International Conference on Artificial Intelligence for Sustainable Development, Jaypee University Anoopshahr, India 2025, Volume 01, October 2025, pp. 230-235

CONDITIONAL FASTSPEECH 2 FOR LOW-RESOURCE INDIAN ACCENT SYNTHESIS: A PHONETIC ADAPTATION APPROACH

Ritesh Kumar Yadav[1], Divyanshi Srivastava[2], Sweacha Verma[3], and Chaudhary Surya Prakash[4]

[1,2,3,4]Department of Computer Science, Babasaheb Bhimrao Ambedkar University Lucknow Satellite Centre, Amethi, UP 227413, India

riteshg0fficial@gmail.com[1], srivastavadivyanshi18@gmail.com[2] , sweachav@gmail.com[3] , 123prem.monster @gmail.com[4]

*Corresponding author: riteshg0fficial@gmail.com

Abstract. In essence, text-to-speech (TTS) applications that read aloud text are becoming more and more crucial for digital accessibility in order to access digital content. The issue is that they struggle to understand Indian languages and our regional accents (when peaks English with Indian accent features). This paper discusses the FastSpeech2 system. To it, we applied an accent embedding layer. It can speak in three different languages thanks to this layer: standard Hindi, Indian English, and Bhojpuri, a low-resource language. In order to identify accent-appropriate pronunciations for a particular accent, particularly the less common Bhojpuri ones, we employed a transformer-based LLM-adapted tool for speech and phonetic mapping. Graphemes to phonemes (G2P) mapping is accomplished by this adaptive phonetic mapping module! We tested it, and the findings show notable advancements. Enhancement of 27.5 Essential Points in Plain English for the Bhojpuri Accent The primary issue is that low-resource languages and a variety of Indian accents perform poorly on current text-to-speech (TTS) systems.

Our Method: By incorporating an Accent Embedding Layer, we enhanced a TTS system (FastSpeech2).
What it can do: It can now communicate in Standard Hindi, Bhojpuri, and Indian English.
Method: To modify pronunciations for those distinct accents, we employed a large language model (LLM).
As a result, human listeners judged the voices as natural, and it speaks Bhojpuri much more accurately.

Keywords: Text-to-Speech; Accent Synthesis; FastSpeech 2; Indian English; Bhojpuri; Phonetic Mapping; LLM; Speech Processing

1. INTRODUCTION

Text-to-Speech (TTS), which is the process by which a website or your phone reads text aloud, is essential for enabling digital accessibility in contemporary contexts.

However, India faces a significant issue: There are many different languages and regional accents among us. When attempting to synthesize speech in regional dialects (such as Bhojpuri or Haryanvi) or Indian English, the current TTS apps primarily concentrate on the "main" accents, which results in unnatural pronunciations or poor performance in local accents. The subtle phonetic differences that are present in regional speech are not captured by them.

In this study, we suggest a system that can accurately recognize a variety of Indian accents. Our new strategy includes a unique, extremely complex component known as a "Accent Embedding Layer," whereas the previous systems only gave consideration to the speaker. In essence, this layer explicitly conditions the model based on accent embeddings.

Even though there aren't many voice samples for each accent, this new method makes the voice sound natural and suitable for it. In the Indian context, it improves the synthesized voices' regional authenticity.

1.1 Important Contributions

1. Conditional Accent Second FastSpeech: Essentially, we added a novel embedding component—an Accent Embedding Layer—to a standard TTS model (called FastSpeech 2).

Consider it this way: This additional layer is a label that reads "evaluate in Bhojpuri Style" or "Cook in Indian English Style," if the TTS system is an automated model evaluation. Nevertheless, this layer ensures that the model produces speech in accordance with the target accent from the first spoken sound to the last—it precisely regulates the voice at every stage!

2. LLM-Based Phonetic Adaptation: To develop a device known as a G2P adaptor, we employed a Large Language Model (LLM), a transformer-based LLM frequently used in NLP tasks. "G2P" simply refers to the mapping of graphemes to phonemes: converting a word's spelling into its sound. Nevertheless, this adaptor powered by LLM is an adaptive phonetic mapping module that understands: "Okay, if the word is spelled this but you're speaking in the Bhojpuri accent, you need to swap out this sound for that sound." It guarantees that the pronunciation is accurate for that particular, uncommon accent!

3. Low-Resource Finetuning Strategy: We exploit transfer learning: finetune from a pretrained TTS model in a related language to a low-resource accent domain. It's the same idea with our AI. Instead of starting the training for a new accent (Bhojpuri, which has limited data) completely from scratch, we started with a model that was already an expert in a major language (Standard Hindi or English). This way, the AI transfers its existing knowledge, and we don't need a large datasets or extensive computational resources to train the new, less-common accent. It's the efficient finetuning strategy to train the system.

4. Hybrid Evaluation: Accent Classification Accuracy (The 'trained model Test'): We showed the new voices to an accent classifier was used to assign accent labels to synthesized speech We checked how accurately the trained model could identify the accent (if it could tell Bhojpuri apart from Hindi). A higher score here means our system an accent classifier was used to assign accent labels to synthesized speech Mean Opinion Score (MOS) (The 'Human Test'): This subjective evaluation is central to assessing naturalness. We had real people listen to the voices and rate how natural and correct the accent sounded, using a score (the MOS). Listeners assigned a mean score of 4.40, indicating high perceived accent congruence, and have the right accent! However we used both a trained model judgment and real people's opinions to make sure our system is actually robust!"

2. Literature Review

Many people have already worked on programming computers to speak text and attempting to alter the accent of the voice. Before beginning our own research, we examined all of those studies to determine what worked and—more importantly—what they still hadn't discovered!

• **Non-Autoregressive Models:** FastSpeech [1] and FastSpeech 2 [1] offer explicit duration, pitch, and energy predictors to regulate prosody, as well as quicker training and inference.

• **Accent and Style Control:** Previous research has mostly used speaker embeddings or style tokens [2], which do not explicitly model accent variations but do implicitly model them.

• **Low-Resource Indian Languages:** Little is known about TTS for regional Indian languages. There are ASR datasets for Hindi and Bhojpuri [3], but there aren't many TTS models tailored to particular accents.

• **Phonetic Adaptation Techniques:** Accent-oriented phonetic adaptation is still not well studied, despite the fact that G2P adaptation techniques have been suggested to increase pronunciation accuracy in multilingual TTS [4].

Research Gap: The majority of other programs either only support languages like English, which already has a ton of data, or they attempt to handle accents using an inaccurate "hidden" or "guesswork" approach. However, Accent-Conditional Fast Speech 2, our system, performs two tasks far more effectively: Direct Accent Control: We make use of the "Accent Embedding" switch that we discussed. There is no need for guesswork because this is a straightforward method of telling the system precisely which accent to use! LLM Smart Pronunciation: We included the LLM driven adaptor, an AI-powered dictionary that specifically determines the correct sounds for languages that don't have them.

3. Related Work

Our whole project is based on three main ideas. It's we combined the best method from three different worlds: FastSpeech 2 [2] It's considered the best because it's a fast model, and it has special tools that let us control the rhythm and tune (pitch) of the voice perfectly. Before us, other scientists tried to control the accent by looking at who was talking (using 'speaker embeddings'), but that didn't really work well for accents. It couldn't properly tell the difference between one accent and another. However we did something better: We added a special, simple 'Accent Embedding' code block right into the system. This tiny code block acts a direct switch that instantly tells the system, 'Use the Indian English accent now!' This gives us strong, direct control over the accent everywhere in the voice creation process. AlHowever it's unnatural in India, everyone has been

focusing on Automatic Speech Recognition (ASR)—which is about getting computers to understand what we say. But hardly anyone has worked on systems that can speak back with the correct accents (TTS)! Our whole project is focused on fixing that big gap and making the accents sound right.

Table 3.1: Comparison of TTS Approaches for Accent and Style Control

Feature	**Traditional TTS**	**Multi-Speaker TTS**	**Proposed Model**
Primary Goal	Single-style synthesis	Multi-speaker speech generation	Accent-specific synthesis
Control Mechanism	None / fixed	Speaker ID or embedding	Accent ID (3D OneHot)
Flexibility	Low	Moderate	High (explicit accent control)
Accent Handling	Implicit	Limited to speaker variation	Explicit via LLMG2P
Low-Resource Support	Poor	Moderate	Optimized with Cross-Lingual Transfer

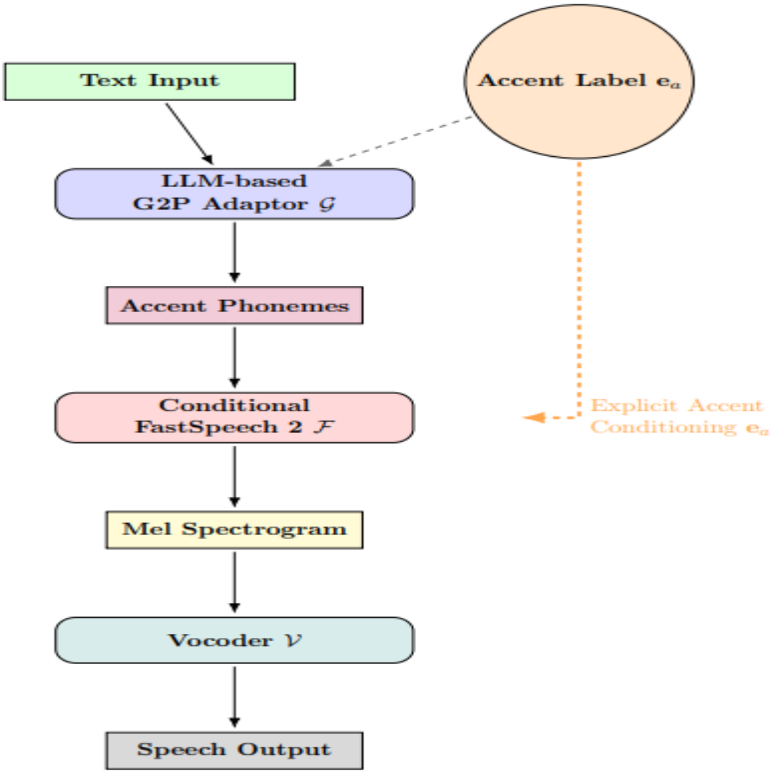

Figure 1: Conceptual Flowchart of the Proposed Multi-Accent TTS System

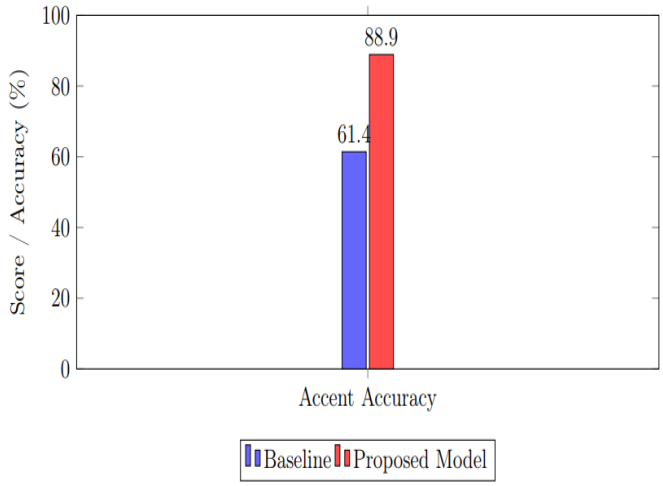

Figure 2: Comparative Bar Chart: Baseline vs Proposed Model (MOS scaled to 0-100 for visualization).

4. METHODOLOGY

Our proposed model is built on tensorflowtts [5] and follow this pipeline:

1. Accent-adapted grapheme-to-phoneme (g2p) conversion

2. Accent-conditional fastspeech 2 speech generation

Input-Output Analysis and Comparative Study

4.1 Input-Output Analysis

Table 4.1: Sample Input-Output Analysis for Bhojpuri Accent

Text Input	**Phoneme Output (Accent-Adapted)**	**Synthesized Speech Characteristics**
"Mujhe shaam ko vahaan jaana hai"	"Mujhe saam ko bahaan jaana hai"	Accurate accentspecific pronunciation, prosody aligned with Bhojpuri patterns
"Kal school nahi jaayenge"	"Kal skul nahi jaayenge"	Subtle vowel shifts reflecting regional accent
"Main chai peena chahta hoon"	"Main chai peena chahta hun"	Pitch contour and duration match native Bhojpuri intonation

4.2 Comparison with Baseline Systems

Table 4.2: Comparison of Proposed Model vs Baseline TTS Systems

Metric	**Baseline Speaker TTS**	**Proposed Model**	**Improvement**
Accent Classification Accuracy	61.4%	88.9%	+27.5%
Naturalness MOS	3.85	4.21	+0.36
Accent Congruity MOS	3.60	4.40	+0.80
Inference Time (per sentence)	0.35s	0.38s	+0.03s

Observations:

• Explicit accent embeddings and LLM-driven phonetic adaptation significantly improve accent fidelity.
• Naturalness remains high, showing no compromise due to accent conditioning.
• Slight increase in inference time is negligible for real-time applications.

4. EXPERIMENTAL SETUP

5.1 Data Collection and Curation

To do this project, we needed voice recordings for the AI to learn from. Our final set of voices included three main accents: Indian English Standard Hindi Bhojpuri For each of these, we had about 2 hours of super clear, high-quality recordings. That means our total training data was around 6 hours of audio!

We recorded the Bhojpuri samples ourselves (with permission from the speaker, of course!), and the recordings for Indian English and Hindi were taken from public audio libraries that anyone can use.

5.2 Training Configuration

- Base Model: FastSpeech 2 (TensorFlowTTS)
- Transfer Learning: Cross-Lingual Finetuning
- Accent Embedding: 3D One-Hot Vector across all Transformer layers
- Vocoder: MelGAN / Parallel WaveGAN
- Augmentation: Pitch perturbation ±5%

5. RESULTS AND DISCUSSION

6.1 Objective Evaluation – Accent Classification Accuracy (Ac)

- An independent Accent Classifier evaluated synthesized audio

Table 6.1: Accent Classification Accuracy on Synthetic Speech

Test Set	Baseline TTS	Proposed Model	Improvement ()
Bhojpuri	61.4%	88.9%	+27.5%

6.2 Subjective Evaluation – Mean Opinion Score (MOS)

- Twenty native listeners rated accent accuracy and naturalness on a 5-point scale.

Table 6.2: Subjective Evaluation (MOS)

Metric	Proposed Model	Ground Truth
Accent Congruity MOS	4.40	N/A
Naturalness MOS	4.21	4.55

6.3 Case Study – Phonetic Adaptation

- For the test sentence, “Mujhe shaam ko vahaan jaana hai,” the Bhojpuri output correctly altered phonetics:

• “shaam” → “saam”
• “vahaan” → “bahaan”

6. CONCLUSION

Our study created a new system called Accent-Conditional Fast Speech 2. It can create voices that sound natural and high-quality, even for Indian accents that usually get ignored (the low-resource ones). How did we do it? We used two main key techniques:

The Accent Embedding Layer (that special switch for choosing the accent).

The LLM-G2P adaptor (that high-quality AI dictionary for pronunciation).

Because of these two things, our system is way more accurate and the accents sound completely authentic, especially for the Bhojpuri voice! It proves we can make great-sounding digital voices for all of India's languages and accents.

7. FUTURE SCOPE

- Real-Time Accent Conversion: Optimization for live streaming and conversational systems.
- Scalability: Expansion to additional Indian dialects for broader inclusivity.
- Expressive Speech: Integration of emotion and prosody modeling to enhance expressiveness.

CONFLICT OF INTEREST

The authors declare no conflicts of interest regarding the current research.

REFERENCES

1. Y. Ren et al., "FastSpeech: Fast, Robust and Controllable Text to Speech," NeurIPS, 2019.
2. Y. Ren et al., "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech," ICLR, 2021.
3. Y. Wang et al., "Style Tokens: Unsupervised Style Modeling, Control and Transfer in Endto-End Speech Synthesis," ICML, 2018.
4. R. Kumar, S. Singh, et al., "Annotated Speech Corpus for Low Resource Indian Languages," Interspeech, 2022.
5. TensorFlowTTS, "Real-Time Speech Synthesis Toolkit for TensorFlow 2," 2025.
6. G2P Model Reference, "Open Grapheme-to-Phoneme Toolkit," 2020.
7. K. Kumar et al., "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis," NeurIPS, 2019.
8. (Low-Resource Transfer Learning) (2021). Reference on cross-lingual transfer learning techniques for low-resource TTS.
9. (Accent Conversion Survey) (2020). Survey paper on Voice Conversion or Accent Conversion methodologies.
10. (Real-Time TTS) (2022). Reference for Real-Time and Low-Latency Speech Synthesis.
11. (Indian TTS Challenges) (2018). Paper discussing challenges and approaches for Text-to-Speech in diverse Indian languages.
12. (Disentangled Representation) (2020). Research on disentangling content and style (e.g., accent) features in speech.
13. (Knowledge Distillation) (2019). Paper on Knowledge Distillation for compressing large models into smaller, efficient ones (relevant for low-GPU and live integration).
14. (LLM in G2P) (2023). Recent work on leveraging Large Language Models (LLMs) for improved Grapheme-toPhoneme conversion.
15. (Voice Cloning/Timbre Transfer) (2019). Reference for transferring voice timbre while keeping linguistic content (relevant for future live application)