-Speaker TTS Models

Models like LeanSpeech and multi-speaker TTS systems focus on enhancing voice cloning for Indian languages, with techniques like few-shot and zero-shot learning improving speaker adaptation.

Challenges in Dialectal Speech

Research on dialectal speech systems, like Arabic and Swiss German, highlights the importance of incorporating dialect-specific features and tokens to improve model performance.

Novelty

01

Dedicated TTS models for dialects are scarce, with most models designed for general tasks and adapted for downstream tasks via prompting or fine-tuning. Our work is novel in evaluating pre-trained models (PTMs) specifically for dialect-based speech generation, an area with limited exploration.

02

We are using the Vaani Dataset for evaluating the models, which has a lot of data pertaining to dialects. This dataset is quite unexplored in research right now and therefore, our novelty lies in the dataset as well.

03

We propose a novel architecture for a dialect-aware TTS model, addressing the gap in generating dialect-based speech using multiple phonetic features. Existing models like AI4Bharat IndicTTS focus on predicting words in a sequence and accents but fail to capture true dialectal nuances and how words are spoken in context.

Implementation – Dataset

Project Vaani
Source:
Curated by IISc Bangalore and ARTPARK, funded by Google

Scope:
150,000+ hours of audio from 59 languages across 12 Indian districts; 14,434 hours open-sourced.

Collection Method:
Real-life speech, code-mixed Hinglish, recorded by locals describing images.

Bihar Dataset:
- 23 lakh audio files from 20 districts.
- 2,77,453 Hindi audio files (~6 seconds each).

Diversity Factors
- Linguistic: Multiple languages/dialects.
- Urban-Rural: Balanced representation.
- Age & Gender: Broad inclusivity.

Implementation

Zero shot Prompting

We tested AI4Bharat IndicTTS and Toucan TTS models using zero-shot prompting to assess their ability to generate dialect-specific speech without prior examples. Both models failed to capture dialect nuances. AI4Bharat IndicTTS produced robotic Hindi with unclear accents, while Toucan TTS synthesized Hindi with an English-like accent due to its multilingual training. Neither model reflected the natural way dialects are spoken.

Few Shot Prompting

We applied few-shot prompting on the Toucan TTS model, leveraging its ability to capture phonetics and tone. By providing example audio files as speaker references, the model mimicked the accent and tone of the samples, a feature designed for multi-speaker speech generation. However, this approach resulted in exact voice mimicry, which deviates from our goal of studying dialect-specific speech generation.

Implementation

Transcription

To train our TTS model on <audio, transcription> pairs, we generated transcriptions for the Vaani dataset using the AI4Bharat IndicConformer ASR model, which supports accurate speech-to-text conversion in 22 Indian languages. However, the model does not generate phoneme-level transcriptions, posing a challenge in training the TTS model to incorporate phonetic nuances effectively.

Fine tuning

To address the limitations of prompting techniques, we fine-tuned the Toucan TTS model on Bengali and Telugu datasets. This allowed the model to learn speech phonetic patterns instead of merely mimicking audio characteristics. Dataset functions mapped audio files to transcripts, while preprocessing incorporated language and accent identifiers. The fine-tuning pipeline utilized the resume_checkpoint argument for seamless continuation from pre-trained checkpoints, enabling effective adaptation for dialect-specific speech generation.

Results of Fine Tuning

Evaluation Metrics

- MCD: Measures spectral quality (lower is better).
- MOS: Combines PESQ (speech quality) and STOI (intelligibility).
- GOP: Evaluates pronunciation accuracy with Posterior, Likelihood, and Likelihood Ratio

Dataset	MCD (Plain)	MCD (DTW)	MCD (DTW_SL)
Bengali	12.82	5.33	5.94
Telugu	10.82	4.12	4.94

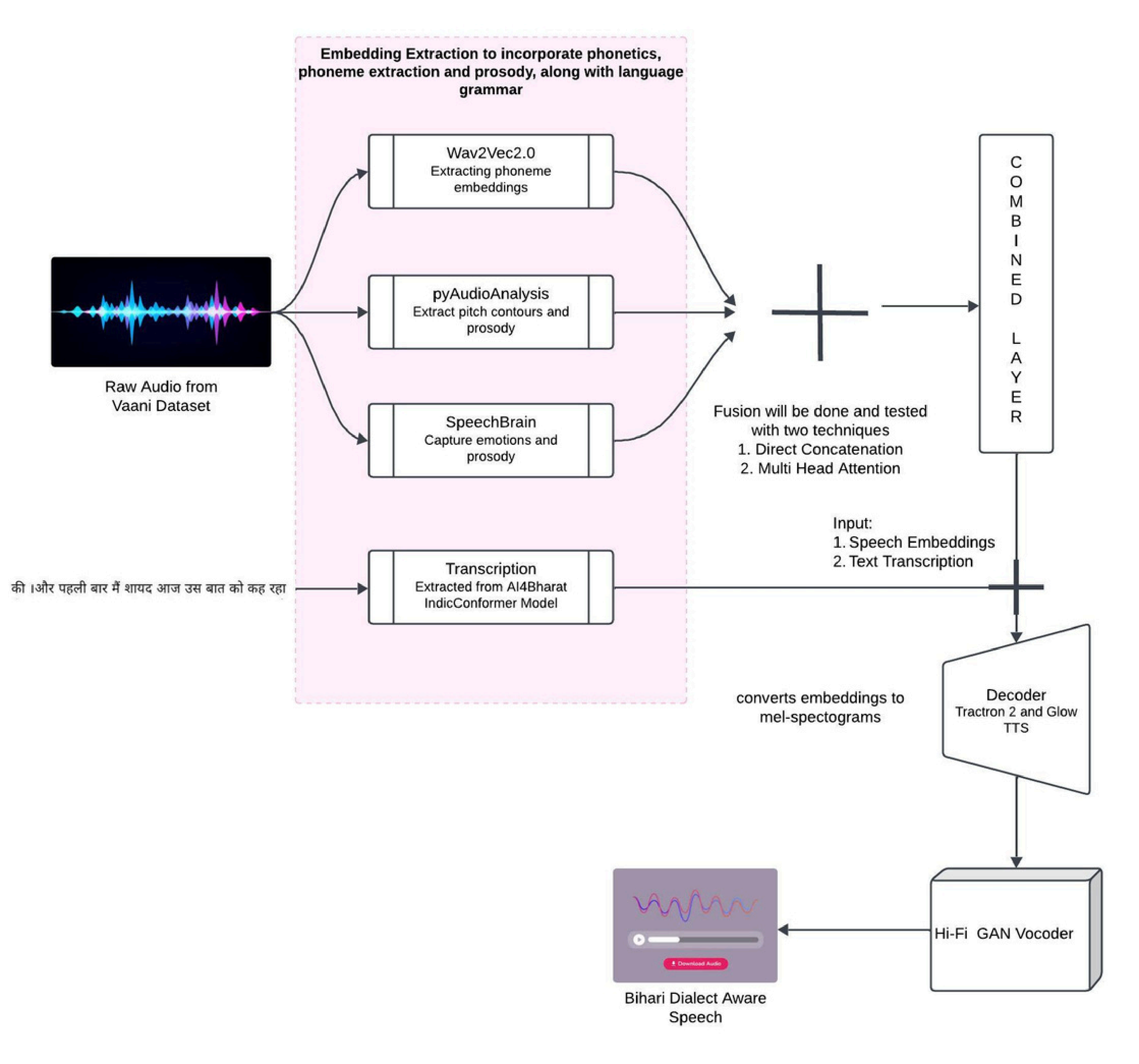
Metric	Bengali	Telugu
PESQ	1.0307	1.0302
STOI	0.1438	-0.037
Est. MOS	0.8748	0.4208

GOP Metric	Bengali	Telugu
Posterior	-15.4751	-17.035
Likelihood	0.03125	0.03125
Likelihood Ratio	0.0031	0.0058

Observations

- Fine-tuning improved spectral quality, especially in DTW modes.
- Bengali outperformed Telugu in speech quality, intelligibility, and pronunciation accuracy.
- As compared to PTMs and our expected goals, improvement is required in both the languages

New Architecture Proposal



Embedding Extraction to incorporate phonetics, phoneme extraction and prosody, along with language grammar

Raw Audio from Vaani Dataset

Wav2Vec2.0
Extracting phoneme embeddings

pyAudioAnalysis
Extract pitch contours and prosody

SpeechBrain
Capture emotions and prosody

Fusion will be done and tested with two techniques
1. Direct Concatenation
2. Multi Head Attention

COMBINED LAYER

की ।और पहली बार मैं शायद आज उस बात को कह रहा

Transcription
Extracted from AI4Bharat IndicConformer Model

Input:
1. Speech Embeddings
2. Text Transcription

converts embeddings to mel-spectograms

Decoder
Tractron 2 and Glow TTS

Hi-Fi GAN Vocoder

Bihari Dialect Aware Speech

We propose a novel multi-encoder single-decoder TTS architecture dedicated to dialect-based speech generation. Unlike existing models requiring phoneme sequences, our approach eliminates the need for custom phoneme dictionaries, reducing reliance on linguists. The raw audio from the Vaani dataset undergoes embeddings for phonemes (via Wav2Vec 2.0), prosody/pitch (via pyAudioAnalysis), and emotional nuances (via SpeechBrain). These embeddings are fused and decoded into mel-spectrograms, converted to audio waveforms using HiFi-GAN. This architecture aims to capture dialectal patterns effectively, addressing phonetic, tonal, and emotional nuances.

Thank You