Onubhuti TTS

Asif Abrar
200041106
asifabrar@iut-dhaka.edu

Shajeed Hossain
200041142
shakun650@gmail.com

Tanvir Hossain Dihan
200041144
tanvirhossain20@iut-dhaka.edu

***Abstract*—This paper presents Onubhuti TTS, a Bangla emotional text-to-speech system that leverages deep learning models for emotion detection. We evaluate the performance of various recurrent neural network (RNN) architectures, including Simple RNN, LSTM, GRU, and their bidirectional variants, alongside transformer-based models such as XLM-RoBERTa, Multilingual BERT (uncased), and BanglaBERT. The models are trained and tested on the Bangla Textual Emotion Dataset, a comprehensive dataset containing multiple emotion categories in Bangla. Comparative analysis using precision, recall, F1 score, and accuracy demonstrates the strengths and limitations of each model in detecting emotions from textual data. The ultimate goal is to integrate the best-performing emotion detection model into a text-to-speech system that can convey appropriate emotional intonation in Bangla, enhancing naturalness and expressivity in synthesized speech.**

I. Introduction

Speech is not only a medium of communication but also a natural carrier of **human emotions**. The ability to express and perceive emotions in speech is essential for effective human–computer interaction. With the rapid advancements in **deep learning** and **natural language processing (NLP)**, emotion-aware systems have gained increasing attention in domains such as conversational agents, mental health support, call center analysis, and assistive technologies. One promising direction is the development of **emotional text-to-speech (TTS)** systems that can synthesize speech with context-appropriate emotional intonation, thereby improving naturalness, expressivity, and user engagement.

Despite the progress in emotion recognition and TTS for high-resource languages such as English and Mandarin, research in **Bangla**—the seventh most spoken language in the world—remains limited. Most existing TTS systems for Bangla are designed for neutral or monotonic speech, with little attention paid to emotional variation. This gap highlights the need for systems capable of accurately detecting emotions from Bangla text and translating them into speech prosody.

In this work, we introduce **Onubhuti TTS**, a system that integrates **textual emotion detection** with **emotional speech synthesis** for Bangla. We explore the performance of various recurrent neural network (RNN) architectures, including Simple RNN, LSTM, GRU, and their bidirectional variants, as well as state-of-the-art transformer-based models such as **XLM-RoBERTa**, **Multilingual BERT (uncased)**, and **BanglaBERT**. These models are trained and evaluated on the publicly available **Bangla Textual Emotion Dataset**, which provides a diverse collection of emotion-labeled Bangla texts.

Through comparative analysis, we examine the relative strengths of **RNN-based** and **transformer-based** approaches in capturing contextual and semantic features of Bangla text for emotion detection. The highest-performing model is then integrated into the proposed **Onubhuti TTS pipeline**, enabling the generation of emotionally expressive Bangla speech. The contributions of this paper can be summarized as follows:

- We benchmark multiple deep learning architectures (RNN and transformer) for **Bangla emotion detection**.
- We provide a detailed performance comparison across models using standard evaluation metrics.
- We propose **Onubhuti TTS**, the first Bangla text-to-speech system that incorporates emotional awareness.

This work demonstrates the feasibility of bridging **emotion recognition** and **speech synthesis** in Bangla, paving the way for more natural, expressive, and user-centric speech technologies in underrepresented languages.

II. Literature Review

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

III. Methodology

A. Data Acquisition

For this study, we utilized the publicly available **Bmotion** dataset hosted on Kaggle. The dataset consists of approximately **27,000 training samples** and **3,000 test samples**, making it one of the largest labeled emotion datasets for Bangla text to date. Each entry in the dataset is annotated with one of six distinct emotions: **Joy**, **Sadness**, **Surprise**, **Disgust**, **Anger**, and **Fear**.

To the best of our knowledge, this dataset has not yet been accompanied by a dedicated research paper or formal benchmark study. This provides an opportunity to establish baseline performance results and analyze the effectiveness of different deep learning models for **Bangla emotion detection**. Our work thus contributes not only toward building an emotion-aware text-to-speech system, but also toward filling this research gap in Bangla NLP.

B. Preprocessing

C. Data Preprocessing

Prior to model training, the raw text data underwent a series of preprocessing steps to ensure consistency, noise reduction, and suitability for deep learning models. The preprocessing pipeline was implemented in Python and applied sequentially as follows:

1. **Text Cleaning:** Removal of unwanted characters such as HTML tags, hyperlinks, digits, email addresses, and non-ASCII elements.
2. **Bangla Script Filtering:** All non-Bangla characters were stripped out using regular expressions, ensuring that only Unicode Bangla script (`U+0980--U+09FF`) remained.
3. **Punctuation and Special Characters:** Unnecessary punctuation marks and special symbols were eliminated to reduce noise in the dataset.
4. **Lowercasing:** All text was converted to lowercase to maintain uniformity.
5. **Stopword Removal:** A curated Bangla stopword list was utilized to remove common but semantically insignificant words. This step helped in focusing on emotionally discriminative words.
6. **Tokenization:** Each cleaned text entry was tokenized into word-level sequences.
7. **Padding and Truncation:** To maintain fixed-length input suitable for RNNs and transformer models, sequences were padded or truncated to a predefined maximum token length.
8. **One-Hot Encoding of Labels:** The six emotion classes (**Joy**, **Sadness**, **Surprise**, **Disgust**, **Anger**, and **Fear**) were converted into categorical one-hot vectors for training.

This multi-step preprocessing ensured that the Bangla text corpus was both **clean** and **normalized**, enabling the deep learning models to focus on learning meaningful emotional patterns instead of noise.

D. Proposed Methodology

The core objective of this study is to evaluate the performance of different deep learning architectures and transformer-based models for **Bangla emotion detection**, and subsequently integrate the best-performing model into an **emotional text-to-speech (TTS)** pipeline. The proposed methodology can be broadly divided into two categories of models: recurrent/convolutional architectures and transformer-based language models.

1) Recurrent and Convolutional Architectures: We first experimented with a set of **RNN-based models** to capture sequential dependencies in Bangla text. The following architectures were implemented:

- **Simple RNN:** A baseline recurrent network with a single recurrent layer followed by dense layers.
- **RNN+LSTM:** Incorporates Long Short-Term Memory units to address vanishing gradient issues and capture longer-term dependencies.
- **Bidirectional GRU (BGRU):** A bidirectional recurrent setup using Gated Recurrent Units to exploit both forward and backward contextual information.

To enhance feature extraction, we also combined **Convolutional Neural Networks (CNN)** with recurrent layers:

- **CNN:** A 1D convolution-based model with pooling layers for local feature extraction.
- **CNN+BGRU:** Combines convolutional feature extraction with bidirectional GRU layers for richer contextual representation.
- **CNN+BiLSTM:** A hybrid model that integrates CNN layers with Bidirectional LSTM layers, enabling both spatial and sequential learning.

2) Transformer-based Architectures: In addition to recurrent and convolutional models, we also evaluated modern transformer-based models that are pre-trained on large multilingual corpora. These models are particularly effective in capturing contextual word representations and semantic nuances:

- **XLM-RoBERTa:** A multilingual transformer trained on 100 languages, capable of strong cross-lingual generalization.
- **Multilingual BERT (Uncased):** A BERT variant trained on Wikipedia text from 104 languages, applied here for Bangla emotion classification.
- **BanglaBERT:** A transformer pre-trained specifically on Bangla corpora, designed to capture the linguistic and syntactic features of Bangla more effectively.

3) Integration into TTS: After benchmarking these models, the highest-performing emotion detection architecture is integrated into the proposed **Onubhuti TTS** system. This system maps the detected emotion labels to corresponding prosodic and acoustic variations in speech synthesis, thereby generating **emotionally expressive Bangla speech**.

IV. Experimental Setup

Table I holds the real essence for our **Deep Learning Model** training setup and architectures.

All experiments were conducted using the **Bmotion** dataset, which was split into approximately 27,000 training samples and 3,000 test samples, with 10% of the training set reserved for validation. The preprocessing steps described earlier were uniformly applied across all models to ensure fairness in evaluation.

For the **deep learning models**, we implemented a variety of recurrent and convolutional architectures, including Simple RNN, RNN+LSTM, Bidirectional GRU (BGRU), CNN,

Model	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14
Simple RNN	Mask	RNN	Dense	–	–	–	–	–	–	–	–	–	–	–
RNN+LSTM	Mask	BiLSTM	Dense	–	–	–	–	–	–	–	–	–	–	–
CNN	Mask	Conv1D	BN	MaxP1D	Conv1D	BN	GlobMaxP1D	Dense	Drop	Dense	–	–	–	–
BiGRU	Mask	BiGRU	Dense	–	–	–	–	–	–	–	–	–	–	–
CNN+BiGRU	Mask	Conv1D	MaxP1D	BiGRU	Dense	Drop	Dense	–	–	–	–	–	–	–
CNN+BiLSTM	Mask	Conv1D	LN	MaxP1D	SpDrop1D	Conv1D	LN	MaxP1D	SpDrop1D	BiLSTM	GlobMaxP1D	Dense	Drop	Dense

TABLE I
ARCHITECTURAL DESIGNS OF DEEP LEARNING MODELS

Model	Precision	Recall	F1 Score	Support	Training Accuracy	Validation Accuracy	Epochs
Simple RNN	0.49	0.48	0.47	2929	53.29%	44.93%	20
RNN+LSTM	0.50	0.50	0.49	2929	56.80%	51.11%	10
CNN	0.62	0.64	0.62	2929	76.85%	62.80%	20
BGRU	0.58	0.58	0.57	2929	73.55%	58.73%	80
CNN+BGRU	0.65	0.65	0.65	2929	**87.09%**	**65.01%**	40
CNN+BiLSTM	**0.69**	**0.70**	**0.67**	2929	65.83%	64.59%	20

TABLE II
PERFORMANCE COMPARISON OF DL-BASED MODELS

CNN+BGRU, and CNN+BiLSTM. These models were trained on sequences tokenized and padded to a fixed length, with embeddings of size 100. Training was carried out using the Adam optimizer and categorical cross-entropy loss, with a batch size of 32. Early stopping was employed to avoid overfitting, and model performance was tracked on the validation set after each epoch.

In addition to the RNN and CNN architectures, we also fine-tuned **transformer-based models**, namely XLM-RoBERTa, Multilingual BERT (uncased), and BanglaBERT. These pre-trained models were adapted for emotion classification by adding a classification head and training with lower learning rates to prevent catastrophic forgetting. Due to their larger parameter sizes, the transformer models were trained for fewer epochs (up to 10) compared to the recurrent and convolutional models.

All models were implemented using TensorFlow/Keras and PyTorch, with pre-trained transformers accessed through the Hugging Face Transformers library. The experiments were performed on an NVIDIA GPU environment, ensuring efficient training and evaluation. Final performance was assessed on the held-out test set using standard metrics including **precision**, **recall**, **F1-score**, and **accuracy**, enabling a direct comparison between traditional deep learning architectures and transformer-based approaches.

V. RESULT ANALYSIS

A. Deep Learning Model Performance Analysis

Table II presents the comparative performance of the evaluated models in terms of training and validation accuracy, as well as precision, recall, and F1-score.

1) Training and Validation Accuracy: From the results, it can be observed that the Simple RNN and RNN+LSTM models exhibit relatively low training and validation accuracies, indicating underfitting. Specifically, the Simple RNN achieves a training accuracy of 53.29% with a validation accuracy of 44.93%, while the RNN+LSTM improves slightly to 56.80% and 51.11%, respectively. These results suggest that such models fail to capture the complexity of the dataset.

The CNN and BGRU models demonstrate higher training accuracies of 76.85% and 73.55%, respectively, but with noticeable gaps compared to their validation accuracies (62.80% and 58.73%). This discrepancy indicates overfitting, where the models learn the training distribution well but struggle to generalize effectively.

The CNN+BGRU achieves the highest training accuracy (87.09%) and the best validation accuracy (65.01%). However, the large gap between training and validation accuracy also highlights significant overfitting. On the other hand, the CNN+BiLSTM exhibits a more balanced performance, with training and validation accuracies of 65.83% and 64.59%, respectively. The closeness of these values suggests better generalization and robustness against overfitting, albeit with slightly lower validation accuracy than CNN+BGRU.

2) Precision, Recall, and F1-Score: In terms of precision, recall, and F1-score, the Simple RNN and RNN+LSTM again perform poorly, achieving F1-scores of 0.47 and 0.49, respectively. The CNN improves significantly with balanced metrics (Precision: 0.62, Recall: 0.64, F1-score: 0.62), outperforming the BGRU (F1-score: 0.57).

The hybrid models, CNN+BGRU and CNN+BiLSTM, provide the strongest results. CNN+BGRU achieves consistent scores across all three metrics (0.65), reflecting solid predictive capability. However, CNN+BiLSTM outperforms all models, achieving the highest precision (0.69), recall (0.70), and F1-score (0.67). This indicates that CNN+BiLSTM not only balances precision and recall effectively but also provides the most reliable predictive performance overall.

Model	Precision	Recall	F1 Score	Support	Training Acc.	Validation Acc.	max_len	Learning Rate & Optimizer
XLM-Roberta	0.44	0.43	0.41	8787	46.48%	43.20%	64	5e-5, Adam
Multilingual BERT Base Uncased	0.51	0.51	0.50	8787	62.84%	**52.85%**	64	5e-5, Adam
BanglaBERT	**0.52**	**0.53**	**0.51**	8787	**64.87%**	52.80%	64	5e-5, Adam

TABLE III
PERFORMANCE COMPARISON OF TRANSFORMER-BASED MODELS.

B. Transformer Model Performance Analysis

Table III summarizes the results obtained from the transformer-based models: XLM-Roberta, Multilingual BERT Base Uncased, and BanglaBERT.

1) Training and Validation Accuracy: The XLM-Roberta model achieves the lowest training accuracy (46.48%) and validation accuracy (43.20%), indicating underfitting and weak capability to model the dataset. Multilingual BERT Base Uncased demonstrates a notable improvement, with training and validation accuracies of 62.84% and 52.85%, respectively. This suggests that the model is better able to capture contextual information and generalize compared to XLM-Roberta. BanglaBERT achieves the highest training accuracy (64.87%) and a comparable validation accuracy (52.80%), only marginally below Multilingual BERT. The closeness of training and validation performance in both Multilingual BERT and BanglaBERT indicates reasonable generalization with limited overfitting.

2) Precision, Recall, and F1-Score: In terms of predictive performance, XLM-Roberta again lags behind with low precision (0.44), recall (0.43), and F1-score (0.41), confirming its weak performance. Multilingual BERT improves significantly, achieving balanced metrics with precision and recall of 0.51 and an F1-score of 0.50. BanglaBERT achieves the best results among the transformer models, with precision (0.52), recall (0.53), and F1-score (0.51). This indicates that BanglaBERT is more effective at capturing linguistic and contextual nuances in the dataset, making it better suited for Bangla-specific tasks compared to general multilingual models.

3) Summary: Overall, deep learning hybrid architectures, particularly CNN+BiLSTM, outperformed transformer-based models in terms of validation accuracy and F1-score, making them more effective for this dataset. However, transformer-based approaches, particularly BanglaBERT, showed promising performance given their smaller performance gap despite being pretrained for broader tasks. This suggests that while CNN+BiLSTM currently provides the strongest results, transformer models such as BanglaBERT hold significant potential and may surpass conventional architectures with further fine-tuning or larger domain-specific pretraining.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.