

BanglaMUX: Enhancing Regional Dialect Detection, Transcription and Translation Performance for Low-Resource Bangla Language

by

Sakib Ul Haque
23341128

Md. Ahnaf Farhan
22241056

Munawar Mahtab Ansary
23341112

Nibir Neelim
21241012

Yasir Arafat
23341117

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2025

© 2025. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sakib Ul Haque
23341128

Md. Ahnaf Farhan
22241056

Munawar Mahtab Ansary
23341112

Nibir Neelim
21241012

Yasir Arafat
23341117

Approval

The thesis/project titled “BanglaMUX: Enhancing Regional Dialect Detection, Transcription and Translation Performance for Low-Resource Bangla Language” submitted by

1. Sakib Ul Haque (23341128)
2. Md. Ahnaf Farhan (22241056)
3. Munawar Mahtab Ansary (23341112)
4. Nibir Neelim (21241012)
5. Yasir Arafat (23341117)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 18, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Farig Yousuf Sadeque
Associate Professor
Department of Computer Science and Engineering
Brac University

Co-supervisor:
(Member)

Nirjhor Datta
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Transcription of various Bangla dialect speeches and translation into standard Bangla text can help the marginalized communities to have better access to information while ensuring their voices are acknowledged and represented in larger texts named “BanglaMUX”. The speech data from speakers of low-resource languages has been preprocessed in chunks with VAD and text data in standard Bangla are trained using techniques of deep learning and acoustic modeling for speech recognition and machine translation models. These models are then adapted to handle unique linguistic features and lexicons of the low-resource languages along with the fine-tuning of parameters and algorithms for enhanced accuracy and robustness. The developed model is expected to capture the intended meaning of the speech and the model can be further enhanced and extended to accommodate new languages and accents in different regions of the world where people can embrace and appreciate the differences in the language rather than limiting themselves. Therefore, this research topic can assist in decreasing the difficulties of several communication barriers experienced by the speakers of various low-resource languages all within a framework.

Keywords: Transcription; Translation; Speech recognition; Machine translation; Deep learning; Acoustic modeling; Low-resource language; Standard Bangla text; Linguistic features; Lexicons; Fine-tuning; Communication barriers; Linguistic diversity; Linguistic preservation; Marginalized communities; Social inclusion

Acknowledgement

First and foremost, all honor is owed to Allah, the Most Gracious and Merciful, who has given us the strength, patience, and resolve needed for this thesis.

Our deepest thanks go to our supervisor, Dr. Farig Yousuf Sadeque, and our co-supervisor, Nirjhor Datta, for their steadfast encouragement and constant support as well as helping moving providing direction at every stage of this research.

We would particularly like to thank Md. Rezuwan Hassan for his assistance especially in handling coding and datasets. His readiness to help at all stages made a positive difference towards completing this work.

Lastly, I would like to express deep appreciation to Bengali.AI for supplying crucial datasets which formed the backbone of this research.

Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Acknowledgment	v
Table of Contents	vi
1 Introduction	1
1.1 Research Statement	2
1.2 Research Objectives	3
2 Literature Review	4
2.1 Survey Methodology	4
2.2 Related Works	4
3 Work Plan	23
4 Methodology	24
4.1 Overview of the Proposed System	25
4.2 Data Collection and Preparation	25
4.2.1 Original Datasets	25
4.2.2 Data Preprocessing	25
4.3 Model Architecture and Training	25
4.3.1 Speech Recognition Model (Tugstugi)	25
4.3.2 Region Detection Model (CNN with Mel Spectrogram)	26
4.3.3 Translation Model (mT5)	26
4.3.4 Region Detection Model (mBERT)	26
4.4 Pipeline Implementation	26
4.4.1 Pipeline 1: Speech-to-Text with Region Detection	26
4.4.2 Pipeline 2: Translation with Region Classification	27
4.5 System Integration	28
4.5.1 BanglaMUX: Unified Pipeline	28
4.5.2 End-to-End Workflow	28

5	Dataset	29
5.1	Analysis of Dataset (Translation)	29
5.1.1	Vashantor Dataset	29
5.1.2	Bengali.AI Dataset	29
5.1.3	Sentence Length Distribution	30
5.1.4	Most Frequent Words	32
5.1.5	Word Cloud	35
5.1.6	Word Co-occurrence Matrix	36
5.1.7	Unique Vocabulary	38
5.1.8	Rare Word Ratio	39
5.1.9	Dataset Preprocessing	40
5.2	Analysis of Dataset (Transcription)	41
5.2.1	Bengali.AI Dataset	41
5.2.2	Audio VS Transcription Length	41
5.2.3	Waveform and Spectrograms	43
5.2.4	Audio Preprocessing and Segmentation	45
5.2.5	Transcribed Texts Preprocessing and Segmentation	46
5.2.6	Data Cleaning and Null Value Removal	47
5.2.7	Text preprocessing and Token Reduction	47
5.2.8	Exploratory Data Analysis on Preprocessed Datasets	48
5.2.9	Dataset Analysis Summary	53
6	Models	55
6.1	Regional Dialects to Bangla Language Translation Models	55
6.1.1	mT5 (Proposed)	55
6.1.2	BanglaT5	55
6.2	Region Detection Models	56
6.2.1	mBERT (Translation)	56
6.2.2	CNN Using Mel Spectrogram (Transcription)	57
6.3	Regional Dialects to their Transcriptions Models	58
6.3.1	Google ASR	58
6.3.2	Bangla ASR	59
6.3.3	Whisper-Small	59
6.3.4	Tugstugi-Whisper-Medium (Proposed Model)	59
7	Evaluation Metrics	61
7.1	Dialect-to-Bengali Translation Metrics	61
7.1.1	Character Error Rate	61
7.1.2	Word Error Rate	61
7.1.3	BLEU Score	62
7.1.4	METEOR Score	62
7.2	Region Detection Metrics	63
7.2.1	Accuracy	63
7.2.2	Precision	63
7.2.3	Recall	63
7.2.4	F1 Score	63

8	Result Analysis	64
8.1	Translation	64
8.1.1	Bangla Dialects to Standard Bangla	64
8.1.2	Region Detection	65
8.2	Transcription	67
8.2.1	Audio to Bangla Transcription	67
8.2.2	Region Detection	70
9	User Interface	72
9.1	Developing The BanglaMUX System	72
9.1.1	Frontend Stack and UI Implementation	72
9.1.2	Backend Integration with AI Models	73
10	Limitations, Failed Experiments and Future Works	74
10.1	Limitations	74
10.1.1	Poor Audio Quality in the Dataset	74
10.1.2	Substandard Transcription Quality	75
10.1.3	Incompatibility of the Transcription Pipeline with the Trans- lation Pipeline	75
10.1.4	Token Length Limitation in Whisper ASR Model	76
10.2	Failures and Observations	76
10.2.1	Learnings from trying to use Wav2vec2 to transcribe Bangla dialect audio	76
10.2.2	Learnings from trying to use Whisper-Small	76
10.2.3	Getting rid of punctuations to generate less tokens	77
10.2.4	Learnings from trying to split speech audio using Voice Ac- tivity Detection	77
10.2.5	Failures in trying to use Forced Aligners to align text	77
10.2.6	Failures in trying to use Whisper-generated word counts to align text	78
10.2.7	Dataset problems discovered while manually trying to align text	78
10.2.8	Learnings from training Whisper-Small on curated-split dataset	78
10.2.9	Learnings from training Tugstugi on curated dataset	78
10.2.10	Transcription Dataset and Translation Dataset misalignment problem	79
10.2.11	Failure to merge Translation Dataset transcripts to align with Transcription Dataset	79
10.2.12	Failure to measure intended pipeline’s Performance	79
10.2.13	Lack of quality in translations generated by Intended Pipeline	79
10.2.14	Challenges in making a usable frontend user interface for the BanglaMUX pipeline	80
10.3	Future Works	80
10.3.1	Addressing Token Length Limitations with IPA	80
10.3.2	Splitting Audio More for Token Optimization	80
10.3.3	Complete Speech-to-Translation Pipeline	81
10.3.4	Developing a Speech-to-Speech and Translation System	81
11	Conclusion	82

Chapter 1

Introduction

Linguistic diversity is one of the major factors that define human civilization. It implies a variety of the cultures of people residing in different parts of the world. This variation frequently presents serious communication obstacles in the areas with limited linguistic resources. Bengali, as a language, encompasses a wide range of dialects that are spoken across Bangladesh and some parts of India, each thriving in its cultural richness. Yet these dialects are largely underrepresented in modern technological applications such as automatic speech recognition and machine translation systems. Thus, speakers of these dialects lack meaningful and easy access to information presented to them in standard language forms which is different from their local dialect and therefore cannot participate effectively in the broader public discourse. For example, the process of writing and usage of standardized Bangla in official and academic communications creates comprehension challenges for the dialect speakers and widens the gap between them and speakers of the standard language.

Bridging this gap brings about better and effective social inclusion of people in the society, making sure that everyone is included in the decision making processes of society at every step. Therefore, to solve this issue for the dialect speakers, we aim to develop robust systems that are able to transcribe speech in low-resource Bangla dialects into Bangla text and subsequently translating the obtained result into its standard form. The main objectives of the research work we present are to develop models by incorporating adaptive tools in deep learning, acoustic modeling and machine translation. These models will recognize the unique linguistic features of these dialect audios to transcribe them to written text and then translate them to standard Bangla all while maintaining the integrity and meaning of the original speech.

Previous research has shown that fine-tuned self-supervised models such as wav2vec 2.0 can be highly effective for the context of low-resource languages. Showrav [1] highlighted that high-performance ASR is feasible even with a small amount of labeled data and limited resources by using a transfer learning-based end-to-end ASR system to Bengali and obtained a Levenshtein distance of 3.819 on a test set of 7,747 samples with just 1000 training speech samples. Meanwhile, the Vashantor framework [2] introduced a dataset of 32,500 sentences from five different regions with distinct dialects, marking the first large-scale attempt to translate Bangla re-

gional dialects into standard Bangla using transformer-based models like mT5 and BanglaT5. Their system showed both dialectal diversity and translation performance potential, achieving a maximum BLEU score of 69.06 and a minimum word error rate (WER) of 0.1548 for the Mymensingh dialect and a BLEU score of 36.75 and a WER of 0.3385 for the Chittagong dialect. Building on these advancements, Samin et al. [3] proposed BanglaDialecto, an end-to-end pipeline for standardizing dialectal Bangla speech with an emphasis on the Noakhali dialect. With a CER of 0.8%, WER of 1.5%, and BLEU score of 41.6%, they fine-tuned Whisper for ASR and BanglaT5 for dialect-to-standard text translation in their diverse dataset. They constructed a diverse dataset and fine-tuned Whisper for ASR and BanglaT5 for dialect-to-standard text translation, achieving a CER of 0.8%, WER of 1.5%, and a BLEU score of 41.6%. The pipeline also incorporated AlignTTS for the generation of standardized speech.

Our proposed system similarly involves training on speech data obtained from native dialect speakers alongside its Bangla transcripts and standard Bangla translations. By fine-tuning parameters of these models and algorithms, we adapt the models to handle the region-specific lexicons and acoustic patterns of the dialects with improved accuracy and usability.

In addition, the implications of this research extend beyond the Bengali language. It can be used to create adaptable and scalable models that can support other low-resource languages globally. Finally, this approach promotes cultural and linguistic diversity within mainstream social integration by enabling marginalized communities to actively participate in the digital world. In this thesis, a detailed understanding is provided on the processes involved in developing these transcription and translation models; the problems faced in this regard are scrutinized and further the implications of these methodologies on linguistic preservation and social inclusion are investigated. With a comprehensive evaluation of the model’s performance, it seeks to provide a positive impact not only on the specific field of natural language processing but also to the furthest extent of expanded research on low-resource languages.

1.1 Research Statement

The research addresses the challenges of the communication obstacles experienced by the different dialect speakers of low resource language. These people are deprived from the use of technological advancements like speech recognition and translation systems for effective communication and are often limited to outdated tools and resources that do not meet their cultural and linguistic needs. Therefore, this study aims to build effective and accurate speech transcription and machine translation models that can help different dialect speakers of Bengali to access information and contribute to public discourse without any language barriers.

1.2 Research Objectives

- Data analysis on the dataset consisting of Bangla dialect speech of different regions and its textual contents of translation in standard Bangla and English.
- Use the most advanced techniques of Machine Learning and Natural Language Processing to identify information and regularities in various regional Bangla dialect speech database related to the corresponding translation.
- Explore the use of suitable and efficient models to identify the region of specific dialect speech.
- Develop models capable of transcribing and translating dialect speech into standard Bangla texts.
- Analyze the efficiency of the proposed models and methods through relevant indices and the use of benchmarks.

Chapter 2

Literature Review

2.1 Survey Methodology

In order to work on our thesis and analyze the possibility of accurately and efficiently translating the dialects of low-resource languages into its standard form, we have thoroughly read a number of technical research articles about the transcription and translation of low-resource languages and its existing models for speech recognition, sequence capturing, and efficient data collection options. Our approach for choosing articles involved the relevant works with a high number of citations in which we highly prioritized those with recent publications. These selected articles have been arranged chronologically based on the date of publication which ensures a coherent body of knowledge that will help with our work in the future.

2.2 Related Works

Paul et al. [4] address the creation of a system for Bangla speech recognition that is organized into two overall phases: speech signal processing and speech pattern recognition. The first phase of speech signal processing involves several major steps, including detection of the beginning and end of speech or spoken words; windowing; filtering; calculation of Linear predictive coding (LPC) and Cepstral Coefficients; and, finally, quantization of the data vectors and create a codebook. The purpose of these steps is to filter out irrelevant sounds and noises and to extract the features of the speech that are necessary to recognize the spoken words. The second phase uses a pattern recognition system based on Artificial Neural Networks (ANN). The ANN will take the extracted features of the speech signal as inputs to recognize the words that were spoken. The system employs the Self-Organizing Map neural network to convert variable-length LPC trajectories into fixed-length feature vectors, which are input into the recognizer. The study explores different neural networks structures including multi-layer perceptrons with both 3, 4 and 5 hidden layers to experiment with performance and accuracy. The overall results show reasonable recognition of Bangla words spoken by both female and male speakers, showing the ability to work with different types of voices. The study mentions prior research on speech recognition, and specifically looked at speaker identification, and LPC coefficients as features for both speech recognition and speaker recognition. This study emphasizes the way recognition technology for speech technologies could create real communication possibilities, especially for those with disabilities, and allow for im-

provements in user interfaces to enable the computer to process human speech as naturally as possible. Also discussed were challenges of speech signal preprocessing, feature selection and neural network training that would allow it to obtain accurate recognition of Bangla speech. Conclusions stated that the study merges knowledge of signal processing with "cutting edge" methods of modern pattern recognition to aid in overall progress of Bangla speech recognition technologies.

Ali et al. [5] introduce a system that identifies spoken Bangla words, targeting the confusion caused by similar-sounding words and syllables. The study points out that Bangla recognition must adapt to regional speech, since dialects shift pronunciation even within small areas. Their approach first digitizes the recorded signal, then processes it through cleaning filters before pulling key features with Mel-Frequency Cepstral Coefficients MFCC and Linear Predictive Coding LPC. Those features are lined up with stored models using Dynamic Time Warping DTW and Gaussian Mixture Models GMM. The team tested four combinations of extraction and matching methods to see which worked best at understanding spoken Bangla. The study found that Model 3, which had MFCC + GMM, provided the highest recognition rate and demonstrated the successful identification of words with limited errors. Model 4 used MFCC, so while it has LPC for compression and completed the recognition tasks faster than the other models which gave it a faster recognition time, its recognition rate is lower, thus, indicating a positive trade-off between time to recognize and recognition accuracy. With Model 2, which has LPC & DTW for matching, it displays less processing time with less recognition accuracy than Model 3. Model 1, which has MFCC + DTW, showed a balanced recognition rate and recognition time, however, it was still not as efficient as Model 3 in recognition time and recognition rate. Therefore, Model 3 emerges as the ideal model offering high recognition rate and relatively fast recognition time. especially for applications where quick processing time and high recognitional accuracy are required or particularly useful such as transcription systems and/or voice-activated applications for Bengali-speaking people.

Kalita et al. [6] focused on the development of a Bengali to Assamese Statistical Machine Translation (SMT) system using the Moses toolkit. The goal was to solve translation demands in a multilingual country like India. The paper employed an SMT framework called Moses and tools like IRSTLM for the language model and GIZA-PP-V1.0.7 for the translation model, all of which operate in a Linux environment. The authors developed a phrase-based translation model by using Moses. The parallel corpus used for training includes 17,100 sentence pairs in Bengali and Assamese. While the Translation Model (TM) aligned the Bengali input with its Assamese equivalent, the Language Model (LM) assisted in producing fluent Assamese output. To enhance translation accuracy, these models were integrated by the decoder. The BLEU score was the standard measure to evaluate the quality of the text translated by machine. It was used to analyze the success of the translation system. The BLEU score obtained from the parallel corpus (Bengali-Assamese) was 16.3. This relatively low score suggested that the translation quality was not optimal. The machine translation system was able to translate some words between Bengali and Assamese. The testing showed that its effectiveness was limited due to the quantity of the dataset and the lack of complete linguistic resources. For

enhancing the measure of translation fluency and BLEU score, still more and better quality corpora are needed along with further and accurate part-of-speech tagging. As mentioned by the authors, the current approach can be improved by considering other methods of evaluation and by adapting shallow syntactic features that can improve the score obtained. The conclusion of the study thus highlighted the need to further enhance the corpora and the assessment procedures so as to get an even better score on the BLEU scale as well as the quality of translation. The ongoing goal is to develop a proficient web-based translating system for translating numerous selections and domains to provide native language content access across India.

Gutkin et al.[7] identified that the first and main subject under investigation in his research is to generate a high-quality TTS synthesizer for Bangla language which is used everyday in Bangladesh and some areas of India. Developing TTS systems for less pronouncing languages such as Bangla poses a lot of difficulties partly because of poor availability of linguistic and speech databases. In this paper, the author's aim was to analyze this process and subsequently propose the use of approaches like crowd-sourcing and statistical analysis. Out of fifteen Bangladeshi employees present at Google's Mountain View campus, those who auditioned were hired from Dhaka and have similar vocal dispositions. It was conducted with an ASUS Zen fan-less laptop, Neumann KM 184 microphone, a USB converter and a preamp that cost below two thousand US dollars in total. Overall, the volunteers volunteered around 250 phrases in three days in an anechoic studio giving approximately 2000 phrases. A phonemic Pronouncing Dictionary was made, which contained over 65000 words transcribed by the linguists. The dictionary was in Bengali as well as Latin which was approximately 81 percent accurate. Two kinds of acoustic modeling were investigated and they include; Hidden Markov Models (HMMs), a technology in which the dependent distribution is conditioned on the related features of linguistics by applying HMMs and Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs) which were employed for discovering the temporal relations and improving the synthesis quality different from the HMMs' frame-independent assumption. The vocoding stage was carried out by using the Vocaine algorithm that recovers the speech based on the obtained parameters of the sound. The first process of text normalization was to tokenize texts and demarcate opinions on the basis of punctuation aspects. Each subtoken was given classification and non-standard opinions were verbalized. Since there was already an availability of the starting resources for Hindi in the form of Google text normalization system, the Kestrel was developed in Bangla. Subjective speech quality in terms of the TTS voices used was determined based on the result of MOS tests which are commonly used to estimate the perceived quality.

Godard et al. [8] presented a paper addressing the challenge of recording and documenting languages on the verge of extinction that lack enough written records and written resources. The main objective of the authors is to aid field linguists in the semiautomatic analysis and annotation of audio recordings of these languages through computational methods. The focus is on developing effective speech technology tools for languages with minimal or no digital resources. The dataset used in this study comprises about 5,000 speech utterances in the Mboshi language (a Bantu C25 language that is spoken by people in Congo-Brazzaville), which are aligned

with French text translations. This dataset also includes non-standard graphemic transcriptions of Mboshi that are closely aligned with its phonology. This corpus was collected and processed under real-world language documentation scenarios, involving both human translators and automated tools. The paper also explores the application of unsupervised models for word discovery in speech data. Specifically, the pipeline mentioned integrates two main components: unsupervised phone discovery (UPD) from speech, identifying pseudophone units, and unsupervised word discovery (UWD), which aggregates these units into potential word-like entities. The authors make use of existing tools like BLSTM for phoneme boundary detection and dpseg (an algorithm for piecewise linear segmentation of ordered data using dynamic programming), a Bayesian non-parametric approach, for the actual word segmentation. The results they found highlight the efficacy of the proposed models in identifying word boundaries and tokens but with challenges in precision. The best model configuration achieved an F-measure of 34.8 for word boundaries but lower scores for word tokens and types. This indicates that it has difficulty in accurately segmenting and identifying discrete words from just speech data. The current models demonstrate limitations in the quality of segmentation and the granularity of pseudo-phone units. Future improvements could involve making the unsupervised phone discovery process more refined and then better integrating bilingual data thus enhancing segmentation quality. Additionally, extending the dataset to include word-level alignments could support more advanced tasks like bilingual lexicon discovery, which may aid further in the robust analysis of endangered languages.

Vaswani et al. [9] introduced the Transformer model. It is a solution to the limitations of recurrent and convolutional neural network architectures in sequence propagation tasks. Language translation difficulties involve converting variable-length symbol sequences from one language to a different one. It demands the use of models that are capable of preserving complicated links between both input and output sequences. RNNs and CNNs are the most recent developments in sequence modeling. However, these systems highlight inefficiencies in performing the task. Parallelization and scalability are challenging with these methods, especially with longer sequences. Transformer attempts to fix these restrictions by keeping or improving performance in sequence propagation tasks. The authors evaluated the Transformer using the dataset of English-French and English-German translations (WMT 2014), which included millions of phrase pairings. They preprocessed the datasets with byte-pair encoding to handle out-of-vocabulary terms and established a shared vocabulary across source and target languages. The Transformer design departs from typical recurrent and convolutional layers by depending only on self-attention processes for both encoding and decoding. Self-attention enables the model to capture global relationships between input and output sequences while remaining unconstrained by sequence length. This strategy aims to increase parallelization and computing efficiency. This is also appropriate for tasks with extensive sequences. Furthermore, the model’s multi-head attention operation helps it to focus on different parts of the input sequence at the same time, strengthening its capacity to record complex data connections. Experiment results verify the transformer model’s usefulness. The large Transformer model surpasses prior benchmarks by more than 2 BLEU on the English-German problem and a BLEU score of 41.0 on the English-French challenge. Even the basic model, which takes far less training time, beats

earlier models. Still, the Transformer model may have drawbacks. Its application to tasks other than sequence transduction, such as picture or audio processing, is yet to be investigated. Furthermore, the Transformer’s enhanced productivity may come at the cost of interpretability, as self-attention processes may result in less intuitive models than conventional systems.

Saurav et al. [10] examine the speech recognition task for the Bangla language with the specific aim of developing a voice search module for the Bangladeshi search engine Pipilika. The research compares the most popular methods for the task by developing two main models, the Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) and Deep Neural Network-Hidden Markov Model (DNN-HMM), with the aim of building a system that could be used for Bangla speech recognition. The authors performed their investigations using a corpus of 9 hours of recordings from 49 speakers with a vocabulary of 500 unique terms. The authors also indicate that they achieved a Word Error Rate (WER) of 3.96% for the GMM-HMM model and a WER of 5.30% for the DNN-HMM model, which is possibly the lowest reported WER for a Bangla speech recognition task of this vocabulary size. The study addresses the problem of speech recognition in the Bangla language that has lagged behind others such as English and Mandarin, where deep learning has made considerable advances. The research uses the Kaldi toolkit for speech recognition, an open-source platform to implement the models and concentrates specifically on the performance of DNN-HMM and GMM-HMM for Bangla speech recognition. In detail, the DNN-HMM model had a lower word misrecognition rate than the GMM-HMM model, but was sensitive to noise, where the background noise could sometimes be mistaken for words. The study outlines a comparison of their results with past works in the area, and stresses that with their method they were able to achieve the best accuracy using the largest dataset for Bangla speech recognition. In conclusion, while Bangla Speech Recognition is developing, this research offers a reasonable head start, demonstrating improvement not only with the vocabulary size, but the accuracy of systems for Bangla speech recognition with possibilities for voice search systems in the country.

Abe et al. [11] presented a model for multi-dialect neural machine translation, specifically made for Japanese, which can convert Japanese word chunks in dialect-to-standard and standard-to-dialect directions with high degree of efficiency. The researchers aimed to bridge the gap between Japanese dialects and standard Japanese through machine translation (MT). They sought to facilitate the application of natural language processing tools that usually use standard Japanese to understand and interact with the dialects of Japan. This is especially important in contexts like personal assistant technologies and smart speakers, which could then communicate in the user’s native dialect. The study utilized the National Dialect Discourse Database Collection, which contains 48 Japanese dialects where each of the dialects is represented by transcribed 30-minute conversations. This corpus provides data of dialects and standard Japanese texts which are parallel to each other, which is essential for training the MT model. They developed a multi-dialect neural machine translation (NMT) system that works specifically for Japanese. This system integrates multilingual NMT to handle multiple dialects and employs ‘kana’, the Japanese phonetic lettering system in an encoder-decoder framework. It also em-

employs 'bunsetsu' (Japanese phrase units) as the translation units rather than a sentence by sentence translation to preserve the word order which is consistent across dialects and standard Japanese. The multi-dialect NMT model was more accurate than baseline models when translating between the Japanese dialects and standard Japanese and vice versa. The ability to employ dialect embeddings within the model helped in visualizing and comparing the geographic and typological connections of dialects. The model was able to capture the phonetic and syntactic resemblance of many dialects which enhanced the translation quality. By taking in only phrases it was able to gain 5.92 points improvement over the same model but with whole sentence inputs in terms of BLEU scores. Although the model indicated good performance, it has drawbacks when it comes to long-range dependencies because it works at the phrase level only. Some possible enhancements in the future might be to incorporate more context data to enhance the context understanding and the quality of the translation. Moreover, future studies could also extend to improving the model's discriminative and translation performance for more subtle dialectal differences that may need more context and knowledge of regional contexts.

Hasan et al. [12] investigated the translation of Bangla to English language pairs by using numerous Neural Machine Translation (NMT) algorithms such as Bidirectional Long Short Term Memory (LSTM) and Transformer based NMT. A source sentence in Bangla is processed by the encoder of a typical NMT into a context vector which will be used by the decoder for the prediction of words in the target language in English. They made use of the publicly available corpora, which included SUPara Corpus, Six Indian Parallel Corpora (SIPC), AmaderCAT Corpus, Indic Languages Multilingual Parallel Corpus (ILMPC), and Penn Treebank Bangla-English parallel corpus (PTB). They combined all these dataset which resulted in the creation of the largest dataset that consists of Bangla and English language pairs. During the training, they used word embeddings that had already been trained to initialize the model's parameters. They used the Glove model for English and the word2vec model for Bangla. The benefit of NMT is that it can be trained in one large neural network to learn the mapping provided by input to the output in a comprehensive manner. The parameters are learned by the model for maximizing the performance of the translation output using minimum domain knowledge. Transformer-based neural networks use attention based mechanisms to deal with long sentences, in which the model learns to do alignment and translation at the same time. Rather than focusing on a single dimension, the transformer analyzes every word according to the types of input sequences that were provided. Their experiments performed better in the case of both ILMPC and the SUPara test sets than all previously reported results. Using the best models, their system showed 14.63% relative improvement on the SUPara and 32.18% on the ILMPC test sets. Their system performed better on the ILMPC dataset when compared to Google Translator but failed to reach that same level of performance on the SUPara dataset. After analyzing the datasets, it was found that some noisy translations affected the performance and evaluation by the merging of the dataset where some of the same words in American and British English had slightly different spellings. Finally, they discovered that NMT struggled to translate rare words and also the meaning of some words changed after adding more data that led to inconsistent translations.

Siddique et al. [13] described the design of a Machine Translation system that is able to translate texts from English to Bangla using a recurrent neural network (RNN) in conjunction with different word attention weights and combined parsing factors. This system was put into practice with the recurrent neural network having both encoder and decoder. This model mapped Bangla and English words using a context vector based on knowledge. The main dataset of their research was collected from various articles that were manually written down in Bangla and English which included 4000 English and Bangla parallel sentences with highest lengths of 8 Bangla and 7 English sentences. All the words in Bangla and English sentences are tokenized according to how frequently they occurred, and then padding is used to ensure that every tokenized sentence has the same length. This allowed the dataset to be prepared for their neural network model’s input by shaping and vectorizing the input data. To create a context vector, inputs are given as tokenized sentences in Bangla and English, where their mapped tokens are converted to attention weights. This is used to calculate scores for encoder and attention weights which represent the Bangla tokenized sequence attention over the English one. The RNN encoder has the embedding layer as its first layer and the GRU layer as its second layer. While the decoder and encoder are almost similar, the decoder also has a dense layer that uses the attention score to calculate the corresponding token sequence. Subsequently, the tokenizer decodes the sequence into Bangla sentences by removing the padding. Finally, the use of linear activation function in the encoder layer and tanh activation function in the decoder layer achieved the best performance. Moreover, efficient results are observed when sigmoid function is used as inputs in the attention layer while softmax is used as output in the attention layer. In a total of 100 epochs, the mean error decreased to 0.107 that indicated a good performance as the error determined the accuracy of the model. Thus, their proposed deep learning approach outperforms traditional translation techniques and has the benefit of accurately capturing the actual meaning of a given sentence based on its context.

Zampieri et al. [14] released a survey outlining all the different works being done inside the natural language processing space to classify or process similar languages, language varieties and dialects. The paper addresses challenges in the natural language processing (NLP) of similar languages, language varieties, and dialects. It focuses on computational methods to improve applications like machine translation, speech recognition, and dialogue systems in handling diatopic language variations which are the geographical variations in the same language. Several datasets were discussed. This included the International Corpus of English, the Arabic dialectal text corpora, and the Discriminating between Similar Language Corpus Collection (DSLCC), which contains texts from multiple countries’ national language varieties. These datasets were compiled using various methods, such as transcribing speech, collecting data from social media, and using written sources that reflect national dialectal or language differences. The paper surveys several computational models and approaches for processing language variations. These include techniques for part-of-speech tagging, techniques for parsing, language and dialect identification, and MT, specifically designed for closely related languages and dialects of a language. The survey covers models that leverage cross-lingual transfer learning, annotation projection, and relexicalization among others. The survey indicates that these specialized NLP models and techniques can significantly improve

the processing of language variations compared to general-purpose models. These improvements are crucial for applications requiring high linguistic accuracy such as machine translation and speech recognition systems. However, specific results or performance metrics from individual studies were not detailed in the summary provided. The paper notes several limitations, including the challenge of collecting and preparing adequate datasets for underrepresented languages and dialects. It also points out the difficulties in achieving high accuracy when dealing with closely related languages or dialects that share many linguistic features. Future work could focus on enhancing data collection methods, developing more robust models for handling linguistic similarities, and improving the integration of these models into practical NLP applications.

Milon et al. [15] described a process of transforming a specific dialect i.e Chittagonian into standard Bangla language. The paper also shows that there are challenges in translating the text which may contain idiomatic expressions of Chittagonian or some words with meanings different from the standard Chittagonian but are contextually appropriate, which cannot be easily translated into Standard Bangla. The main dataset comprises a lexicon that links Chittagonian words with the equivalent Standard Bangla words. This dictionary was made from the data of Noor Mohammad Rafiq’s book which contains a large number of words and example sentences. Secondary data was collected through existing published materials, websites, social media posts and comments. The Chittagonian root words were obtained with a total number of 20,101. These root words give the concrete structure of the words by which the word-to-word mapping works. A complete Chittagonian word list for the rule generation and training of the system consists of 5010 complete Chittagonian sentences. An added 2,230 Chittagonian complete sentences were generated for the purpose of evaluating the capabilities of the system. Fundamentally, the conversion system depends on the Chittagonian to Standard Bangla Bilingual Dictionary, which searches for Chittagonian words and then provides their suitable counterparts in Standard Bangla. It establishes the primary framework for the translation process and its first task is to map words in one language to words in another language. Utilizing an OOV-word decoder Unit ‘Word Suggestion Module’ resolved the out-of-dictionary vocabulary words. This module uses the algorithm of machine learning and tries to guess the Standard Bangla translations by identifying the context or any pattern. Morphological Rules, the implemented system, contains a collection of rules about the correspondence between Chittagonian suffixes to the related Standard Bangla ones. Suffix Handling and transformations to the extracted words are developed in a way so that the converted words do not sound improper and are grammatically accurate in Standard Bangla. It returns an accuracy rate of 94.75% and the workflows could easily be altered as the system develops as said by the author.

According to Bhattacharjee et al.[16], BanglaBERT is a new language model and it is designed to deal with established issues in understanding and processing the Bangla language. It has always been neglected in natural language processing (NLP). In spite of having limited resources, researchers began a comprehensive gathering of information effort and they obtained a large and diverse corpus of Bangla texts from various places. To ensure the quality, this corpus was preprocessed which

included language classification, duplication, and the removal of non-textual material. BanglaBERT has a transformer encoder having 12 layers with 12 attention heads, hidden dimensions totaling 768 and 110 million parameters together in its development. As such, BanglaBERT is able to recognize more complex contextual information in Bangla. It might end up in advanced and complex language representations. BanglaBERT’s pretraining technique is an important improvement since it differs from the masking language modeling (MLM) methodology. On the contrary, it takes advantage of the discriminator-generator structure of the ELECTRA aim. The discriminator determines whether hidden tokens are duplicates or real and the generator accepts this decision. This approach enhances computational efficiency and also gives learning signals greater depth so that training data can provide more profound insights. Using the Adam optimizer and a carefully chosen learning rate with a linear warmup schedule, BanglaBERT was tuned during the initial pre-training phase on the large Bangla text corpus. This optimization technique, having been carried out over a number of training stages, allows BanglaBERT to obtain full representations of the Bangla language. The end result of their work is a robust language model that excels in diverse Bangla language comprehension jobs. BanglaBERT has been tested on diverse tasks, including sentiment classification, emotion classification, document classification, named entity identification and natural language inference. In all of these tasks, BanglaBERT often beats previous multilingual and monolingual models by 1% to 3.5%.

Kowsher et al. [17] discussed BnVec, a free library designed for word embedding in Bengali language. Despite Bengali’s widespread adoption, there are no thorough word embedding methods. This is the basis for BnVec. Sentiment analysis, named entity identification, and document categorization all rely significantly on word embedding. However, the lack of suitable word representation models particularly created for Bengali is an important obstacle to realizing the full potential of NLP in Bengali. As a result, BnVec focuses on gathering a diverse corpus of Bengali text from several sources, including news items, social media material, and Bengali Wikipedia. This corpus is carefully preprocessed to remove non-Bengali substance, punctuation, normalize Bengali letters, URLs, and emoticons, tokenize the text, and delete stop words. This data processing and handling ensures the precision and importance of the data used for training the word embedding models. BnVec implements five essential models for word embedding: CountVectorizer, TF-IDF, Hashing Vectorizer, Word2Vec, and fastText, all of which are optimized for Bengali. The set of tools also includes GloVe, a model that learns embeddings from word co-occurrences, and fastText, which is well-known for its ability to handle subword input. These models are trained on a preprocessed Bengali corpus that uses various vector dimensions, window widths, and iterations. The authors’ research setups were carried out in Python, with Google Colab and Tensor Processing Units (TPUs) for efficient execution. The training performance of each model and architecture is evaluated to assess computational efficiency and efficacy in capturing semantic connections in Bengali text. BnVec is basic to install with pip and provides users access to models that have been trained and vectorizer classes. It permits the use of word embedding techniques without requiring massive training datasets, minimizing time and computational resources, especially for devices with restricted specifications. BnVec has been studied both qualitatively and quantitatively. The qualitative eval-

uation concentrates on the manner in which the models learn word similarities via analysis of word vectors using clustering techniques. In quantitative evaluation, text categorization is performed out employing the Bangla film review corpus. The Skip-gram model with SVM provides accuracy considered to be highest (85.92%) and an F1 score of 84.09%.

Yi et al. [18] discussed the problem of Bengali voice recognition, which lags behind improvements in languages like English. Because of the lack of transcribed Bengali voice data, especially for supervised learning, the language is regarded as low-resource in this context. To deal with this difficulty, the paper offers wav2vec2, which is a self-supervised learning model suitable for raw audio data. The wav2vec2-large-xlsr-53 model is chosen and fine-tuned using the Bengali Common Voices Dataset. This model employs a traditional neural network (CNN) to encode voice sounds and a Transformer network for contextualization, allowing it to successfully capture speech patterns and characteristics. The preprocessing stage consists of various stages that optimize data for training. Firstly, sampling the audio files at 16kHz, converting characters in label strings to integer hashes, and finally filtering unreliable data based on upvote and downvote metrics. The Transformers library implementation pads each array to match the longest array in the batch for efficient training. Training is done in two phases. Initially, the model is trained for 71 epochs followed by further training on a combined dataset of training and validation sets for 7 more epochs. The fine-tuning process is to enhance the model’s exposure to Bengali speech and text labels for better performance in speech transcription tasks. The training uses the pytorch-based implementation of the Transformer model. The results describe the efficacy and efficiency of this approach in Bengali speech recognition. The Word Error Rate (WER) of this model is 0.2524 on the validation set, indicating accurate transcription. Evaluation on a hidden test set yields a Levenshtein Distance of 2.60753. It reflects the model’s ability to transcribe Bengali speech with relatively low error rates. The model outperforms other submissions in a Kaggle competition with a Levenshtein Distance of 6.2341 on the hidden dataset. While the study highlights the effectiveness of wav2vec2 in Bengali speech recognition and the benefits of fine-tuning on specific datasets, it also acknowledges certain limitations. For example, the previously trained model’s focus on multilingual voice data, with a major emphasis on European languages, could hinder its performance on Bengali due to deviations in phonetics and linguistic structure.

Kale et al. [19] introduced mT5, a multilingual variant of T5 to construct a vastly multilingual model with the least possible changes from the recipe that was used to make T5. A new Common Crawl-based dataset consisting of 101 languages was used to pre-train this model. They modified the training of mT5 and presented its cutting-edge performance on several multilingual benchmarks. In order to train mT5, they introduced mC4 dataset made up of texts from 101 different languages using the public Common Crawl web scrape. Masked language model was used to pre-train T5 in which a mask token is used to replace successive spans of input tokens, and the model is then trained to recover the tokens that were hidden. They considered five different model sizes with an increasing number of parameters starting from 300 million to 13 billion parameters. The increased vocabulary in mT5 accounted for the rise in the number of parameters as compared to the

corresponding version of T5 model. They also employed a very high character coverage and activated the byte-fallback feature of SentencePiece so that every string could be encoded uniquely in order to support languages with large character sets. Since mT5 is an encoder-decoder model, its parameter count is almost double that of correspondingly-sized encoder-only models like XLM-R where both models have nearly the same computational cost for text classification. Their biggest model mT5-XXL achieved the best results on all classification and QA tasks compared to the latest technologies. When a pretrained multilingual model was fine-tuned in English and tested on other languages, zero-shot cross-lingual span selection created a more challenging scenario for the generative models. The mT5 model remarkably performed “spontaneous” translation by generating legal non-English predictions despite never seeing parallel training data but only the English targets in fine-tuning. However, “accidental translation” was sometimes observed in the zero-shot setting where a portion of the model’s prediction was translated into the wrong language. Thus, the paper illustrated that the recipe of T5 can be straightforwardly applied to the multilingual settings by achieving strong performance on a variety of benchmarks.

Sultana et al. [20] provided a comprehensive overview on Automatic Speech Recognition (ASR) advancements on the Bangla language. It discusses the advancements in developing the Bangla ASR system, especially in the last 10 years, and emphasizes several design factors such as vocabulary size, speaker dependent/speaker independent, classification approaches and many others. The paper also discusses different challenges for Bangla speech recognition, like variations in regional accents, phonemic systems, and speech patterns. This paper stresses challenges for the various dialectal possibilities that Bangla language poses since the words can even mean differently from one region to another, or it could be only a pronunciation difference with the same meaning. For instance, the word “লাফ” (leaf) might be pronounced as “লাফ” in the Sylhet region. Other challenges include insufficient annotated speech datasets, the presence of homophones, and the mismatch between spoken and written forms of words. The paper also discusses consonant clusters, word boundaries, and noise in the surroundings of Bangladeshi atmosphere, which confuse a recognizer. It also reviewed the development of Bangla ASR systems including the move towards Deep Learning approaches, in particular Artificial Neural Networks (ANNs) and the improvements made as larger training data become available. Despite this advancement, the paper stresses that more development is needed with language and pronunciation models before we can say Bangla ASR can effectively accommodate all its features. Finally, the paper addresses the growing applicability of the machine learning models and how ASR may continue to develop as larger datasets are made available. It also informs researchers how despite the advances made there is still much more that may be done with respect to the issues created by dialect, background noise, and homophones in order to make a Bangla ASR fully usable.

Sarker [21] proposed BNLP that lowers the obstacles for doing different NLP tasks for Bengali such as tokenization, word embedding, named entity recognition(NER), and part of speech(POS) tagging facilities. In addition, BNLP provides a number of popular text preprocessing methods like the removal of stopwords, punctuations, and words that are not in Bengali. BNLP also provides various pre-trained Bengali mod-

els such as sentencepiece, word2vec, fasttext, CRF-based NER tagging model, and CRF-based POS tagging. They used Bengali raw text data from wikipedia dump dataset and different news portal sites from where they used crawl news articles for training sentencepiece, word2vec and fasttext where Wikipedia corpus included 1818523 sentences with 32908419 tokens and News article corpus contained 4017940 sentences with 60526710 tokens. Moreover, they employed nltr datasets for POS tagging which contain a total of 2997 sentences. Furthermore, they utilized NER-Bengali-Datasets for NER that has 67719 data in total. Tokenizers BNL provides a pre-trained sentencepiece model that can be used by anyone to tokenize Bengali text. By calling train API, users can also train their own sentencepiece model in Bengali. BNL provides two distinct methods for embedding Bengali words either by using a pre-trained model or by training Bengali word2vec or fasttext model from the beginning. Additionally, BNL offers a practical starting option for POS tagging in Bengali by providing techniques of using a pre-trained CRF-based model for the tagging of parts of speech from any sentences that are given. BNL also provides an option for using custom POS datasets to train a CRF-based POS model. Besides, NER Tagging BNL offers a practical way to introduce NER to Bengali by providing techniques where the named entities are tagged by using a pre-trained CRF-based NER model from a given sentence. Finally, their study showed an evaluation result of 68.88 F1 score for the NER model and 80.75 F1 score for the POS tagging model. Thus, BNL pre-trained model yields noteworthy results with high accuracy in the tokenization of texts, word embeddings, NER, and POS tasks for Bengali.

Bhattacharjee et al. [22] presented a BERT-based Natural Language Understanding (NLU) model specifically pretrained for Bangla named BanglaBERT. The authors wanted to improve Bangla NLU by developing language-specific models because they were aware of the shortcomings of current multilingual models which frequently exhibit poor performance on low-resource languages. The authors gathered 27.5 GB of text material in Bangla from 110 well-known Bangla websites in order to pre-train BanglaBERT. This extensive dataset serves as the foundation for efficient pre-training, which enables the model to pick up on the nuances of the Bangla language. Question Answering and Natural Language Inference are the two new datasets that the authors developed for Bangla. These datasets serve as benchmarks for assessing the model’s performance on a variety of NLU tasks such as text classification, sequence labeling and span prediction. The authors compiled these tasks into the first-ever Bangla Language Understanding Benchmark (BLUB). Alongside BanglaBERT the authors introduced another model named BanglishBERT which is pretrained in both Bangla and English to facilitate zero-shot transfer learning. BanglaBERT and BanglishBERT were evaluated against widely used multilingual models. The evaluation results showed that BanglaBERT performs noticeably better than XLM-R and mBERT. Specifically, BanglaBERT achieved a 6.8-point and 4.3-point higher BLUB score compared to mBERT and XLM-R respectively. BanglishBERT performed much better than these models in the zero-shot cross-lingual transfer condition. It outperformed the others by 15.8 and 10.8 BLUB scores respectively. The authors concluded that language-specific models offer significant advantages over multilingual models for low-resource languages. The lack of Bangla-Specific Benchmarks was highlighted by the study also. With that in mind, the authors proposed

future research directions such as expanding BLUB to include other NLU tasks like dependency parsing and exploring the benefits of initializing Bangla Natural Language Generation (NLG) models from BanglaBERT.

Sahoo et al. [23] aims to fill one of the key gaps in the field of language detection by proposing an approach to handle the Santali language. A newly developed dataset and an improved machine learning method are shown in the study to clearly recognize Santali when written in the Odia script but not in the original script. The focus of the work is the use of a deep supervised autoencoder to perform the language identification task. An autoencoder is a neural network used to create compressed representations of data, often for the purpose of feature extraction or for the reduction of dimensions. In this study, labeled data was used to train the autoencoder to distinguish between a text written in Santali language and Odia language. A deep supervised autoencoder was also tested and it was able to correctly classify the test dataset 100% of the time. This high accuracy has clearly depicted the capability of the model in differentiating between Santali and Odia both of which employ the same script. This supports the efficiency of this methodology especially for other low resource language detection problems. The authors also recommended several directions of further research focusing on enhancing the model’s reliability and applicability. Some of the future works are introducing large and diverse dataset with new Santali texts, comparing the performance of the deep supervised autoencoder when applied to the other similar dialect of Odia language, and finally developing larger NLP tools and technologies for Santali. The authors intend to contribute to continued work and preservation of such endangered low-resource languages as Santali in today’s world with their findings and the provided dataset.

Xue et al. [24] presented ByT5 which is a version of multilingual T5 model that does not use tokens. It can process byte sequences and operate on raw text directly without relying on vocabulary that is already learned to map words or subword units to tokens. Using tasks from the well-known XTREME benchmark suite, which includes XNLI, PAWS-X, XQuAD, MLQA, TyDiQA, and WikiAnn NER, they compared ByT5 and mT5 models with the same parameters. They also used Dakshina benchmark having 12 South Asian languages for translating a Latin script word to native script and used SIGMORPHON 2020 having 15 languages for mapping a word to its pronunciation. In this work, they adopted a more straightforward method and demonstrated that the Transformer architecture can be simply modified without experiencing an unfavorable increase in computational cost as the byte sequences are often much longer than token sequences. Through ablations, they discovered that a “heavier” encoder greatly improves byte-level models by decreasing decoder capacity while masking of longer ID sequences benefits the pre-training task. In multilingual models, large vocabularies can account for a substantial percentage of the model’s parameters. Hence, ByT5 transfers these parameters into the transformer layers by taking them out of the vocabulary. ByT5 achieved much better performance than the previous state-of-the-art results of mT5 on all the tasks and model sizes where there is an availability of gold training in all languages. ByT5 also beats mT5 at smaller sizes on the translation tasks but the results are inconsistent at bigger sizes of the model. It was observed that mT5 performed better in languages with greater SentencePiece token compression rate like Telugu, while ByT5 performed better on

languages with a lower compression rate like Indonesian. They also found that ByT5 was more resistant to noise in a range of tasks and languages compared to mT5. However, ByT5 processes 1.5 to 2.6 times slower which is even more noticeable for tasks involving longer input sequences, but the model has the ability to generalize better from a limited amount of training samples. The bytelevel models are thought to be more data efficient learners as ByT5 performed really well even though it received four times less texts for pretraining than mT5.

Tomal et al. [25] classified two different dialects on the basis of local Bengali text. This classification gets more challenging when dealing with a language that is only spoken among local people but is not widely available in written form except. The dataset included two widely spoken dialects from the regions of Chittagong and Pabna, which have a large number of speakers. This dataset contains about 5000 data in sentences and paragraphs on these regional languages that are also annotated with their corresponding dialects. These were collected from local Bangla dramas, news portals, and local social media contents including videos containing regional dialects where the speeches were converted to their corpus. Since the algorithms are unable to comprehend raw text, various ways of processing methods such as word embedding techniques and text feature extraction are used to ensure that the corpora is understandable to their algorithms. Word embeddings provide a method for using an efficient, dense vector representation where related words have similar encodings of floating point values and are trainable parameters that do not require manual specification of the values. Besides, three different techniques are used for feature extraction such as Term Frequency-Inverse Document Frequency (TF-IDF), CountVectorizer, and Word2vec. Machine learning algorithms such as Decision Tree (DT), Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), K Nearest Neighbor (KNN), Random Forest (RF), and Logistic Regression (LR) were used for classifying the Bangla language dialect with this massive quantity of data. Finally, the Logistic regression showed the best performance on CountVectorizer having an accuracy of 96%. The macro-average score on TF-IDF vectorizer demonstrated that the Support Vector Machine performed best with 96% accuracy. With 89% accuracy, Random Forest provides the greatest performance on Word2Vec. For future work, they hope to work on the detection of speech from local people to classify their dialects and increase its accuracy by expanding their dataset and implementing more models and algorithms, such as LSTM, BiLSTM, as well as other neural network-based and deep learning algorithms to their dataset.

Hossain et al. [26] address several key problems in the domain of speech recognition and classification, particularly for the Bangla language. The paper has some major concerns that are being articulated as key issues of the Bangla Regional Speech recognition and classification. Detection of Synthesized Speech aimed at the possibility of identifying both natural and synthesized speech in the Bangla language, which has so far been disregarded in academic literature despite the advancement in some speech synthesis systems. Enhanced Classification Accuracy is a step where we see age in the model as an improvement for the methods that are used to isolate the speech because the human voice might change with age and influence the results in relation to speech recognition. The natural non-compact speech database used in this study comprises 30 hours of Bangla vocals and seven different regional dialects.

These dialects include Khulna, Bogra, Rangpur, Sylhet, Chittagong, Noakhali and Mymensingh. It is determined that the dataset is provided with variations of pronunciation and linguistic features in such regions; thus, it provides a strong premise for training and testing the models used in speech recognition. A stacked Convolutional Autoencoder (SCAE) helps one to get more detailed feature maps from the raw MFECs input in the given system. Multi-Label Extreme Learning Machines (MLELMs), subsequently, are employed for the purpose of generating the soft label and the hard label for the classification feature sets following the feature extraction by SCAE. The MLELMs provide adequate control of the multiple-label nature of the problem in situations where a given spoken sample needs to be labeled under one or many categories such as dialect and age. Implementing the operation of the age aspect in the class factors improved the accuracy and helped in the subsequent formation of a more accurate classification model. Age enhances the accuracy from the possibility of eighty-five percent to a possibility of ninety-five percent. Overall Classification Accuracy system tries doing the classification of the synthesized Bangla speech compared to the regional dialects where it gives an accuracy of 95 percent for the identification. Thus, it is evident that the methodology that has been proposed below is helpful in escaping large errors within the estimates.

Chowdhury et al. [27] focused on developing a machine translation (MT) system which translates a low-resource language to a high-resource language; they focused on Lambani which is a low-resource tribal language spoken in Karnataka, India and translated it to English. Since there was limited data available for Lambani and considering the syntactic similarity of other Indian languages, the researchers incorporated a transfer learning strategy. Thus, the method involved reinstating the languages of the new domain to previously developed MT models to transfer gained features and experiences into the new domain of translating from Lambani. This approach entailed procuring a dataset of Lambani text that was then used to estimate and check the MT models. The researchers experimented with the technique called “layer freezing” which implies that only specific layers, either those of the encoder or decoder network, are frozen during training to determine if the result will be improved. With the kind of parent models under consideration, the study conducted the following experiments in order to assess the efficiency of the developed English-Lambani MT system. Results showed that the English-Kannada model provided the best performance, reflecting the closer linguistic and geographical ties between Kannada and Lambani. One of the significant factors of the study was the layer freezing impact. The detailed analysis of models with changes in their architecture proved that freezing the encoder layer improved the performance of the English-Marathi-Lambani model and increased the BLEU score by 3 points. However when freezing the decoder layers, the decrease in the model’s performance was observed which implies that the decoder benefits from more data and freedom during training to learn translation patterns. The researchers recommended increasing the size of the Lambani dataset and testing strategies to add noise to the training data in order to enhance the accuracy and generalization capabilities of their model. This work shows that one can build on prior knowledge to improve access and communication with speakers of underrepresented languages and contribute to findings on transfer learning for machine translation of low resources languages such as Lambani.

Bhattacharjee et al. [28] presents a comprehensive benchmark for Bangla language generation models - a widely spoken yet underrepresented and low resource language. The authors intended to set out references and standards for assessing the NLG (Natural Language Generation) in Bangla. They tried to fill the lack of resources for Bangla NLG, which is very important for the further development of NLP in this low-resource language setting. The work of the authors proposed a new dataset for dialogue generation in Bangla. It uses other datasets for many different NLG tasks. These tasks include things like question answering, machine translation, text summarization, dialogue generation, headline generation tasks and cross-lingual summarization datasets which all together form a huge amount of data covering many fields. The authors designed a model called ‘BanglaT5’, a sequence-to-sequence Transformer language model specifically tailored for Bangla language. They then pretrained this model. This model is based on the popular Transformer architecture. It was trained on a 27.5 GB sized corpus of Bangla text data, as the authors’ aim was to create a strong baseline for the BanglaNLG benchmark. ‘BanglaT5’ demonstrated state-of-the-art performance across various NLG challenges contained within the BanglaNLG benchmark. It showed significant performance improvements over multilingual models in the same sets of tasks. BanglaT5 demonstrated state-of-the-art outcomes in several tasks, outperforming existing models by very noteworthy margins (up to 9% absolute gains on some tasks and 32% relative gains on other tasks). It demonstrated effective handling of Bangla text in diverse NLG applications. Despite all this, the study acknowledged potential limitations due to the presence of some small amount of unwanted or offensive texts in the training data, which could introduce biases. The authors suggested caution when using BanglaT5, especially in real-world applications. They then recommended further refinements and improvements of the model to include more tasks and improve its robustness and applicability.

Kuparinen and Scherrer [29] addressed the issue of precisely modeling dialectal differences in multilingual neural language models, with a special focus on neural dialect-to-standard congruence. The research uses manual transcription and normalization procedures for preparing the data for training. It uses datasets of Norwegian and Finnish dialects. This study trains a Transformer-based NMT model with six encoder and decoder layers to carry out normalization tasks. Speaker representations are gained after training the model. Then they are examined using principal component analysis (PCA) and clustering techniques. It shows dialect continuity and differences within the datasets. The data creates a relationship between the learned representations and the traditional dialect classifications. The model can detect the dialectical variations precisely. Frequently, agglomerative clustering has higher outputs than k-means clustering. A Transformer with Neural Machine Translation (NMT) architecture is used. The model handles tasks that require sequence to sequence mechanism. According to Vaswani et al. (2017), this model has encoder and decoder with 6 layers. 8 attention heads and 512-dimensional vectors are present in every layer. 100,000 steps and eight batches of gradient accumulation with a 500 tokened batch size is required for the training process. The left parameters have the default values used by OpenNMT-py toolbox. It starts with a learning rate of 4 and after training, the NMT model is examined through principal component analysis (PCA) and clustering. The dialect continuity is shown in three areas. A

distinct color stream (red, green, and blue) on a map represents each of the areas. This graphical representation extracts subtle differences and changes between dialectal areas in the datasets. To detect distinct dialect differences, trained speaker embeddings are combined with k-means clustering and hierarchical agglomerative clustering with Ward linkage. In the dataset of Finnish and Norwegian, the study’s findings indicate a significant relationship between learnt dialect representations and traditional dialect classifications. Agglomerative clustering with Ward linkage provides better results than k-means clustering in this process. PCA is used to show the dialect continuum in the datasets with discrete clusters denoting various dialectal locations. The Finnish clustering algorithms match conventional dialect areas by considering small variations within larger dialect categories. The primary dialect divisions are similarly reflected in the Norwegian clustering results. It demonstrates the model’s accuracy in capturing dialectal distinctions.

Badawi [30] analyzed the complexity of Human Language such as morphemes, syntax as well as semantic relations and realized why the task of machine translation is hard. Kurdish is indeed very low-resourced for MT as there is a dearth of linguistic resources. The need for Advanced Translation Models is what we can see in this field as there are some disadvantages in the rule-based and statistical approach such as the fact that it is not effective and time-consuming. About the dataset, Tanzil Corpus has 92,354 parallel texts of Quran translations which means that there are sufficient raw sentences for training the model. There are 2,358 parallel sentences from TED Talk subtitles in the TED corpus, and the topics vary widely. KurdNet (Kurdish WordNet), a tool including 4,663 definitions that have been translated from the Princeton WordNet to Kurdish and can be employed in different NLP tasks. Auta Corpus Consists of 100,000 preprocessed texts from a total of 229,222 parallel texts in physically aligned format, collected from various text categories and specialization fields. In order to achieve the desired results, a transformer-based neural network model which has shown high effectiveness in the MT task due to the attention system was used. The architecture includes CNN which makes use of the multi-head self-attention mechanism and position-wise feed-forward network. Secondly, an Attention Mechanism is used, where the model employs multi-head attention at three different stages. It also includes positional encodings added to the transformed input embeddings to tackle the sequential nature of the data. The proposed transformer model has been trained on the dataset and graded based on a BLEU score of 0.45, indicating high-quality translation. The training was performed using both CPU and GPU to measure the time cut. For example, the process of one epoch on a CPU was 3 hours and 37 minutes, while on a GPU: less than 5 minutes. The model tested was able to translate unseen texts to near-fidelity, although there were cases of redundant words and grammatical mistakes in some of the sentences. Such factors are due to the comparatively small sample size, and the author claimed that a larger sample would enhance the quality of translation further.

Faria et al. [2] attempted to translate dialects from different regions into standard Bangla and accurately identify the regions from where they originated. Their research tackled this critical problem by introducing the “Vashantor” dataset that focused on the translation of the Bangla language from the dialects of five different regions of Mymensingh, Sylhet, Noakhali, Barishal, and Chittagong regions. This

dataset consists of 2,500 data in Bangla, Banglish, and English each; 12,500 data for regional Bangla and Banglish dialects each; and 12,500 samples for region detection. Websites, social media platforms, and discussion boards served as the primary data collecting sources. Manual translations of regional dialects into standard Bangla from humans were collected to serve as the ground truth for ensuring the final standard Bangla output regardless of any regional dialect input. They used Fleiss’s and Cohen’s Kappa to validate the dataset. They also used cosine similarity to quantitatively assess the similarities and differences between standard Bangla language and Bangla regional dialects which showed that the linguistic similarity slope decreased from Mymensingh to Chittagong. They presented the mt5-small and BanglaT5 for the translation of dialects from different regions into standard Bangla. These machine translation tasks were evaluated using Metric for Evaluation of Translation with Explicit ORdering (METEOR), Recall-Oriented Understudy for Gisting Evaluation (ROUGE), Bilingual Evaluation Understudy (BLEU), Character Error Rate (CER), and Word Error Rate (WER). As for the region detection from where these dialects originated, they used mBERT and Bangla-bert-base models with measuring its performance through Log loss, Accuracy, Precision, Recall and F1 score. Finally, it was observed that BanglaT5 outperformed mT5 in all areas and versions. In general, the study showed that Mymensingh had the best performance measures with BLEU score of 69.06 and average word error rate of 0.1548, while Chittagong had the lowest performance measures in terms of both mT5 and BanglaT5, with the BLEU score of 36.75 and average word error rate of 0.3385. For region detection, Bangla-bert-base performed marginally better with slightly higher accuracy of 85.86% and a lower log loss compared to the 84.36% accuracy of mBERT.

Islam et al. [31] analyzed that transcription of the Bangla language phonemes into IPA is rather problematic, mainly owing to the intricate phonological and phonotactic structure of this language and much more because of the phonetic dependencies on context and phonetic mutations of consonants. The several varieties in the other regions of Bangladesh have made this task more complex with the variation in the phonemes and there is no standard form of spelling as well. This is even more complex since the test data may contain certain words that the model has never seen during the training process and this is a hard challenge to any word-based model. To overcome the issues, the paper suggested the District Guided Tokens (DGT) technique which introduced a district token before each input text. This assisted the model in establishing regional origin of the text and it is able to capture the phonetic nature of each region. Similarly, to avoid the OOV word problem, the authors employed the Byte-level Transformer Model (ByT5). Unlike other models that operate on the word level, ByT5 operates on the byte level and thus has a better chance of pointing out hitherto unseen words during a training process. By adopting DGT and byte-level processing, the fundamental objective of enhancing transcription accuracy was realized. This has been evidenced by the reduced WER scores that the ByT5 model yielded compared to the other models such as mT5, BanglaT5 and umT5. The corpus consists of Bengali texts and their corresponding IPA transcriptions have been collected from six districts of Bangladesh. The dataset is divided and the testing set contains many out-of-base vocabulary (OOV) words. It is used with transformer-based models including the byte-level ByT5 model where characters are treated as bytes, not words. It is especially useful in the case of OOV

words because it allowed for their improved management. Naturally, the ByT5 model managed to outperform all the other models because of the fact that the model is excellent for handling OOV words; however, the authors agreed that the problem of OOV words still persists (about 46.97%) and it is clear that there still is the scope of enhancing the performance of the classifiers for special cases like this.

Samin et al. [3] identify the challenge of standardizing regional Bangla dialects, presenting a major hurdle in communication, mainly for formal, educational, or technological purposes. Despite Bangla being the fifth most spoken language globally, research into dialectal speech processing has been limited due to the lack of comprehensive datasets and the complexity of dialect variations. The Noakhali dialect is specifically addressed in this work by presenting an AI-based end-to-end pipeline for dialect speech conversion into standardized Bangla speech. To help with this work, the authors created the Noakhali Dialect Dataset (NDD). It has 10 hours of audio from 24 native speakers and includes dialect text along with standard Bangla translations. After breaking it down, they ended up with 7,200 samples. They cleaned up the audio to reduce noise and split it into 5-second clips to make it easier for training models. The setup includes three main parts: Whisper ASR to turn dialect speech into dialect text, BanglaT5 and other multilingual models like mBART50, mT5, and IndicBART for translating from dialect to standard Bangla, and AlignTTS, a text-to-speech model that creates standard Bangla speech from the translated text. The tests showed that Whisper-large V2 did really well when it was fine-tuned, hitting a Character Error Rate (CER) of 0.8% and a Word Error Rate (WER) of 1.5% for automatic speech recognition (ASR). For translation, the fine-tuned BanglaT5 scored the highest with a BLEU score of 41.6, doing better than other multilingual models. The authors point out that this is the first approach like this for Bangla dialects. Right now, the system mainly works with the Noakhali dialect, but they plan to broaden the dataset to cover more Bangla dialects and improve support for more languages. Overall, the BanglaDialecto system is a solid step towards making language technology more inclusive and helping more Bangla speakers connect digitally.

Chapter 3

Work Plan

We have three main phases for our thesis, starting with Pre-Thesis 1 followed by Pre-Thesis 2 and lastly the final defense. Our Pre-Thesis 1 started in Spring 2024 and lasted for around 4 months, the whole spring semester to be precise. This phase had formation of our thesis team, topic and supervisor selection of thesis domain, topic and supervisor, summarisation of related researches and works and finally the submission of the report. Pre-Thesis 2 was conducted in the Summer 2024 semester which also had a timeline of 4 months. In this phase, we worked on analyzing the data and building the translation models that we want to implement. After that, we submitted a report. Finally, in Spring 2025, we analyzed our results and compared them with existing works to prove the depth and accuracy of our work. Here, we completed our paper and will be facing our Thesis Defense in front of the qualified panels. The Gantt Chart is a visual representation of the whole Thesis timeline. The three main phases are shown in different colors for more clarity.

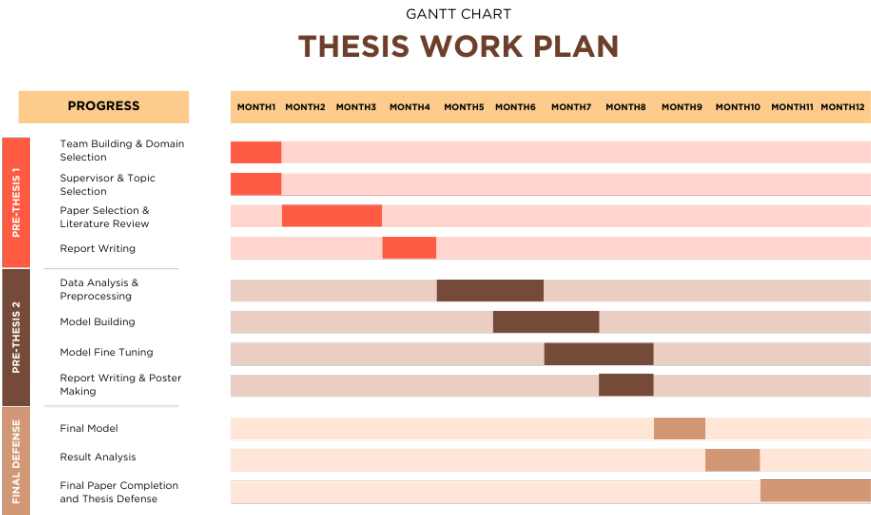


Figure 3.1: Workplan Gantt Chart

Chapter 4

Methodology

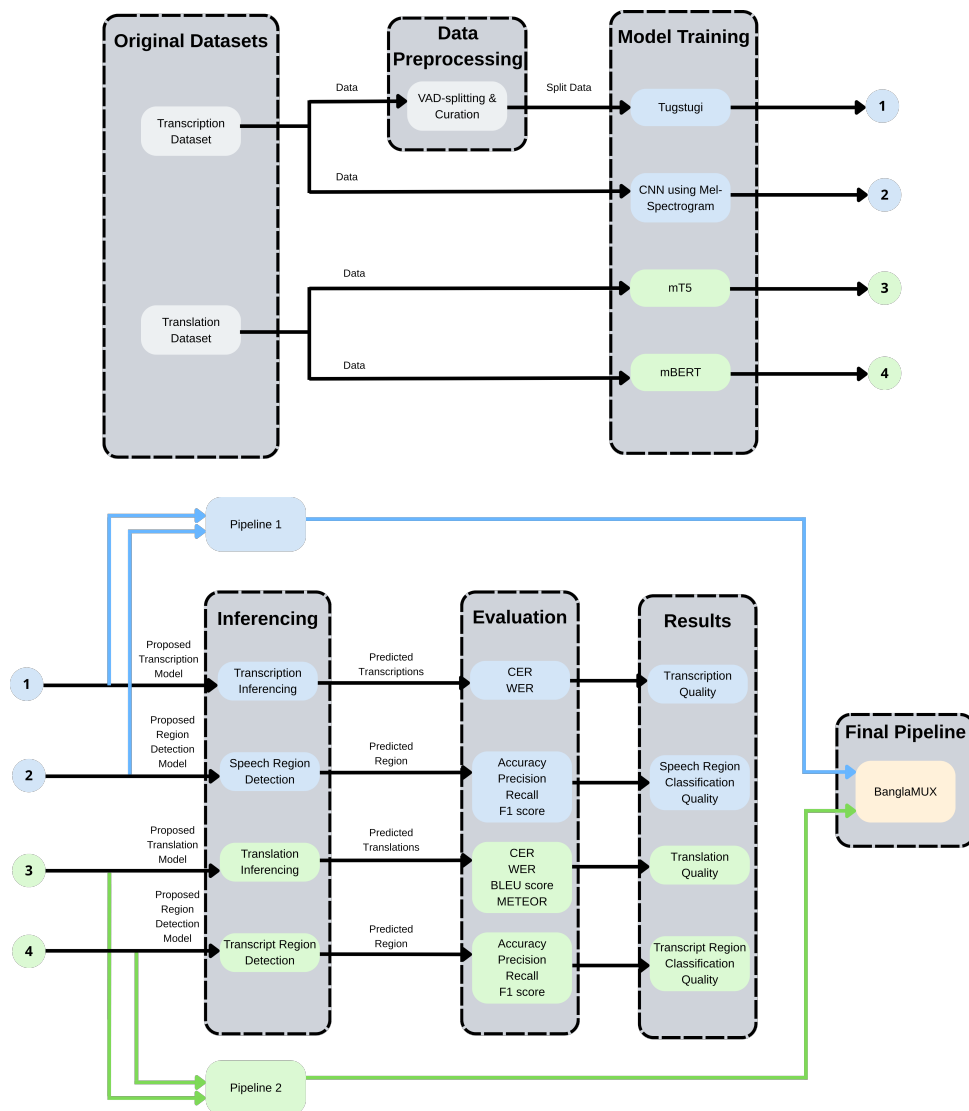


Figure 4.1: Methodology for the pipelines of BanglaMUX

4.1 Overview of the Proposed System

The system consists of two pipelines: a transcription pipeline for capturing the dialectal speech and converting it into text, and a translation pipeline for addressing the dialectal text. The dual-pipeline with region detection supports routing to region-specific trained models, resulting in optimal performance when faced with different Bengali dialects. The figure 4.1 proposes a dual-pipeline system for both transcribing and translating regional dialects in Bangla.

4.2 Data Collection and Preparation

4.2.1 Original Datasets

We primarily relied on two separate data sets; the Bengali AI Regional ASR competition dataset [32] and the Vashantor dataset [2]. The competition data represented normalized Bengali speech and text examples, Vashantor contained various regional dialect distinctions. The transcription data was made up of audio elements annotated with transcriptions of different Bengali dialects. The translation data represented parallel text corpora with cross-dialect translations.

4.2.2 Data Preprocessing

Voice Activity Detection (VAD) was used for audio pre-processing to detect pauses and silences in recordings, allowing us to identify and automatically split large audio files into smaller audio files consisting only of speech. Using VAD removed background noise and silence to produce speech-only audio segments in small, manageable segments of transcribed audio. Text data was normalized, and quality controlled, in addition to validation of dialect classifying to ensure consistency of training data with all regional variations.

4.3 Model Architecture and Training

4.3.1 Speech Recognition Model (Tugstugi)

The Tugstugi-Whisper-medium-small model refers to a fine-tuned variant of OpenAI’s Whisper model, specifically adapted for the Bangla language. We trained separate Tugstugi models for each target region, allowing them to tailor to the unique acoustic and linguistic features of the associated dialects. Each model was fine-tuned with distinct dialect training data so we could maximize each model’s potential to recognize speech patterns local to that region, including pronunciation variation and vocabulary.

4.3.2 Region Detection Model (CNN with Mel Spectrogram)

The CNN-based region detection system employs Mel Spectrogram representations of audio inputs to identify the dialectal origins of the audio inputs. The architecture processes the data by convolutional layers to extract spatial hierarchies of features from the spectral data, into classification layers which output a probability distribution over the various regional dialect classes.

4.3.3 Translation Model (mT5)

Models for multiple regions were fine-tuned mT5 models for the translation of text in specific regional dialects to standard Bengali. Each model is able to uniquely leverage the syntactic structure and vocabulary of their region.

4.3.4 Region Detection Model (mBERT)

Acting as a text-based region detection system, this model deciphered linguistic trends, determining source dialects, before routing text to the appropriate mT5 translation models.

4.4 Pipeline Implementation

4.4.1 Pipeline 1: Speech-to-Text with Region Detection

The transcription pipeline takes audio input as WAV files and processes it to two levels of processing. At the initial level, a CNN region detection model works on Mel Spectrograms to pick one from among dialects considered to be the origin. According to classification results, the system would send the audio to the Tugstugi model for that particular region trained with that particular dialect.

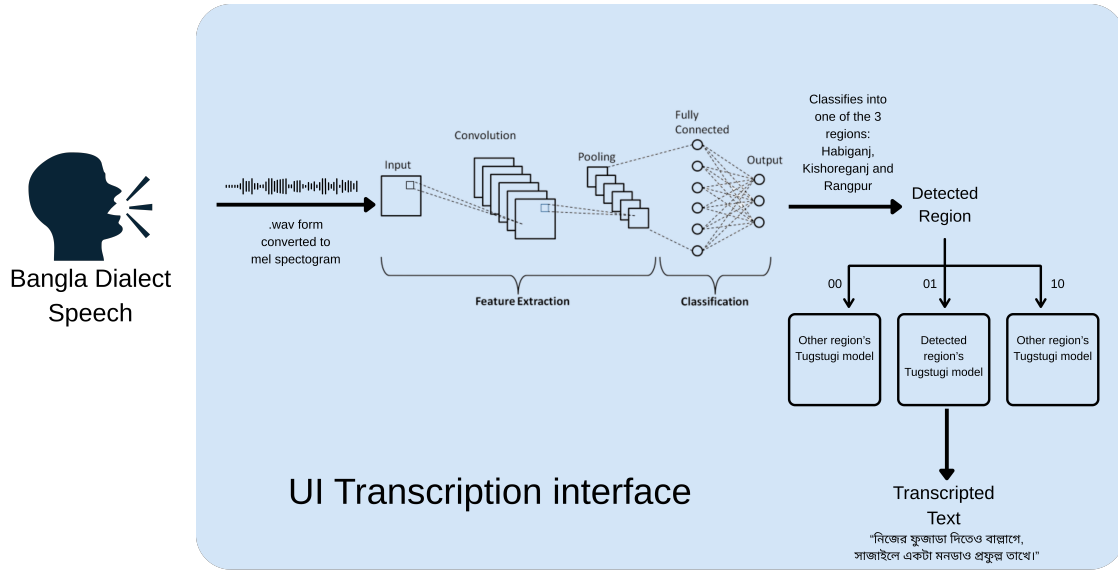


Figure 4.2: Pipeline for Transcription

4.4.2 Pipeline 2: Translation with Region Classification

This pipeline would first take the textual inputs and classify their dialect using mBERT, and then mT5 models specific to regions would translate them. mBERT studies the set of linguistic patterns and vocabulary to identify a source dialect and then routes the text to one of a series of mT5 models trained in that dialect.

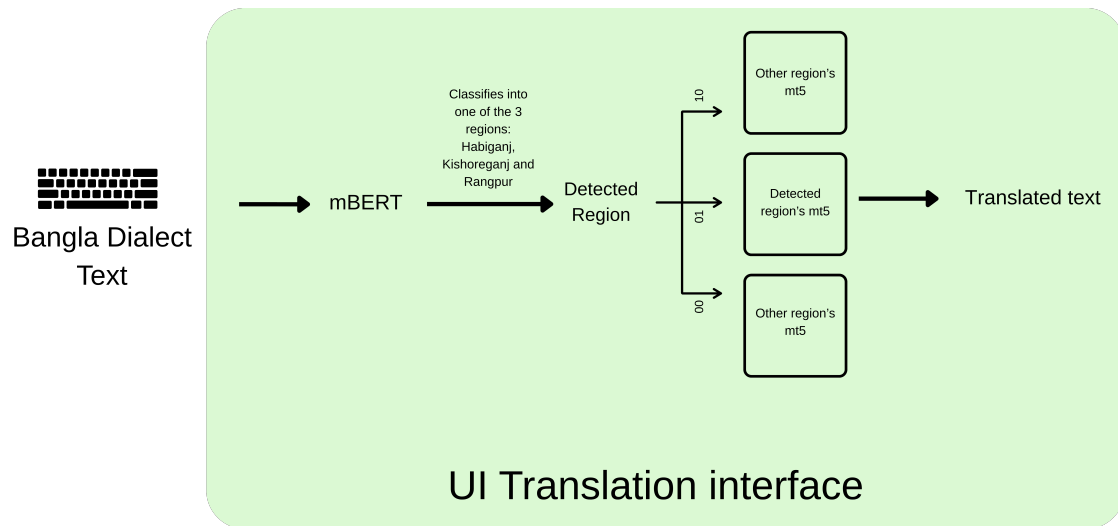


Figure 4.3: Pipeline for Translation

4.5 System Integration

4.5.1 BanglaMUX: Unified Pipeline

BanglaMUX combines the two pipelines into a single platform whereby the user can specify input type and direct it to the appropriate processing pipeline. Hence, the system remains modular while making efficient use of resources on both audio and text processing tasks.

4.5.2 End-to-End Workflow

The system begins with the user input, which is fed into transcription or translation pipelines. Each pipeline executes specific workflows that output transcribed or translated text, maintaining semantic content but adapting it to target formats or dialects.

Chapter 5

Dataset

5.1 Analysis of Dataset (Translation)

Our work involves two major data sets: the Vashantor Dataset and the Bengali.AI Dataset, both were obtained from external sources.

5.1.1 Vashantor Dataset

The Vashantor Dataset was found in a research paper on a similar topic. Their efforts proved fruitful in compilation of the different set of sentences in Bengali in the different dialects like Barishal, Mymensingh, Chittagong, Sylhet, Noakhali. This dataset was collected by web scraping which is a technique of collecting text data from the World Wide Web including web sites, social networking sites, newsgroups and forums etc. The data collected aimed at capturing natural language as seen in everyday dialogues or interactions of people.

5.1.2 Bengali.AI Dataset

The Bengali.AI Dataset was provided to us as private; no research has ever been done on it. Bengali.AI is an initiative for the collection of speech data across various dialects in Bengali, hence including Rangpur, Habiganj, and Kishoreganj among many others. This dataset was collected by recording spoken Bengali from different speakers, capturing the unique features of regional speech. These recorded speech utterances were then transcribed into text to ensure suitability for dialect variability studies. Both these datasets are major contributors towards the analysis of linguistic diversity across diverse regions and help in developing transcription and translation models.

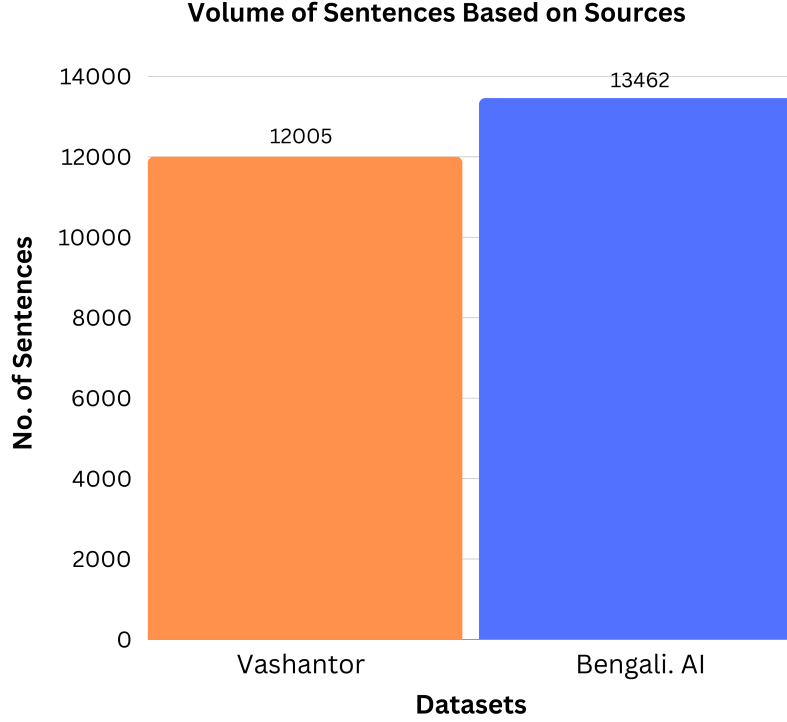


Figure 5.1: Volume of Sentences

From the figure 5.1, the total number of sentences for both datasets can be obtained. The Vashantor Dataset contains 12,005 sentences which were collected from five different dialects, namely Barishal, Mymensingh, Chittagong, Sylhet and Noakhali. The Bengali.AI Dataset contains 13,462 sentences which are representative of dialects from Rangpur, Habiganj, and Kishoreganj. As shown by the figure, the overall volume of sentences in each dataset makes them quite substantial and hence representative of dialectal variations. With the sentence count of the Vashantor dataset being somewhat lower than that of Bengali.AI, it indicates a narrower focus on a variety of regional dialects. In contrast, the larger volume allows the Bengali.AI dataset to be more representative of dialectical nuances within its selected regions.

5.1.3 Sentence Length Distribution

We used the Sentence Length Distribution technique to understand how sentence lengths vary across different datasets. We included the Vashantor and Bengali.AI dataset. This helped us see patterns in how sentences are structured. When we first looked at the Vashantor dataset, we noticed that most sentences are between 4 to 7 words long. Majority of the sentences had 6 words. It is interesting because this tells us that the dataset tends to have shorter sentences. We did not stop there. We analyzed datasets of Vashantor from different regions- such as Mymensingh, Barishal, Sylhet, Noakhali, and Chittagong, and found something similar. In all these datasets, sentences with 6 words were the most common. To give a clearer preview: in the Mymensingh dataset, there were 477 sentences with 6 words, Barishal had 491, Sylhet also had 477, Noakhali had 459, and Chittagong had 453. We visualized these numbers using bar charts to make the comparisons easier to understand.

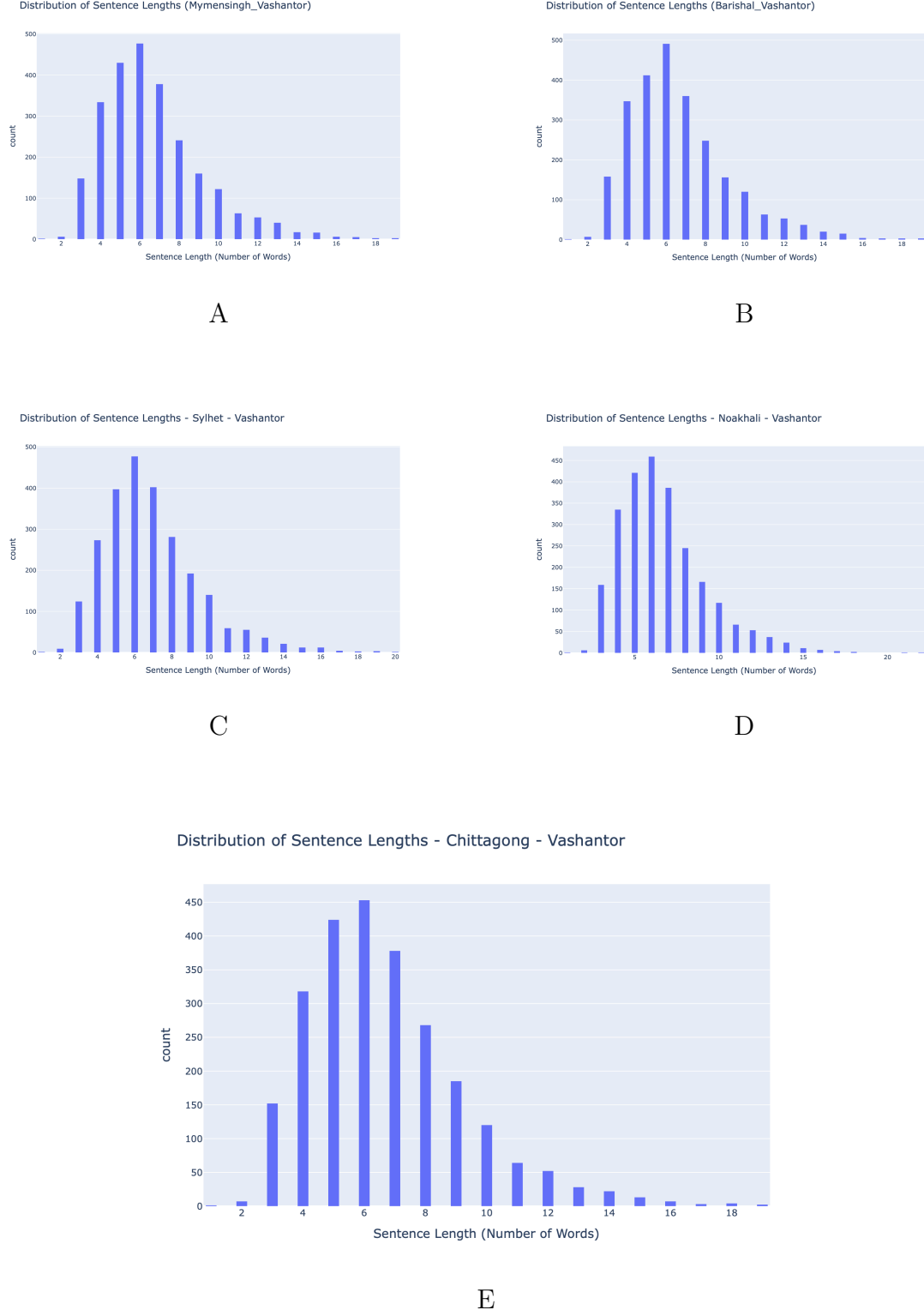


Figure 5.2: Sentence Length Distribution of Vashantor Datasets - A. Mymensingh, B. Barishal, C. Sylhet, D. Noakhali, E. Chittagong

For the Bengali.AI dataset, we analyzed regions like Kishoreganj, Habiganj, and Rangpur. Kishoreganj people usually use lesser words as most of the sentences had only 2-4 words. Habiganj had 2-6 words mostly in their sentences. Similarly,

Rangpur had also shown 2-5 words in most of their sentences. The most common sentence lengths for each of the regions are accordingly 1217, 306 and 1115.

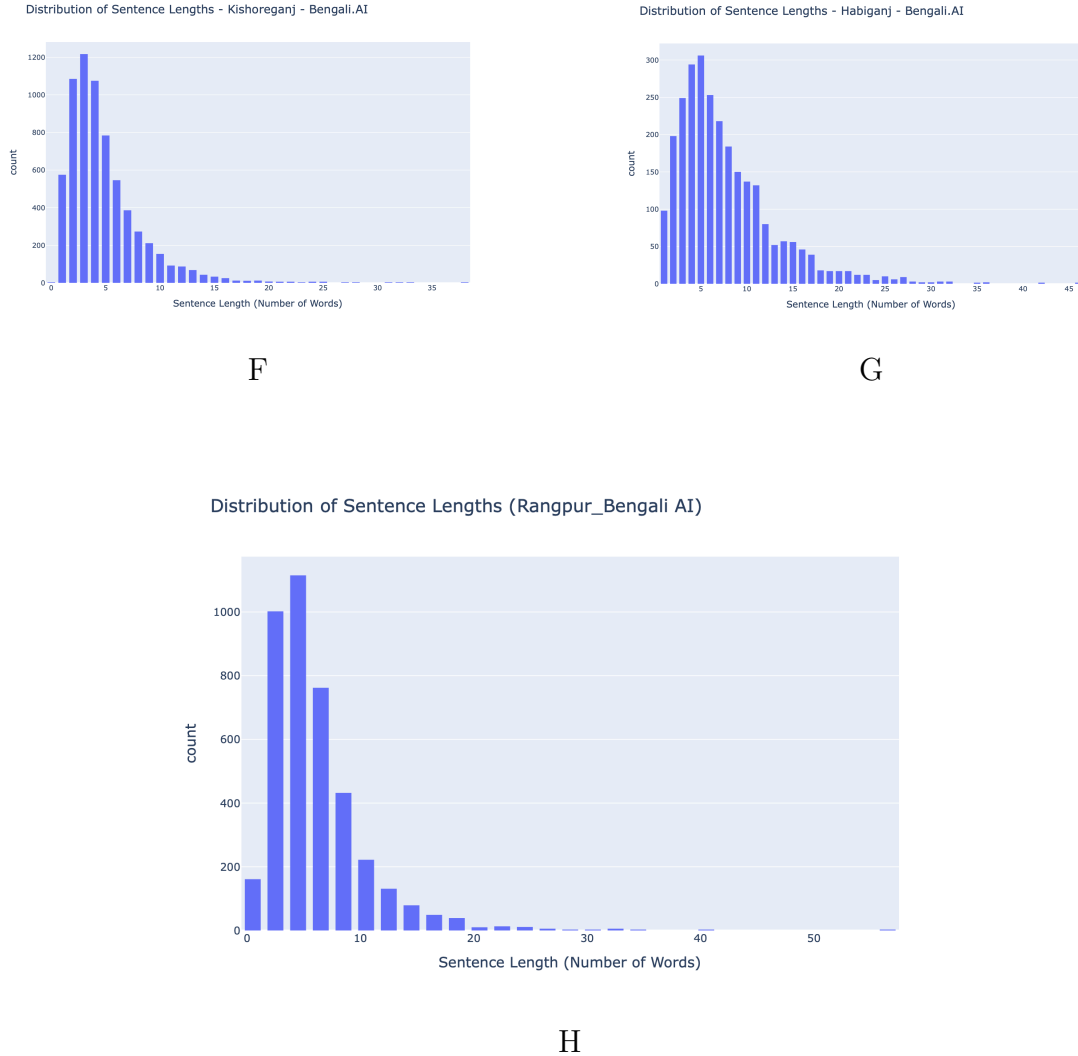


Figure 5.3: Sentence Length Distribution of Bengali AI Datasets - F. Kishoreganj, G. Habiganj, H. Rangpur

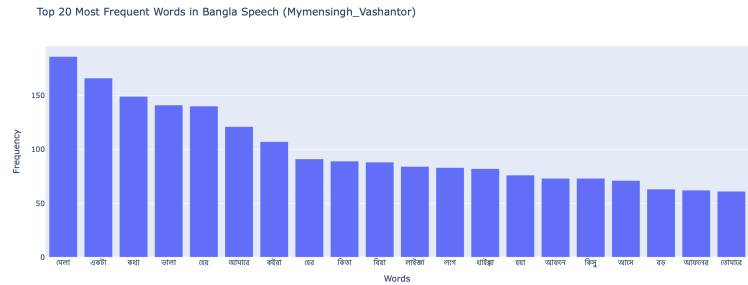
5.1.4 Most Frequent Words

We have also conducted the most frequent word analysis so that we can understand which region uses which words the most. It can give us a pattern of identifying regions through sentences and most used words will be taken into consideration. The top 5 most frequent words of different datasets are:

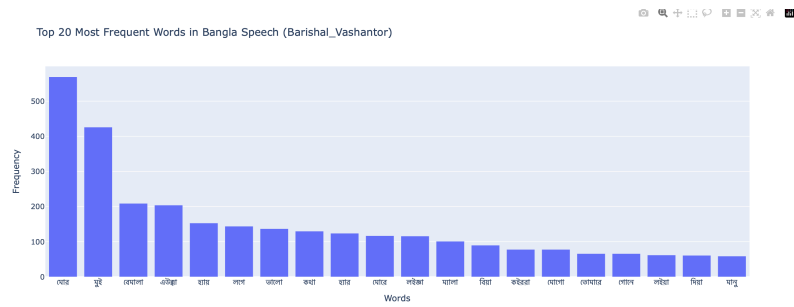
Mymensingh (Vashantor) - মেলা (186), একটা (166), কথা (149), ভালা (141) and হেয়ে (140).
 Barishal (Vashantor) - মোর (569), মুই (426), বেমালা (209), এউক্কা (204) and হ্যায় (153).
 Sylhet (Vashantor) - মোর (205), মুই (201), কিতা (174), ভালা (147) and লগে (144).
 Noakhali (Vashantor) - লগে (139), ভালা (133), আরে (123), কতা (108) and লাই (100).
 Chittagong (Vashantor) - ন (196), নো (188), আঁই (165), আঁর (161) and হতা (141).
 Kishoreganj (Bengali.AI) - ফরে (226), অহন (220), এইডা (204), দিয়া (200) and অইছে (188).

Habiganj (Bengali.AI) - একটা (227), কিতা (220), তে (163), লইয়া (109) and মানে (86).
 Rangpur (Bengali.AI) - এলা (225), ভালো (172), তোর (159), মুই (150) and ভাই (143).

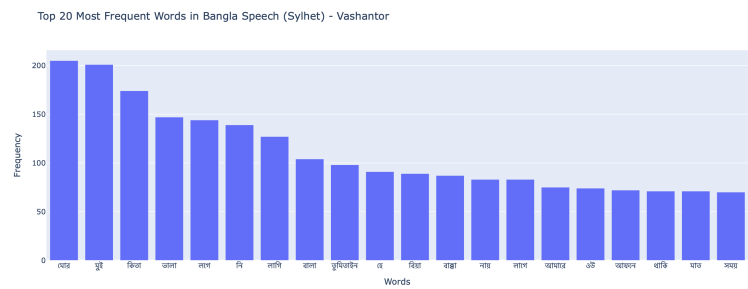
The graphs 5.4 illustrate better.



A

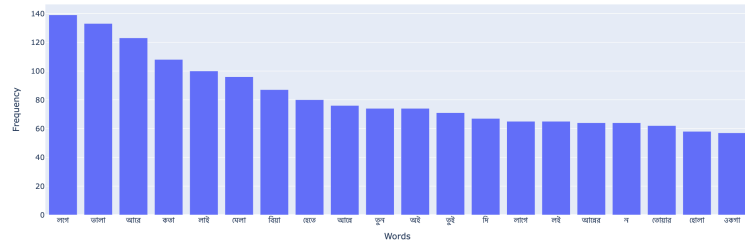


B



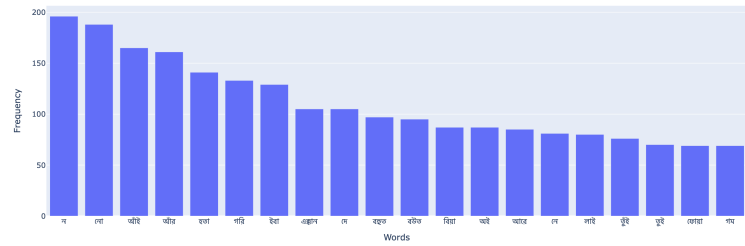
C

Top 20 Most Frequent Words in Bangla Speech (Noakhali) - Vashantor



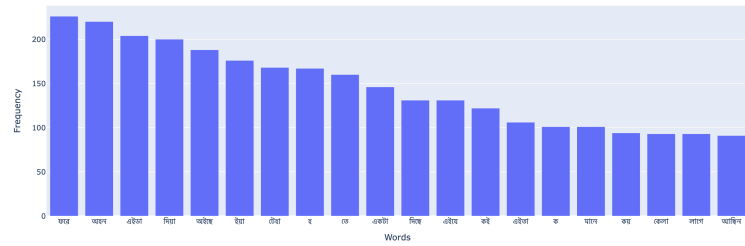
D

Top 20 Most Frequent Words in Bangla Speech (Chittagong) - Vashantor



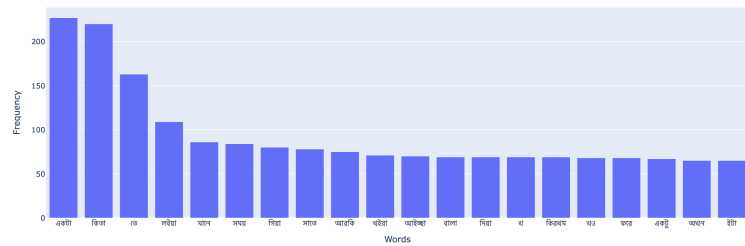
E

Top 20 Most Frequent Words in Bangla Speech - Kishoreganj - Bengali.AI



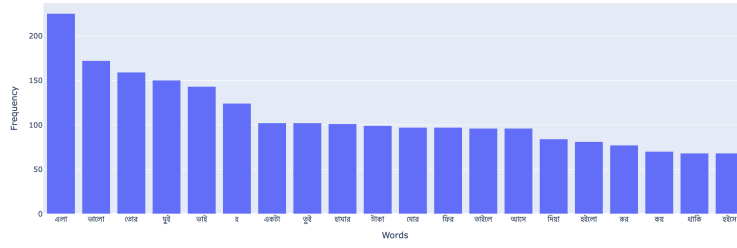
F

Top 20 Most Frequent Words in Bangla Speech - Habiganj - Bengali.AI



G

Top 20 Most Frequent Words in Bangla Speech (Rangpur_Bengali AI)



H

Figure 5.4: Most Frequent Words of Different Regions - A. Mymensingh, B. Barishal, C. Sylhet, D. Noakhali, E. Chittagong, F. Kishoreganj, G. Habiganj, H. Rangpur

5.1.5 Word Cloud

We know that the boldness and intensity of a word in word cloud means the frequency level of that particular word in the dataset. So, for even clearer visualization, we have also generated the word clouds for the 8 different datasets. The word clouds 5.5 are given below.



A



B



C



D



E



F



G



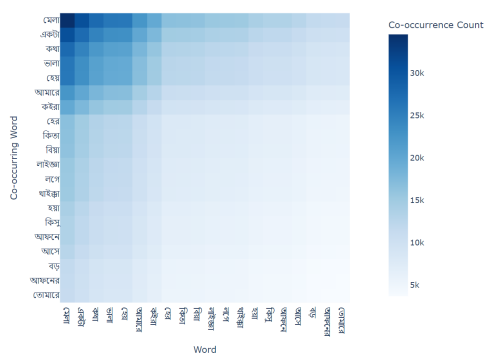
H

Figure 5.5: Word Cloud Comparison - A. Mymensingh, B. Barishal, C. Sylhet, D. Noakhali, E. Chittagong, F. Kishoreganj, G. Habiganj, H. Rangpur

5.1.6 Word Co-occurrence Matrix

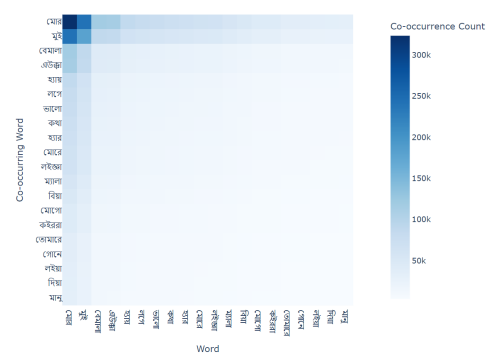
We also visualized the data from every source and dataset using Word Co-occurrence Matrices, a way of figuring out how often a pair of unique words appear together in sentences, helping us understand if they have a contextual relation or not. Noteworthy co-occurrences from each of the sets are each as follows:

Word Co-occurrence Matrix for Top Words in Bangla Speech (Mymensingh_Vashantor)



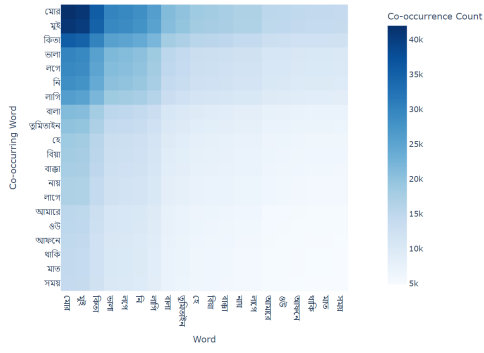
A

Word Co-occurrence Matrix for Top Words in Bangla Speech (Barishal_Vashantor)



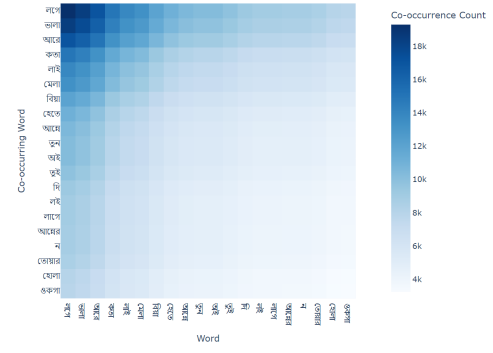
B

Word Co-occurrence Matrix for Top Words in Bangla Speech (Sylhet) - Vashantor



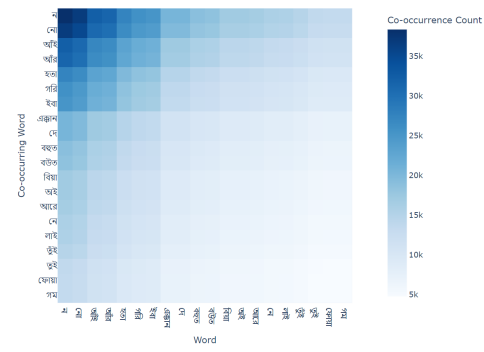
C

Word Co-occurrence Matrix for Top Words in Bangla Speech (Noakhali) - Vashantor



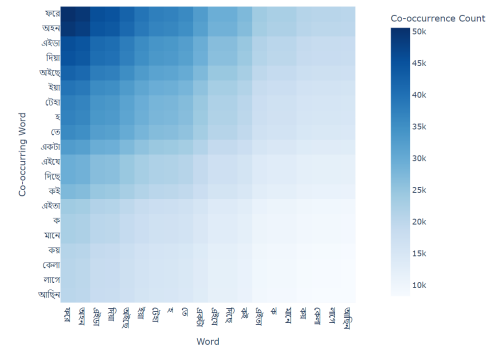
D

Word Co-occurrence Matrix for Top Words in Bangla Speech (Chittagong) - Vashantor



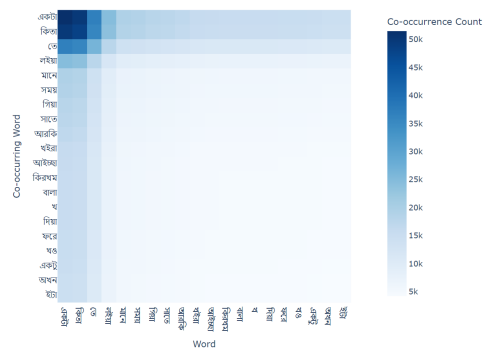
E

Word Co-occurrence Matrix for Top Words in Bangla Speech - Kishoreganj - Bengali.AI



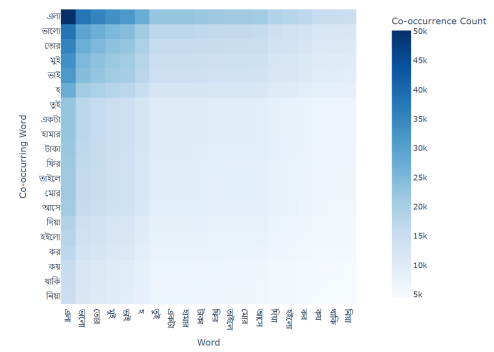
F

Word Co-occurrence Matrix for Top Words in Bangla Speech - Habiganj - Bengali.AI



G

Word Co-occurrence Matrix for Top Words in Bangla Speech (Rangpur_Bengali AI)



H

Figure 5.6: Word Co-occurrence Matrices of different sets of locations based on source - A. Mymensingh, B. Barishal, C. Sylhet, D. Noakhali, E. Chittagong, F. Kishoreganj, G. Habiganj, H. Rangpur

5.1.7 Unique Vocabulary

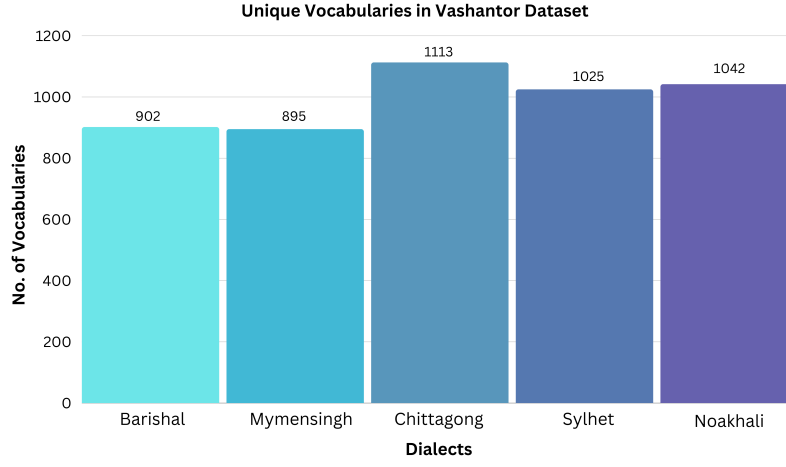


Figure 5.7: Unique Vocabulary - Vashantor

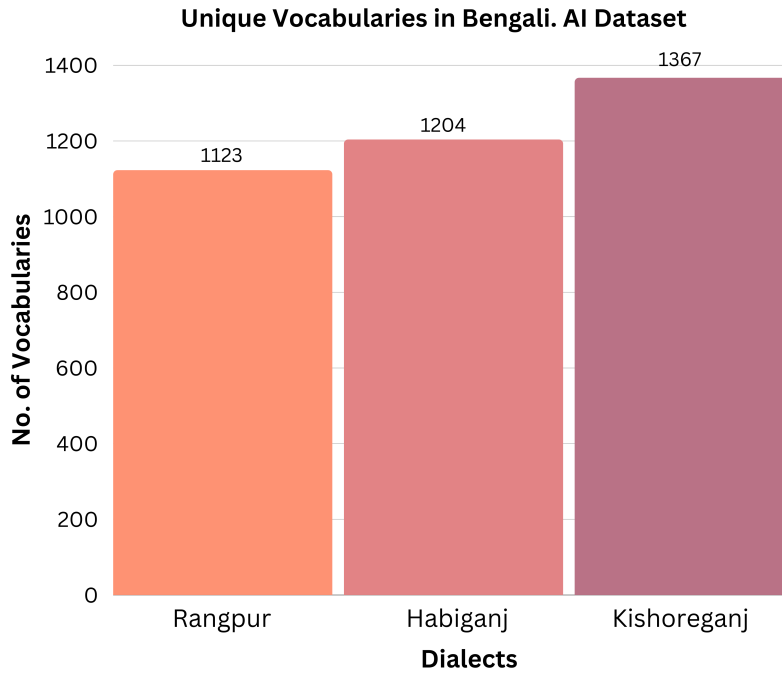


Figure 5.8: Unique Vocabulary - Bengali.AI

The figures [5.7] and [5.8] contain unique vocabularies within the two sets of data. Unique vocabularies are words or lexical items which occur in a particular dialect without repetition, within the corpus being analyzed. These words show the linguistic richness and diversity of dialects, with an insight into the regional variation that exists in vocabulary usage. Figure [5.7] shows the unique vocabularies of the Vashantor Dataset. It is clear that Chittagong contains the highest lexical diversity with 1113 unique words, Noakhali contains 1042, and Sylhet has 1025 unique terms. Unique vocabularies of Barishal and Mymensingh are 902 and 895 words

respectively, which is a lot fewer in these dialects. For this purpose, Figure [5.8], represents the case of the Bengali.AI Dataset, placing Kishoreganj at 1367 unique words, signifying being rich in vocabulary among the analyzed dialects, followed by Habiganj with 1204, and Rangpur with 1123 unique terms. All these figures put together show that Bengali.AI Dataset represents generally a wider range of dialectal vocabulary compared to Vashantor Dataset, which is likely to result in more nuanced transcription and translation tasks.

5.1.8 Rare Word Ratio

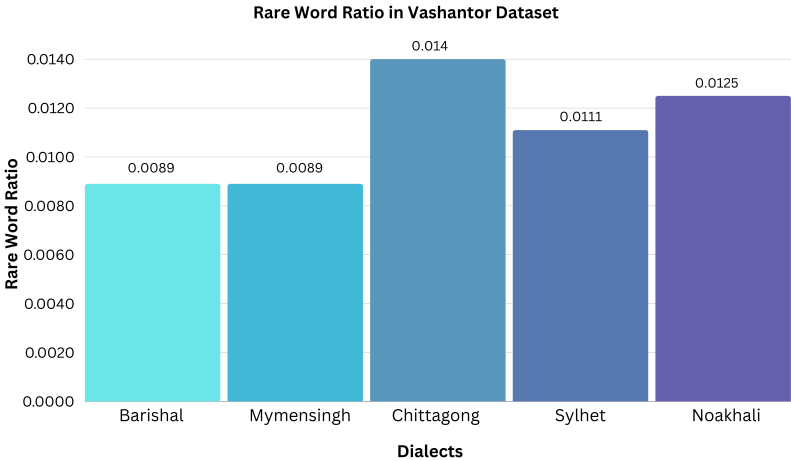


Figure 5.9: Rare Word Ratio - Vashantor

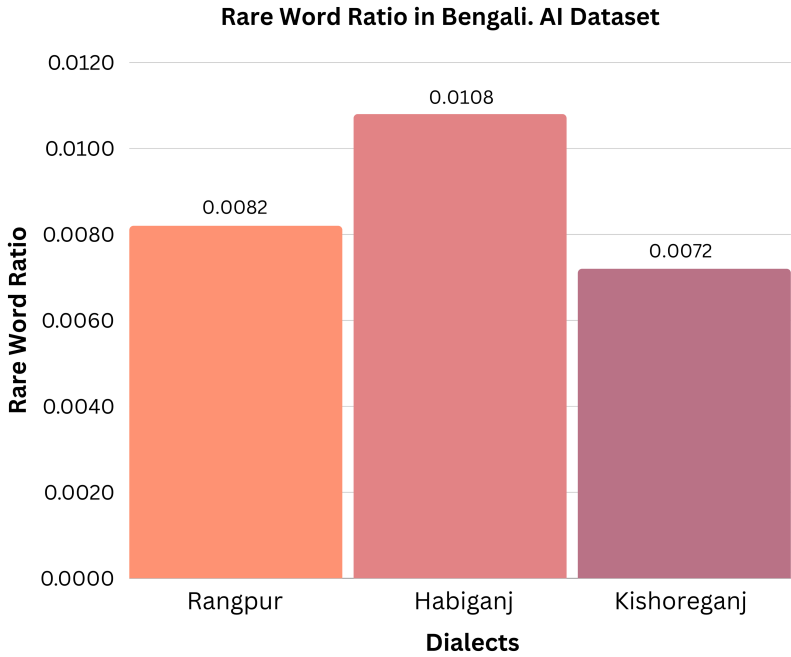


Figure 5.10: Rare Word Ratio - Bengali.AI

The figures [5.9] and [5.10] plot the rare word ratio in both the datasets. The rare word ratio signifies that the words in the dataset occur very infrequently. It draws on lexical richness and variability from a certain dialect. A higher rare word ratio indicates high presence of less common terms that can reflect uniqueness of the dialect, or its divergence from more standardized forms. Figure [5.9] indicates the ratio of rare words in the Vashantor Dataset. From Figure [5.9] it is observed that the word ratio for Chittagong is 0.014, which is the highest among these five, thus it has a richer set of uncommon terms, followed by Noakhali with 0.0125 and Sylhet with 0.0111. The other two dialects, Barishal and Mymensingh, have only 0.0089, and thus, a vocabulary relatively more standard. On the other hand, Figure [5.10], illustrating the Bengali.AI Dataset, shows that the maximum of Habiganj takes a value of 0.0108, while Rangpur and Kishoreganj take a lower value of 0.0082 and 0.0072 respectively. Comparing these figures gives an idea that the Vashantor Dataset has more varied rare terms in general, which may be concentrated with more localized words in the case of the Chittagong dialect. While rare word usages in the Bengali.AI Dataset are comparatively less varied. This finding suggests that not only the dataset and dialect choice matter a lot in availability of rare words, but further influences the transcription models across variable speech patterns.

5.1.9 Dataset Preprocessing

The Vashantor dataset was already split into texts which consists of 75% data for training, 15% data for testing and 10% data for validation. On the other hand, we first shuffled the dataset from Bengali.AI of each region and then manually split the text data into 80% for training, 10% for testing and 10% for validation. We also dropped any null values present and all the other columns except the ones that are required for training. Removing unnecessary columns acts as a feature selection during model training and aids in improving the performance of the model and reducing computational complexity with the reduction in data noise.

For translation tasks, we have chosen the input text from both the datasets which contain regional dialects. The human generated translations of regional languages to standard Bangla act as the target label. The regional dialects are different from region to region but the Bangla translations are the same for all the regions. Regardless of the regional dialect of the source, these translations are global bridges that guarantee the end result to be in standard Bangla. For text preprocessing, we utilized a normalizer to clean the data before tokenization. Then, we tokenized both input and target texts converting them into tokens so that the model can process it for training. The variable-length input and output sequences are padded to the maximum length in each batch during training. Moreover, the training data is shuffled so that the model does not learn from the order of the data. However, the test and validation data are not shuffled to ensure consistency in the evaluation. Finally each batch of the test data set is sequentially processed by the model to generate the translations and the generated IDs are decoded back into the human readable text.

As for region detection tasks, we have chosen the input text and region name from

both Vashantor and Bengali.AI datasets separately. For each dataset, we first concatenated all the text data for all the regions having only those two columns and then shuffled the training data so that the model does not learn from the order of the data. For text preprocessing, we mapped the categorical region names to integer labels as the models require numerical input rather than categorical data. We used Bangla BERT tokenizer for converting the text data into tokens for the models to understand. Finally, the text from each regional dialect can be predicted by the models to which region it belongs.

5.2 Analysis of Dataset (Transcription)

The dataset for transcription we worked on had been prepared by Bengali AI for a competition. The dataset has different regions but we worked on 3 regions - “Kishoreganj, Habiganj and Rangpur”. There are a total of 4517 data points in the entire dataset.

5.2.1 Bengali.AI Dataset

This dataset was collected by recording spoken Bengali from different speakers, capturing the unique features of regional speech. These recorded speech utterances were then transcribed into text to ensure suitability for dialect variability studies. Both these datasets are major contributors towards the analysis of linguistic diversity across diverse regions and help in developing transcription and translation models.

5.2.2 Audio VS Transcription Length

All three regions show a strong positive correlation between audio length and the length of transcription, which is unsurprising since longer audio recordings will naturally contain more spoken content needing longer transcriptions. However, the relationship had quite a bit of variation showing that patterns of speech, pace, and the density of content range considerably across each region.

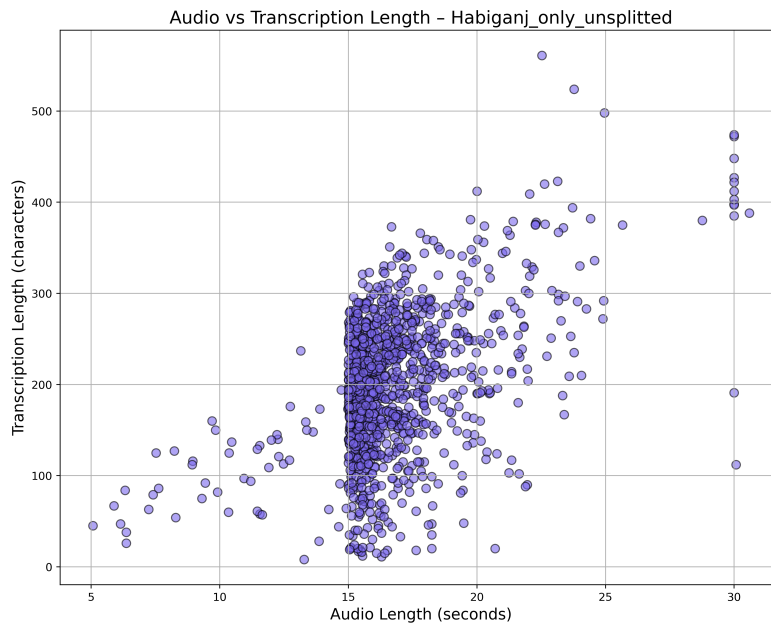


Figure 5.11: Audio vs Transcription Length of Habiganj

Habiganj

The figure [5.11] displays the most dense data distributions, with the most data points being seen from the 12-18 seconds of audio length and 100-300 characters of transcription. This region seems to contain the most linear and predictable relationship.

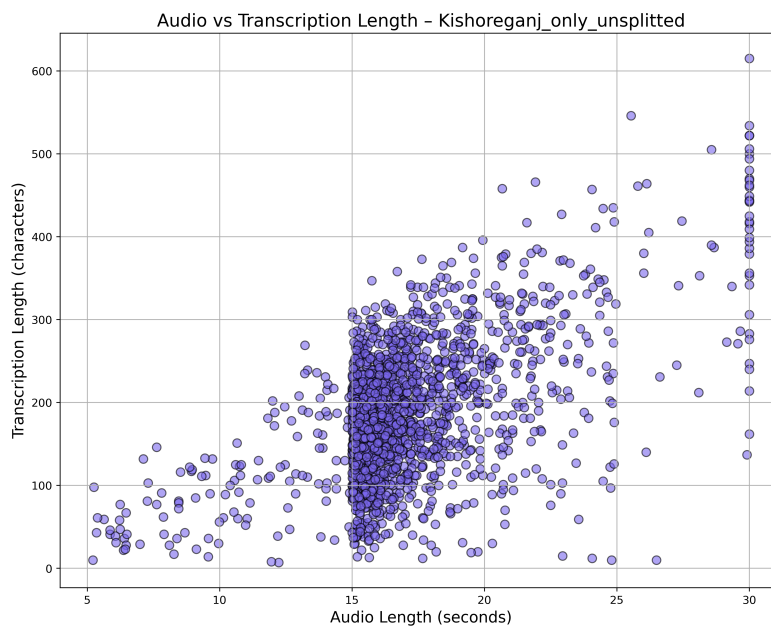


Figure 5.12: Audio vs Transcription Length of Kishoreganj

Kishoreganj

In the figure [5.12] seems to follow a very similar pattern as Habiganj but with a little more scatter further into the longer audio length ranges (20+ sec). There is significant vertical clustering around the 30-sec mark indicating that some recordings may have reached a maximum duration limit.

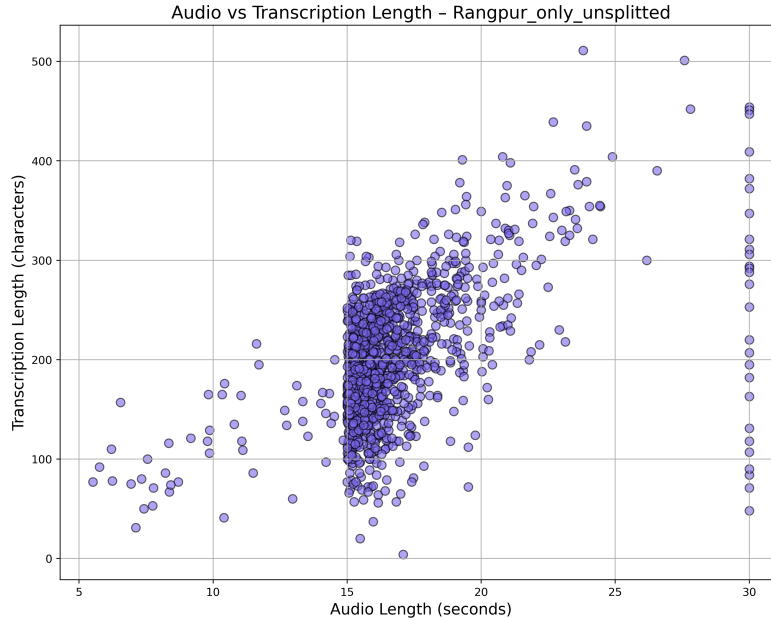


Figure 5.13: Audio vs Transcription Length of Rangpur

Rangpur

[5.13] shows an even more dispersed distribution, including data points with larger variance in transcription length within the same audio duration. This suggests that there could be more variety in speaking patterns or types of content.

5.2.3 Waveform and Spectrograms

Habiganj Audio Analysis

The Habiganj waveform [5.21] is consistently high-amplitude speech and shows significant amplitude variation (from -1.0 to +1.0). The waveform shows a significant silent gap in the 7-8 second range which could be a pause or word boundary. The signal maintains some energy during the majority of the recording; there are dense oscillations throughout the signal suggesting speech activity. The spectrogram [5.21] showed extensive frequency information in the 0-2048 Hz frequency range, which had the highest energy (yellow/orange regions) in the lower frequency spectra (0-1024 Hz) which to be attributed to fundamental pitch and lower-order harmonics that are expected from human speech. The spectral energy display maintains continuous patterns with some temporal representation which follow the amplitude changes

seen in the waveform.

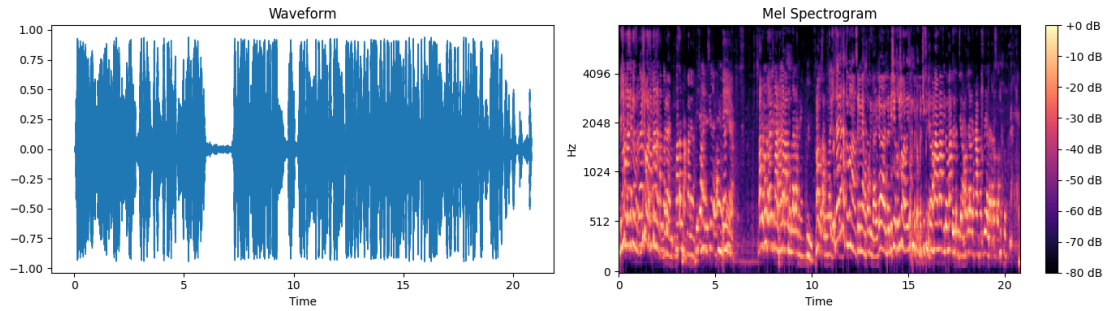


Figure 5.14: Waveform and Spectrogram of a sample from Habiganj

Kishoreganj Audio Analysis

The amplitude of the recorded signal from Kishoreganj [5.22] is substantially lower compared to Habiganj, with most activity within the ± 0.2 range. However, there is a sharp amplitude peak between 12-13 seconds with amplitude of about ± 0.8 before quiet periods. The signal appears to sound a lot more silent with maybe low-level background noise. The spectrogram [5.22] captures the characteristics of the waveform, which shows a relatively quiet typical look (the darker purple areas) framed with the major bright vertical band that corresponds to the amplitude bounce in the waveform. The frequency distribution does not show as consistent of an energy distribution as Habiganj, and has more sparse spectral content.

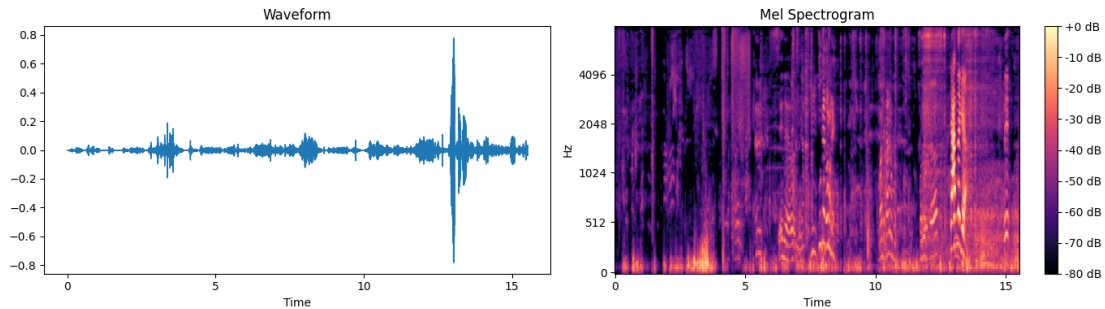


Figure 5.15: Waveform and Spectrogram of a sample from Kishoreganj

Rangpur Audio Analysis

The Rangpur [5.23] recording of 30 seconds has moderate amplitude fluctuations with speech activity at a relatively steady level. The amplitude generally remains within a ± 0.5 range (mostly $+0.5$) with occasional peaks, indicating the speaker's speech is conversational and varies naturally in prosody. The spectrogram features [5.23] the most detailed and complicated frequency patterns of the three regions. There are unmistakably different patterns of vertical striations and formant structures located across the time axis indicating rich phonetic content. The audio energy

is distributed fairly evenly across the frequency range with clear and relatively consistent patterns over time.

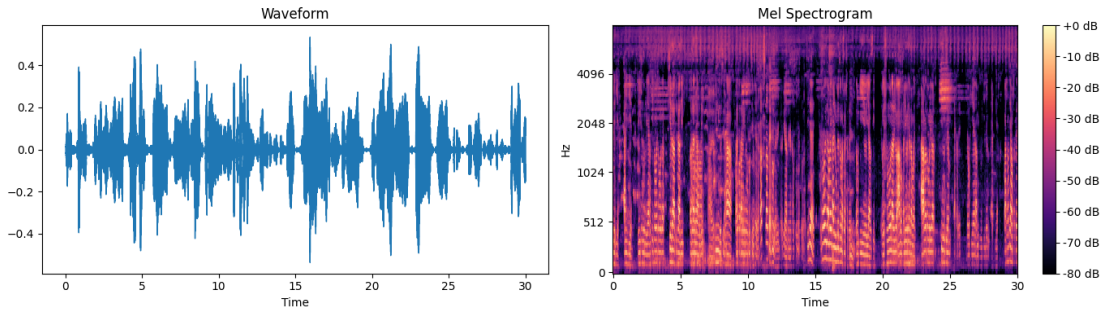


Figure 5.16: Waveform and Spectrogram of a sample from Rangpur

5.2.4 Audio Preprocessing and Segmentation

Given Whisper’s computational limits define a maximum token balance of 448 tokens, audio recordings that exceeded this capacity needed to be pre-processed, to maximize automatic speech recognition (ASR) accuracy. We implemented an audio chopping strategy to work with Whisper in conjunction with its VAD capabilities, which automatically identified segments of audio and modified it depending on detected voice notifications.

Voice Activity Detection (VAD) Implementation

Voice Activity Detection is an important stage of preprocessing that essentially distinguishes speech from non-speech segments in an audio recording. Whisper’s VAD algorithm uses spectral and energy based detector methods that assess periods of vocal activity and periods where there were either silences or potentially background noises. The methods used to categorize vocal activity within the audio segment may use energy, energy thresholds, spectral centroid changes or variations, and zero-crossing rate measures, among other acoustical features, to determine whether human speech is present (or proximate) or not. The VAD mechanism works by monitoring characteristics of properties of the current audio signal and developing adaptive staging thresholds to classify the current time segment. The algorithm finds long stretches of silence or low change in sound level, which typically means there is a break or pause in the speech, and does little when it comes to marking these areas as likely split areas of the audio. This guarantees that audio chunks are created from natural linguistic boundaries, not arbitrary time intervals, preserving the latent semantic meaning of speech units.

Segmentation Strategy

VAD-based segmentation has clear benefits over fixed-duration chunking. By recognizing natural pause boundaries, the system keeps contextual consistency over each

audio segment, and limits the chances of cutting words and partial phrases that negatively affects transcription accuracy. This segmented methodology reduces the chances of segments being too long to fit within Whisper’s token limit as well as maintaining the linguistic structure during ASR processing. Each of the segmented audio files are individually processed through the Whisper ASR system so that each chunk outputs an individual transcription. This approach respects the technical limits of the ASR model, and it may also enhance the overall quality of our transcription by allowing the ASR to concentrate on short, coherent segments of speech, without the computational overhead of processing too much audio context at once. This pre-processing method illustrates the adaptive audio segmentation use case in today’s ASR workflows including the ability to segment transcriptions of longer recordings that exceed model specific limits while producing a transcription that incorporates different and varying regional speech patterns.

5.2.5 Transcribed Texts Preprocessing and Segmentation

After the VAD audio segmentation process, an important alignment process was needed to define the chunks but also align the chunked audio files with the transcriptions. Since the Voice Activity Detection (VAD) algorithm created multiple audio segments out of each recording, the transcriptions also needed to be segmented, by hand, to preserve their precise time and content alignment with their reported audio chunks.

Manual Transcription Segmentation Process

Manual segmentation of the transcriptions was developed in response to the difficulty of aligning the automatically generated audio passages and textual representations of the audio. This required a great deal of time and was a time bad for 14 days (20 hours and 55 minutes total for the full dataset) to complete the alignment of the entire dataset. Manual alignment consisted of an ordered review of each transcription to find logical segmentation that was inline with the VAD silence boundaries in the audio files. Manual alignment was a thorough process that required identifying the audio segment boundaries and accompanying transcription and reviewing each one to pinpoint a good division that preserved semantic meaning while ensuring audio-to-text alignments remained accurate. Manual alignment took an extraordinary amount of time to complete due to the variability in the speech in the different regional recordings and the varied quality of the audio recordings seen in the Habi-ganj, Kishoreganj, and Rangpur recordings.

Districts	Duration	Total chunks	Average chunk size	Type
Habiganj	5 Hours 26 Minutes	2,287	8.56 seconds	Spontaneous
Rangpur	6 Hours	2,475	8.74 seconds	Spontaneous
Kishoreganj	9 Hours 28 Minutes	3,963	8.61 seconds	Spontaneous
Total	20 Hours 55 Minutes	8,725	8.63 seconds	Spontaneous

Figure 5.17: Preprocessed Dataset Information

5.2.6 Data Cleaning and Null Value Removal

After the audio segmentation process was completed, and the alignment of the manual transcription was complete, the dataset underwent an extensive data cleaning process in order to establish quality and integrity. During the data cleaning process, all rows that had a null value were discovered and eliminated from the dataset. The null values in the dataset were generally caused by audio segments that only contained silence, background noise, or not enough speech activity to provide useful transcription results from the ASR system. The null or empty transcription entries could potentially lessen the quality of any further analysis and model training. The complete segmented dataset was searched systematically for cases where the fields for transcription were null, empty, or undefined. Each row would be removed if any of those conditions existed, and we would then have a clean, complete dataset with at least one valid entry for transcription for every audio segment.

5.2.7 Text preprocessing and Token Reduction

During the data preprocessing pipeline, further text cleaning operations were run on the transcriptions to format them for subsequent processing and analysis. This preprocessing problem focused on removing irrelevant textual items that would increase the token count, but otherwise not provide value with respect to linguistic substance. All of the conventional punctuation marks were manually stripped from the transcription text - that is, periods, commas, question marks, exclamation points, and any other traditional punctuation symbols. The theoretical reasoning behind the removal of punctuation in transcription text is twofold: it reduces overall token count, it standardizes text formatting across all transcriptions, and when the transcription text is fed into downstream natural language text processing tasks, punctuation, if not dealt with, could serve as noise. Also, angle brackets and whatever is between them, were detected and removed from the transcriptions. These brackets often contain metadata, speaker identification tags, and/or other annotation markers. While this information may be useful in some contexts, these pieces of metadata do not typically provide underlying linguistic information for speech recognition and linguistic analyses and increase token count.

5.2.8 Exploratory Data Analysis on Preprocessed Datasets

Audio vs Transcription Length

The segmented Habiganj data [5.18] shows a strong grouping of points in the 5-12 range, with transcription lengths typically between 50-200 characters. The relationship is still positively correlated with more pronounced clustering patterns suggesting satisfied segmentation with respect to natural boundaries of speech articulation. There are still a few outliers in the 15-25 range with higher transcription character counts suggesting these segments included particularly dense, or quick, speech content.

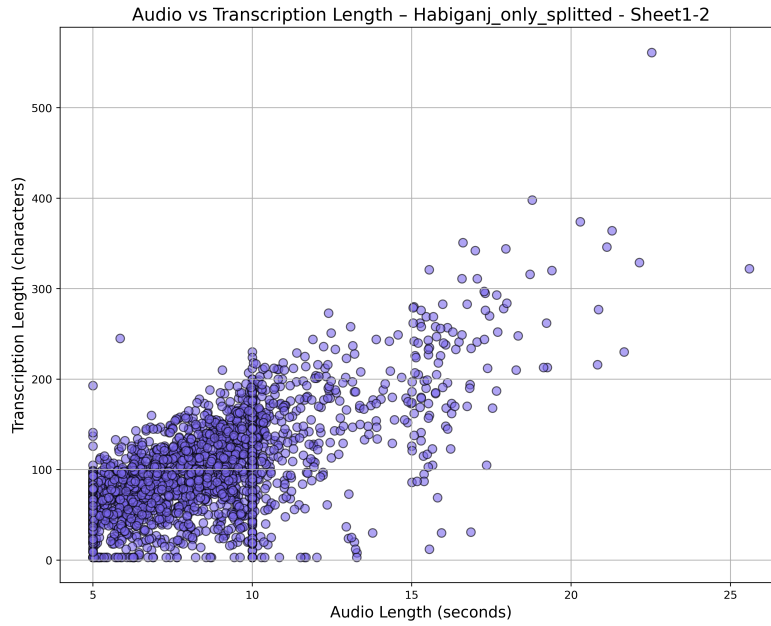


Figure 5.18: Audio vs Transcription Length of Preprocessed Habiganj

Like Habiganj, most data-points of Kishoreganj [5.20] are grouped in the shorter duration ranges and there is a dense collection in the range of 8-15 seconds of audio length. The figure that compares transcription character distribution denotes acceptable equivalence in the 100-250 form transcription character length for most of the segments. The pre-processing has removed the extreme variation in the original data, and retains the original foundation of the audio-text relation.

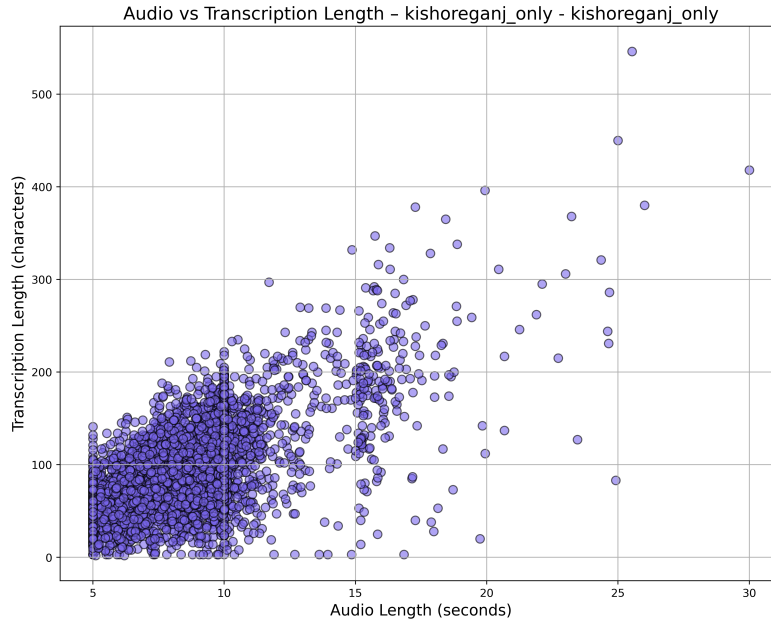


Figure 5.19: Audio vs Transcription Length of Preprocessed Kishoreganj

The segmented Rangpur data [5.20] shows the most uniform distribution of the three regions, with the data points properly represented across the 5-20 seconds. It seems that the preprocessing was most effective for Rangpur, producing a more equal segment size while still keeping the characteristics of natural speech flow.

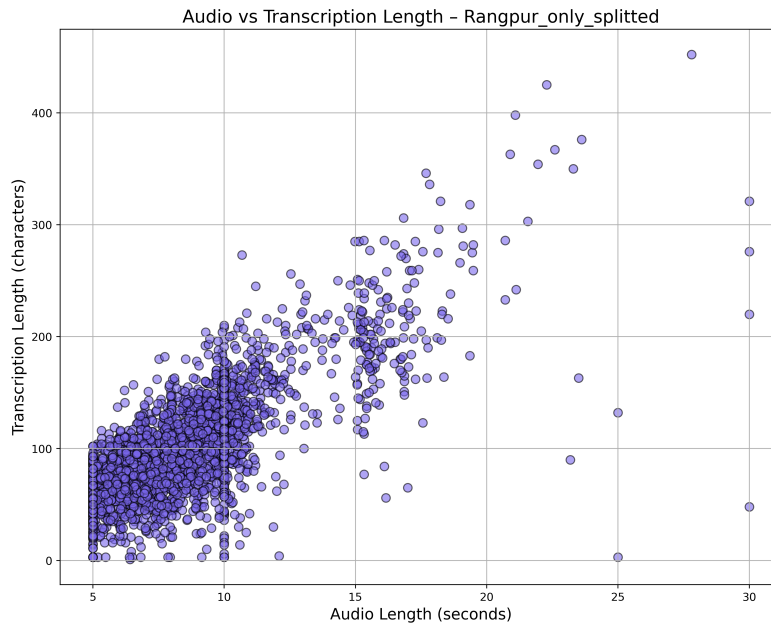


Figure 5.20: Audio vs Transcription Length of Preprocessed Rangpur

Waveform and Spectrogram

The Habiganj [5.21] clip contains a high-quality audio sample with steady amplitude undulations that run full amplitude between ± 1.0 for about 20 seconds. Habiganj displays consistent speech activity with a distinct gap of silence around the 7-8 second mark and clearly marking an area where the VAD algorithm would ordinarily find segmentation boundaries. Consistency in continuous high amplitude oscillation clearly indicates clean speech record with minimal background noise. The respective spectrogram displays a lot of spectral content with concentrated energy (i.e., the yellow/orange areas) primarily in the 0-2048 Hz range. The frequency distribution suggests fairly distinct formant structures and harmonic contours typical of natural speech. Changes in spectral energy over time correspond well to changes in amplitude in the waveform, indicating coherent segments of speech with clear phonetic transitions.

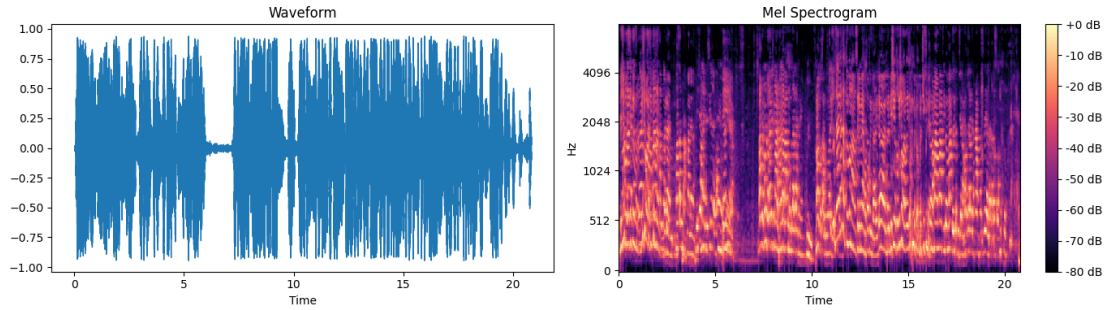


Figure 5.21: Waveform and Spectrogram of a sample from Preprocessed Habiganj

The Kishoreganj segment [5.22] is a much smaller, roughly 9-second chunk of audio with fairly moderate amplitude levels (mostly in the ± 0.2 range) and a few distinct peaks (around ± 0.15 - 0.20). In this case, it appears this segmentation has adequately separated a coherent speech unit, given the amplitude envelope is natural and not abruptly truncated at the boundaries. The spectrogram depicts a greater range of energy distribution compared to Habiganj, with clear spectral activity occurring at certain times. The frequency content in the spectrogram encapsulates identifiable speech characteristics, with formant structures being noted throughout the duration of the segment. Some areas showed distribution of energy with very low amplitude which corresponds with the lower amplitude regions in the waveform, indicating natural pauses or quieter regions of the speech.

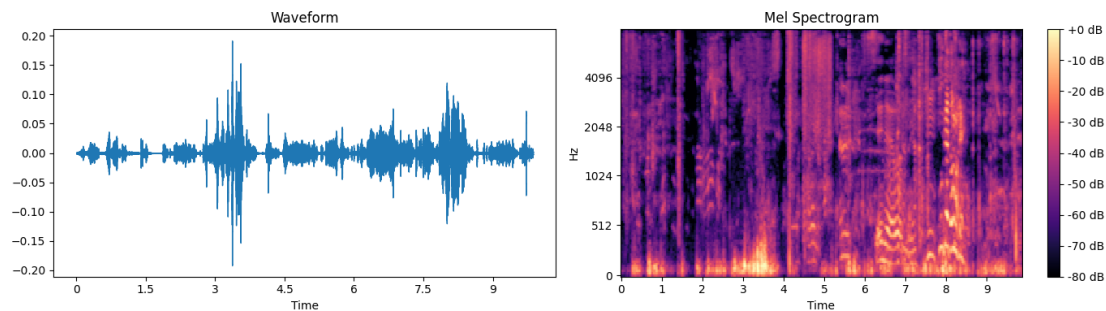


Figure 5.22: Waveform and Spectrogram of a sample from Preprocessed Kishoreganj

This portion of Rangpur [5.23] shows a consistently moderate-amplitude speech activity for about 9 seconds that varies approximately ± 0.4 in amplitude. The waveform shows regular patterns of conversational speech that vary naturally with prosodic variation which is a good indication that segmentation was performed accurately with respect to meaningful linguistic boundaries. Mel Spectrogram Features: The spectrogram displays the most complicated and detailed patterns of frequency of the three segments. The rich harmonic structures are active across the entire frequency range. The temporal-frequency representation shows clear formant tracks and consistent energy distribution, suggesting the phonetic information was dense, yet articulated clearly through the segment.

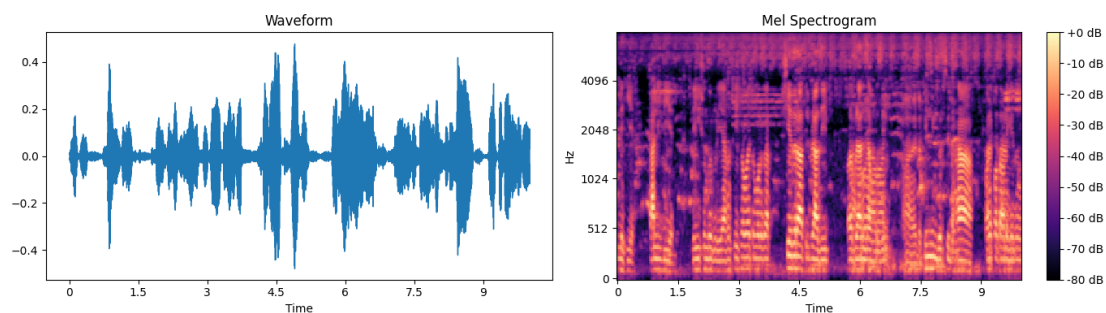


Figure 5.23: Waveform and Spectrogram of a sample from Preprocessed Rangpur

t-SNE (t-distributed Stochastic Neighbour Embedding)

The t-SNE embedding [5.24] indicates that the acoustic features that informed this visualisation - presumably derived either from mel-frequency cepstral coefficients, or some other spectral features or properties of speech - appear to demonstrate regional differences in the speech data. The distinct separation between regions suggests machine learning models trained on this data should perform quite well at regional classification problems. The visualization demonstrates that the preprocessing pipeline has kept the essential acoustic features that differentiate the three regions while forming a data structure for a further pattern recognition and clustering analysis. This dimensional reduction is effective in showing that there are enough acoustic distinctions among the regional speech in the dataset to allow for

automatic regional identification/classification.

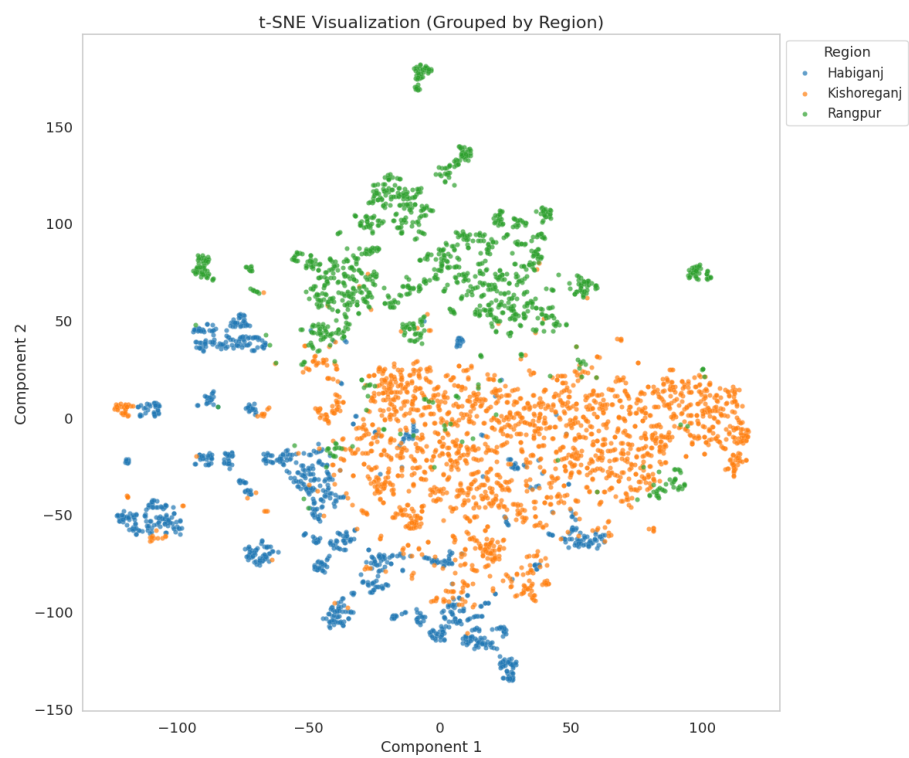


Figure 5.24: t-distributed Stochastic Neighbour Embedding

Regional Gender Distribution Patterns

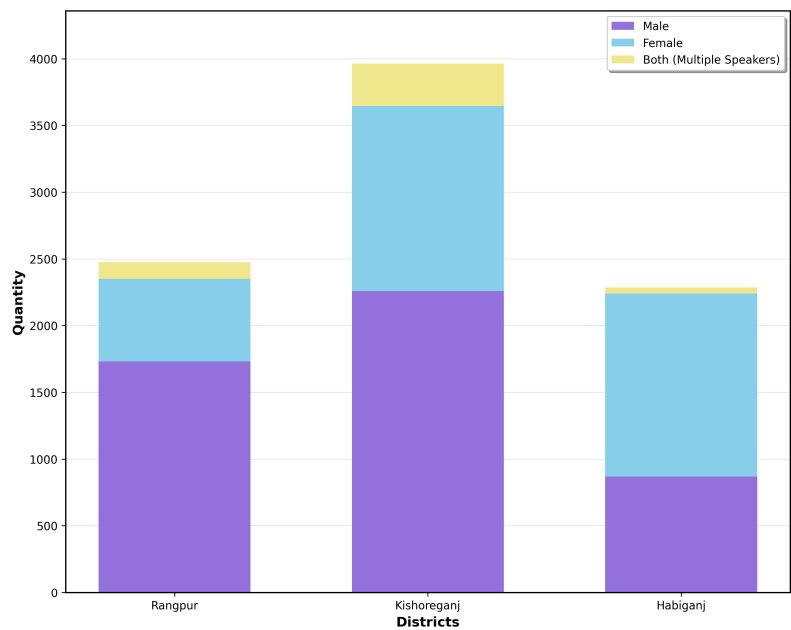


Figure 5.25: Male-Female Ratio in Audio Samples

Kishoreganj accounts for the largest overall contribution to the dataset when compared to the other two districts. The gender breakdown reveals somewhat of a male dominant bias, as can be seen by the segments (purple) that show male speaking, which is around 56%, female speaking segments (light blue) which accounted for approximately 35%, and multi-speaking segments (yellow) approximately 9%. The data collection for Kishoreganj has a marked male speaker bias. Rangpur, the second largest regional contribution is more gender imbalanced than Kishoreganj. Approximately 70% of the segments have male speakers, around 24% have female speakers, and multi-speaker recordings have about 6%. Rangpur has the greatest gender imbalance of the three regions while still showcasing the male-dominated pattern. Habiganj offers the least dataset contribution but has the most evenly distributed gender ratio of the three projects. The male speakers contribute about 38% of segments, while female speakers contribute significantly more at around 58%. The multi-speaker recordings use around 4%. Both the gender ratio and use of female speakers is unique as Habiganj is the only region where female speakers outnumber male speakers. As such, even though the contribution of segments is lower in Habiganj, it creates a reverse pattern from the other two districts.

5.2.9 Dataset Analysis Summary

The full dataset contains 8,725 audio files across three district regions, including Kishoreganj with the most files (3,963), Rangpur with 2,475 and Habiganj with 2,287 files. The total time is an estimated period of 20 hours and 34 minutes. Each recording has an average duration that varies by region, ranging from 8.324 to 9.048 seconds, therefore the recording lengths reflect successful VAD, as segmentation was within appropriate processing time.

Dataset Analysis Summary

Subset	Samples	Duration (H:M:S)	Avg Rec. len.	WPM	WPS	Uniq. Words	OOV %	Characteristic	Avg. Phone Count	Annot. length perc(%)	Compl.
Habiganj	2287	5:26:21	8.562	128	2.131	10518	77.610	0	4.477	97.289	0.973
↳ Habiganj Train	1830	4:20:32	8.542	127	2.110	8753	77.985	–	4.484	97.268	0.973
↳ Habiganj Test	229	0:33:39	8.818	122	2.038	1666	81.453	–	4.460	96.070	0.961
↳ Habiganj Valid	228	0:32:10	8.466	144	2.397	1909	82.347	–	4.434	98.684	0.987
Rangpur	2475	6:00:31	8.740	137	2.286	11757	76.890	0	4.268	99.354	0.994
↳ Rangpur Train	1984	4:48:58	8.739	136	2.274	9967	77.245	–	4.272	99.194	0.992
↳ Rangpur Test	248	0:34:55	8.448	132	2.193	2004	83.184	–	4.148	100.000	1.000
↳ Rangpur Valid	243	0:36:38	9.048	149	2.477	2198	81.529	–	4.360	100.000	1.000
Kishoreganj	3963	9:28:34	8.608	121	2.021	17713	77.728	0	4.553	99.394	0.994
↳ Kishoreganj Train	3168	7:36:28	8.645	122	2.036	15252	78.855	–	4.545	99.590	0.996
↳ Kishoreganj Test	412	0:58:57	8.587	119	1.981	2852	80.224	–	4.758	97.330	0.973
↳ Kishoreganj Valid	383	0:53:08	8.324	116	1.936	2331	78.335	–	4.404	100.000	1.000

Figure 5.26: Dataset Summary

Chapter 6

Models

This research used mT5-small and BanglaT5 for converting various regional dialects to the standard Bengali language and result analysis. These models are especially designed to understand the range and richness of eight regional accents, so that the translation is as close to the original as possible and in the right context. In addition, this research utilized mBERT and CNN for the region detection task to identify the source region of input text and audio. For transcription part, Tugstugi was used as the proposed model and other models such as Whisper-Small, Google ASR, Bangla ASR were used to compare and analyse the inference results. The models' usage is inspired by their extraordinary ability to identify any small deviation in language that is associated with regional dialects.

6.1 Regional Dialects to Bangla Language Translation Models

6.1.1 mT5 (Proposed)

A version of T5 text-to-text transformer model, that is multilingual, is called Massively Multilingual Pre-trained Text-to-Text Transformer (mT5). It has a strong encoder-decoder design. For the pre-train process, an immensely large dataset of 101 languages was used. The mT5 model varies in sizes. The range is from 300 million to 13000 million parameters. Hence, it is really powerful and large. It is very well-known for its remarkable performance in multilingual translation. This is why it is often the go-to choice of people needing translation of copious and diverse languages. And we fine-tuned the model for better inferences in Bangla.

6.1.2 BanglaT5

An advanced sequence-to-sequence transformer model developed specifically for Bengali is called BanglaT5. It is pre-trained on the massive "Bangla2B+" dataset. The dataset comprises 5.25 million documents retrieved from a well chosen list of online sources. The text data size was 27.5 GB. It is built on the original Transformer design and the base framework popularly known as T5. The model has individually 12 layers and attention heads for both, 768 (hidden size), 2048 being the size of feed-forward.

6.2 Region Detection Models

6.2.1 mBERT (Translation)

Multilingual BERT (mBERT) is a transformer-based language model from Google Research that adds on to context-based BERT architecture by including an additional 104 languages. In regards to Arabic dialect detection in speeches, mBERT uses text (whether transcribed or not) to help capture the linguistic characteristics, vocabulary, and grammatical behaviours that identify regional varieties, making it useful to text based regional analysis to some extent. The model consists of 12 transformer encoder layers with 768 hidden dimensions, 12 self-attention heads, and approximately 110 million parameters. Each layer incorporates multi-head self-attention mechanisms, position-wise feed-forward networks, residual connections, and layer normalization. The model uses WordPiece tokenization with a vocabulary of 119,547 subword units that spans multiple languages, enabling effective representation of diverse linguistic patterns including regional dialectical variations. In terms of regional speech detection, mBERT takes the transcribed text from the ASR systems and applies its linguistic framework from the contextual transformation to this text to determine regional speech markers such as vocabulary choices, syntactic structure, and morphological differences. This is aided by the fact that mBERT is pre-trained on multilingual corpus of language models, which provides strong cross-lingual representations of speech that can distinguish small differences in dialect. After fine-tuning with transcription data that has regional labels, mBERT is able to learn distinct features to classify Habiganj, Kishoreganj, and Rangpur speech features based on the textual structure alone.

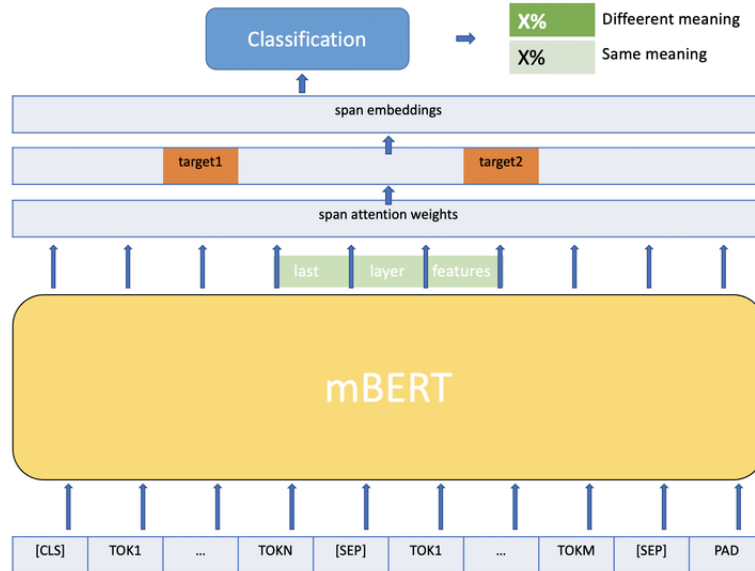


Figure 6.1: Architecture of mBERT

The mBERT architecture has been modified for regional classification by adding a classification head in the form of a linear layer that maps the representation

of the [CLS] token to probabilities for regional categories. This approach utilizes the model’s deeper contextual representation of language to find regionally-specific linguistic patterns, thereby acting as a complement to audio-based detection approaches, and is especially useful in cases where audio may be degraded without transcription being compromised.

6.2.2 CNN Using Mel Spectrogram (Transcription)

The Convolutional Neural Network (CNN) approach to regional detection processes mel-scale spectrogram from audio signals directly, by conceptualizing the regional speech classification process as a computer vision task in which spectrograms represent 2D images that capture time-frequency representations of speech. The acoustic-based approach can extract the prosodic, phonetic, and spectral characteristics that distinguish regional accents in speech. The CNN architecture generally consists of several convolutional layers with filters of varying sizes (3x3, 5x5) and then a max-pooling operation to extract hierarchical features from mel spectrogram inputs. There are batch normalization layers in the architecture to stabilize training, ReLU activations, and dropout layers for regularization. The architecture is such that it will reduce spatial dimensions while increasing depth of features until it reaches fully connected layers that take the extracted features and produce regional classification probabilities. Input mel spectrograms (usually 128 mel-frequency bins) pass through successive convolutional layers that learn to identify region-specific acoustic patterns such as formant structures, pitch contours, and spectral energy distributions. The earlier layers will learn low-level acoustic features such as patterns of fundamental frequency whereas the deeper layers learn complex combinations of acoustic features that could represent regional accent characteristics, pronunciation differences, and prosodic signatures unique to a dialectal region or group. The CNN is able to learn the acoustic signatures that define Habiganj, Kishoreganj, and Rangpur speech, such as distinctions in vowel formant frequencies, differences in consonant articulation patterns, prosodic features (i.e. rhythm and intonation patterns) that complement the standard text, by pinpointing audio characteristics that may not be transmitted through text, e.g., pronunciation differences, rate of speech differences, and acoustic characteristics indicative of regional speech.

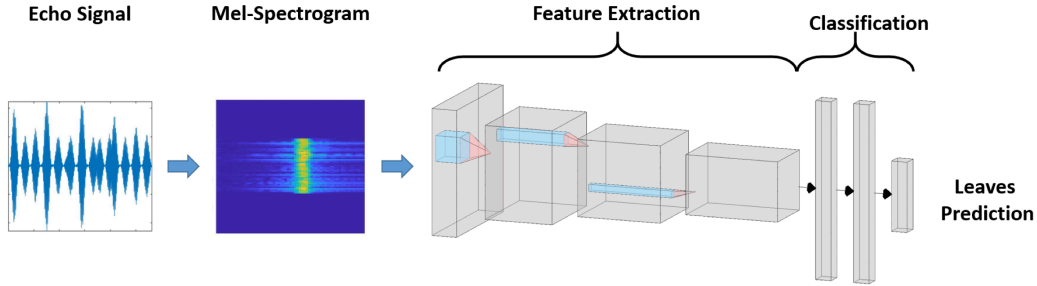


Figure 6.2: Architecture of CNN Using Mel Spectrogram

A traditional implementation may include 4 - 6 convolutional blocks with convolutional, batch normalization, activation, and pooling layers followed by 2 - 3 fully connected layers for classification. The final architecture provides a good trade-off between model complexity and training efficiency, while guaranteeing sufficient representational capacity to discriminate between small acoustic differences due to the regional dialects in the speech data set.

6.3 Regional Dialects to their Transcriptions Models

6.3.1 Google ASR

Google ASR is an automatic speech recognition service, production level, from Google Cloud built on state-of-the-art neural network architectures including deep recurrent neural networks (RNNs), and transformer-based models. Google ASR operates as an end-to-end deep learning system and is built on audio signal processing architectures that use multiple layers of neural networks, optimized for transcribing (in real-time and batch) audio from a wide variety of languages and acoustic environments. The base-level architecture utilizes convolutional neural networks for acoustic feature extraction, then bidirectional LSTM (Long Short-Term Memory) networks and attention mechanisms for sequence modeling. Google's selection applies algorithmic improvements (e.g. CTC (Connectionist Temporal Classification) loss functions, beam search decodes, language models) to yield state of the art transcriptions. The system uses mel-frequency cepstral coefficients (MFCC) or raw waveform inputs to yield audio representations for sequence-to-sequence transcription. Google ASR is capable of recognizing speech in over 125 languages and variants. The system uses transfer learning and multi-task training methods to exploit cross-lingual acoustic similarities while respecting cross-linguistic performance. The model capitalizes on speaker adaptations, exploits noise robustness, and provides automatic punctuation. For dialectal applications, the system could recognize and adapt for accent variations, since it depends on how well regional variants, such as Habiganj, Kishoreganj, and Rangpur dialects, are represented in the data it was

trained in.

6.3.2 Bangla ASR

BanglaASR refers to a contextual automatic speech recognition system built for Bengali/Bangla language processing, with language-specific acoustic models, pronunciation dictionaries, and linguistic knowledge about Bengali phonetics and grammar. This framework usually consists of deep neural networks that capture the properties of Bengali speech, particularly unique phonemics, phonetics, and pronunciation. BanglaASR systems are typically built using a hybrid DNN-HMM (Deep Neural Network - Hidden Markov Model) architecture or end-to-end neural approach, trained on Bengali speech corpora. The acoustic models are often created to appropriately categorise Bengali-specific phonemes, tones, and pronunciation that might be unusable in multilingual systems. The architecture includes Bengali-specific feature extraction and acoustic modelling considering the unique aspects of the Bengali Language. BanglaASR systems generally use convolutional neural networks to extract features from audio signals, then recurrent networks (LSTM/GRU) to model the temporal dynamics of speech sequences. The systems leverage Bengali language models to provide additional context to improve transcription accuracy. The systems can also invoke attention mechanisms to improve alignment with the audio and text sequences, and are trained with CTC to model the temporal dynamics and inherent variable-length audio-to-text mapping of speech recognition tasks.

6.3.3 Whisper-Small

Whisper Small is a transformer-based automatic speech recognition model from OpenAI with an encoder-decoder model type designed for multi-language and robust speech transcription. The model has roughly 244 million parameters, is resource-efficient, and allows for high-level accuracy of transcriptions while capturing speech in many languages or acoustic environments. The encoder consists of 12 transformer blocks with 768 hidden dimensions, 12 attention heads, and processes mel-scale filter bank features extracted from audio input. The decoder contains 12 transformer layers with similar dimensionality, utilizing cross-attention mechanisms to attend to encoder representations while generating transcription sequences. The model incorporates positional encoding, layer normalization, and residual connections throughout the architecture. Whisper Small has been trained on a gigantic corpus of 680,000 hours of multilingual audio, including tasks that are more than transcription, language identification, voice activity detection, timestamp prediction, and several others. This multi-task training allows the model to handle multiple acoustic conditions, accents, and speaking styles, which is especially useful in regional dialectical differences and difficult audio conditions.

6.3.4 Tugstugi-Whisper-Medium (Proposed Model)

Tugstugi is an optimized version of OpenAI's Whisper Medium model tailored to perform better on regional languages and dialectical differences specially pre-trained

on Bangla. The underlying Whisper Medium base model has about 769 million parameters, and the transformer has a larger structure than Whisper Small, while adopting the same encoder-decoder framework. The model has 24 transformer blocks in the encoder and decoder units, they both have 1024 hidden dimensions and 16 attention heads which can improve representational capacity for complex linguistic features. The more parameters to work with enables the model a more thorough knowledge of variation in regional speech, phonetics (subtleties), and dialect because it is more challenging for small models with bandwidth and semantic constraints as well.

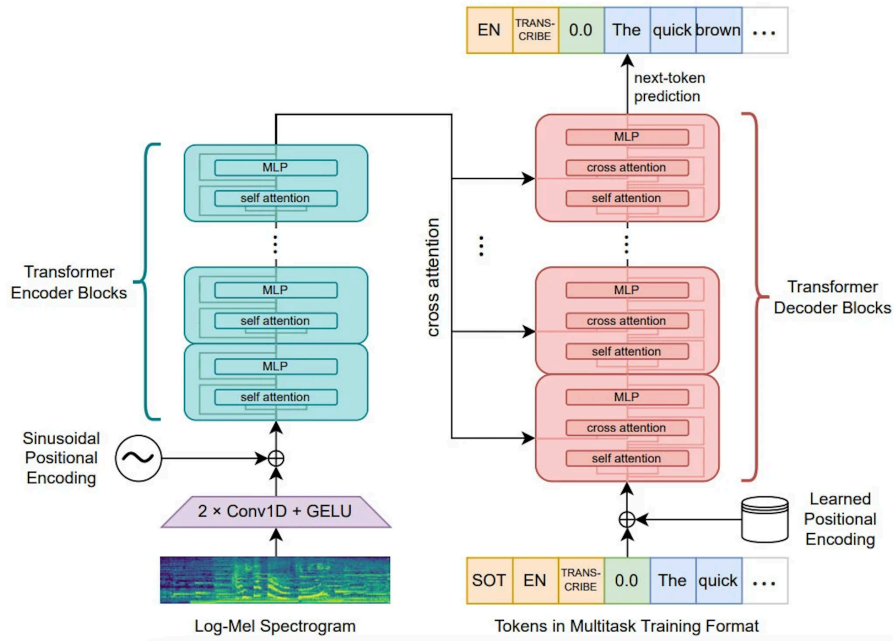


Figure 6.3: Architecture of Whisper

Tugstugi's fine-tuning process is intended to increase transcription performance on specific regional dialects and languages, possibly including new training on regionally-specific speech corpora. This training allows the model to learn pronunciation, vocabulary, and prosodic differences that may be specific to geographical areas, which is really useful for users needing high levels of accuracy on regional dialectal speech like Habiganj, Kishoreganj and Rangpur dialects in this context.

Chapter 7

Evaluation Metrics

In this section we delve into various evaluation systems to evaluate our model's translation and region detection capabilities, dividing the metrics used into two primary categories: Region Detection Metric and Translation Metrics.

7.1 Dialect-to-Bengali Translation Metrics

7.1.1 Character Error Rate

Character Error Rate (CER) is usefully suited in the evaluation of accuracy for character-level generation of text. It quantifies the amount of difference between generated text and the the original in terms of substitution, insertion and deletion errors. CER is given as a percentage or a fraction where lower percentage or fraction represents higher level of accuracy of the replicated text.

$$\text{CER} = \frac{S + D + I}{N} ,$$

Where:

- S : Number of substitutions.
- D : Number of deletions.
- I : Number of insertions.
- N : Number of total characters in the text in question.

7.1.2 Word Error Rate

WER is a performance measurement benchmark that quantifies the absolute difference of generated text against the ground truth by the word. It quantifies errors of substitution, insertion, deletion and shift of words as well as permutation of words. WER is useful in evaluation of performance of the translation systems, where lower WER implies better translation accuracy and WER values near zero suggesting a translation in the reference.

$$\text{WER} = \frac{S + D + I}{N} ,$$

Where:

- S : Number of substitutions.
- D : Number of deletions.
- I : Number of insertions.
- N : Number of total words in the reference text.

7.1.3 BLEU Score

BLEU stands for Bilingual Evaluation Understudy, a measure of quality for translations generated by any machine translation system. In other words, it measures the closeness of a machine-generated translation to a human reference translation, considering the accuracy and fluency of the output. When calculating the BLEU score, it takes what it factors as well as four components including precision, recall and a brevity penalty to measure generated text appropriately in a proper manner. What BLEU does is compute the matching of n-gram, where n is the number of words being considered in a group, between the text generated by the machine and the reference text. Higher scores indicate better translation quality. BLEU scores range from 0 to 1. With languages like Bengali, where higher scores indicate closer alignment with human translations, this metric works especially well for assessing translations in this language.

$$\text{BLEU} = \text{BP} \times \exp \left(\sum_{n=1}^N w_n \log p_n \right) ,$$

Where:

- BP: Brevity Penalty, which accounts for short translations.

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

- c : Length of the candidate translation.
- r : Length of the reference translation.
- p_n : Precision of n-grams.
- w_n : Weight of each precision (typically $\frac{1}{N}$ for uniform weight).
- N : Maximum n-gram length (e.g., 4 for BLEU-4).

7.1.4 METEOR Score

METEOR (Metric for Evaluation of translation with Explicit Ordering) is a robust metric for measuring translation quality regarding regional dialects to Bengali language translation. The metric will ensure translation quality by comparing it with human-constructed references.

$$\text{METEOR} = \frac{10 \times P \times R}{R + 9 \times P}$$

, Where:

- P : Precision (the proportion of matched words to words in the candidate).
- R : Recall (the proportion of matched words to words in the reference).

7.2 Region Detection Metrics

7.2.1 Accuracy

In region detection, accuracy might be defined as the ratio of correctly classified regions against the total number of regions in the dataset. It gives a measure of the efficacy of the model in properly assigning an unseen text to its corresponding region.

$$\text{Accuracy} = \frac{\text{Number of Correctly Detected Regions}}{\text{Total Number of Regions}} \quad (7.1)$$

7.2.2 Precision

Precision shows the ratio of true positives (number of correctly predicted occurrences of a given region) to the sum of true and false positives (occurrences predicted as a member of the region). It tells the extent by which a model correctly predicted a particular region out of the total number of instances it classified as that region.

$$\text{Precision} = \frac{\text{Correctly Detected Regions}}{\text{Total Detected Regions}} \quad (7.2)$$

7.2.3 Recall

In region detection for regional dialects, recall quantifies the model’s capacity to retrieve the text samples from the datasets “Vashantor” and “Bengali.AI” with specific region. It shows the effectiveness of the model in establishing the correct region of texts, while avoiding to miss any instance of classifying an instance into the wrong region.

$$\text{Recall} = \frac{\text{Number of Correctly Detected Texts for a Specific Region}}{\text{Total Number of Texts Belonging to that Region}} \quad (7.3)$$

7.2.4 F1 Score

F1 Score is the average of precision and recall with their harmonic mean, and here we use it to judge the quality of region detection. It is particularly useful when used for region detection tasks and trying to minimize errors of the first and second type.

$$\text{F1 Score} = \frac{2 \cdot \text{Precision for Region Detection} \cdot \text{Recall for Region Detection}}{\text{Precision for Region Detection} + \text{Recall for Region Detection}} \quad (7.4)$$

Chapter 8

Result Analysis

8.1 Translation

8.1.1 Bangla Dialects to Standard Bangla

With the help of the mT5 and BanglaT5 models, we obtain two translated texts for each input text given. This provided sixteen translations that will cover the eight regional dialects. By using this method, we can compare one translation option with the other and choose which one is closer to the standard Bangla language. To determine the quality of our translations from regional dialects to Bangla, we then applied four types of metrics to measure the performance.

Region	CER	WER	BLEU	METEOR
Rangpur	0.1965	0.3584	0.4701	0.6588
Habiganj	0.1376	0.2779	0.5189	0.7117
Kishoreganj	0.2109	0.4172	0.4491	0.6375

Table 8.1: Performance of Trained mT5 on the Merged Dataset

From the table 8.1, we can see that mT5 was trained on the merged dataset where there were data of Habiganj, Kishoreganj and Rangpur. To improve the performance, we decided to implement the mux model where mT5 is trained on the 3 regions separately and the text is passed to it through region detection (mBERT), working like a MUX for better performance.

Table 8.2 and 8.3 show the performance of two models: mT5 and BanglaT5 in translating regional Bangla text from eight different regions: Chittagong, Noakhali, Sylhet, Barisal, Mymensingh, Rangpur, Habiganj, and Kishoreganj to standard Bangla. mT5 is found to outperform BanglaT5 in most regions. A similar pattern is observed for the Bengal.AI dataset where mT5 outperforms BanglaT5 in most regions.

Region	Models	CER	WER	BLEU	METEOR
Mymensingh	mT5	0.0827	0.1695	0.6477	0.8169
	BanglaT5	0.0823	0.1548	0.7402	0.8258
Barishal	mT5	0.1343	0.2464	0.5294	0.7430
	BanglaT5	0.1691	0.2762	0.4852	0.7001
Sylhet	mT5	0.2131	0.3639	0.4029	0.6263
	BanglaT5	0.2481	0.3911	0.5126	0.6770
Noakhali	mT5	0.2264	0.41875	0.3469	0.5842
	BanglaT5	0.2971	0.48562	0.4848	0.5909
Chittagong	mT5	0.2888	0.4665	0.3081	0.5555
	BanglaT5	0.2255	0.4658	0.3100	0.5859

Table 8.2: Performance Overview of mT5 and BanglaT5 on all the Bangla regional dialects from Vashantor Dataset

Region	Models	CER	WER	BLEU	METEOR
Habiganj	mT5	0.1200	0.2600	0.5097	0.6885
	BanglaT5	0.1714	0.3384	0.5112	0.6349
Rangpur	mT5	0.1126	0.2353	0.6187	0.7545
	BanglaT5	0.1452	0.2140	0.5825	0.7026
Kishoreganj	mT5	0.1886	0.3752	0.4701	0.6683
	BanglaT5	0.2023	0.3664	0.4845	0.6722

Table 8.3: Performance Overview of mT5 and BanglaT5 on all the Bangla regional dialects from Bengali.AI Dataset

8.1.2 Region Detection

Table 8.4 and 8.5 presents an analysis of the region detection models mBERT on Vashantor and Bengali.AI datasets. mBERT model correctly predicted approximately 85.33% of the instances. On the other hand, it showed an accuracy of 91.95% on the Bengali.AI dataset. Therefore, mBERT performed well in classifying the dialects for the region detection task. A total of eight different regions are assessed on the precision, recall, and f1-score from both datasets before proceeding to the region specific metrics.

Region	Precision	Recall	F1-Score
Mymensingh	0.7885	0.9547	0.8637
Barishal	0.9436	0.9840	0.9634
Sylhet	0.9533	0.6533	0.7753
Noakhali	0.8031	0.8267	0.8147
Chittagong	0.8708	0.8987	0.8845

Table 8.4: Performance Overview of mBERT for region detection on the Vashantor Dataset

Region	Precision	Recall	F1-Score	Accuracy
Rangpur	0.8520	0.9516	0.8990	0.9516
Habiganj	0.9621	0.8943	0.9269	0.8943
Kishoreganj	0.9433	0.9127	0.9278	0.9127

Table 8.5: Performance Overview of mBERT for region detection on the Bengali.AI Dataset

We improved the performance of our models by fine-tuning their hyperparameters which are the external configuration variables that have control over the regional dialect translation and region detection performance by the models. For both mT5 and BanglaT5 models, the key hyperparameters included the fixed number of 50 epochs for each region with a learning rate of 0.001 and a fixed batch size of 16 using the AdamW optimizer and a sequence length of 128. Similarly, for all the regions, mBERT is trained with consistent hyperparameter values that include 10 epochs with a learning rate of 0.00002 and a batch size of 16 using the AdamW optimizer and a sequence length of 128 shown in the figures 8.6 and 8.7.

Hyperparameter	Configuration
Learning Rate	0.001
Batch Size	16
Number of Epochs	50
Optimizer	AdamW
Sequence Length	128

Table 8.6: Hyperparameter Tuning on both mT5 and BanglaT5 models for Regional Bangla Dialect Translation into Standard Bangla

Hyperparameter	Configuration
Learning Rate	0.00002
Batch Size	16
Number of Epochs	10
Optimizer	AdamW
Sequence Length	128

Table 8.7: Hyperparameter Tuning on mBERT for Region Detection

8.2 Transcription

8.2.1 Audio to Bangla Transcription

Model Performance

The table 8.8 shows the results of the inferences on different models using our proposed model. The Habiganj outperformed all the regions and Kishoreganj performing less than the other two regions.

Region	Word Error Rate (WER)	Character Error Rate (CER)
Habiganj	0.6154	0.2406
Kishoreganj	0.8000	0.4440
Rangpur	0.7101	0.3437

Table 8.8: Model Performance Summary

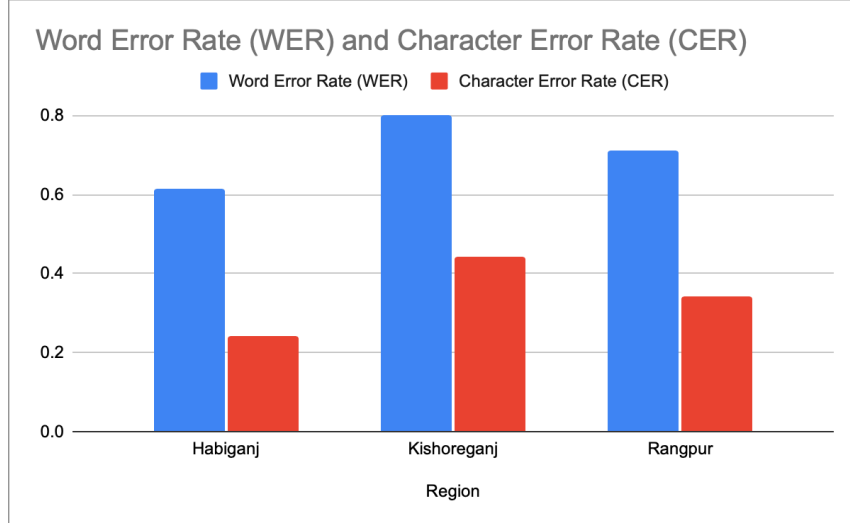


Figure 8.1: Model Performance Summary in Chart

Model Inferences

Samples of the inferences from the test data by our proposed model are given below.

test_habiganj (29)_chunk_00 1.wav	তেনন কুন অসুবিদা লাগে না। এরফরে তো ফুরা-ফুরিবারে খরি না, খরতাম ফারিও না।	Habiganj	test	12	তেনন কুন অসুবিদা লাগে না। এরফরে তো ফুরাফুরিতাবে খরি না, খরতাম ফারিও না।	0.25	0.06666667014
test_habiganj (30)_chunk_00 0.wav	ওই যে আমার ফড় যে, সেলফ নাই, সেলফ ছাড়া, এরফরে সেলফ থাকলেওই কিতা, সেলফের মাইকে যদি আমার কিয়ইজ	Habiganj	test	18	ওইযে আমার ফড়া যে সেলফ নাই, সেলফ ছাড়া, সেলফ তাকলেওই কিতা, সেলফের মাইকে যদি আমার কিয়ইজ	0.722222209	0.1836734712
test_habiganj (30)_chunk_00 1.wav	কুন রাখি, এরফরে, এরফরে খাই যে, ইতা জুটা থাল-রাসন রাখি তাইলে কিন্তু আমারা কড় রাখান নিমম না। ফড়টাইন ইতা শবিরে জামগাত রাখান লাগে। এখন আমি কিতা খবু? জানালৈ সহিতে কুনমতে রাখছি। রইছা	Habiganj	test	32	কুনরাখি, এরফরে খাই যে ইটা জুটাত আলবাসিনরাখি। তাইলে কিন্তু আমারার ফড় রাখান নিমম না। ফড়-উটাইন একটা শবিরে জাগাত রাখান লাগে। এখন আমি কিতা খরিমু? জানালা সহিতে কুন মতে রাখছি। <>	0.6875	0.2270270288
test_habiganj (31)_chunk_00 0.wav	জানালার মাইকে রাখছি, ওখন ইখালেই	Habiganj	test	5	<> যেখালে আমার মাইকে রাখছি। এখন ইখালে	1.399999976	0.46875
test_habiganj (31)_chunk_00 1.wav	একটু ইয়া খরি। আবার মালা-চালা জডি। কিন্তু ফুরা-ফুরিবারে খরা ওয় না। বাড়িত থাকতে যেমন খরতাম এমন, এমনবারে কিন্তু খরতাম ফারি না।	Habiganj	test	21	একটুই খরি। আবার মালা-চালা-খি। কিন্তু ফুরাফুরিতাবে খরা ওয় না। বাড়িত তাকতে যেমন খরতাম, এমন, এমনভাবে কিন্তু খরতাম ফারি না।	0.380952388	0.08943089098
test_habiganj (32)_chunk_00 0.wav		Habiganj	test	0	<>	1	1
test_habiganj (32)_chunk_00 1.wav	আচ্ছা, ভুমার রাসা-টায়া নিয়ে এখন খওছান, কিতা খরতাম ফারা?	Habiganj	test	9	আচ্ছা, ভুমার গানগা টোয়া নিয়া এখন খওছান কিতা খরতাম ফারা।	0.555555582	0.137931034
test_habiganj (33)_chunk_00 0.wav	তুমি রাসা-টায়া খরতে বালা ফওনি? আমি তো ছুই থাকতে থিকাওই খালি হইতাম, রানতাম, রানতাম। ওখন আমার মা বেশি ছুই থাকতে দিতে না। দিচ্ছা থাকতে তো আমি	Habiganj	test	26	তুমি রাসা-টায়া খরতে বালা খওনি? আমি তো ছুই থাকতে থিকাওই খালি হইতাম, রানতাম, রানতাম। এখন আমার মাই বেশি ছুই থাকতে দিতে না। দিচ্ছা তাকতে তো আমি	0.4230769277	0.1020408198

Figure 8.2: Habiganj Inferences

test_kishorega nj (176)_chunk_0 01.wav	রেড অর্থ লাল, রেড অর্থ লাল, রেড অর্থ লাল, রেড অর্থ	Kishoreganj	test	11	রে, ডক্তলা, রে, ডক্তলা, রে, ডক্তলা, রে, ডক্তলা	1	0.4799999893
test_kishorega nj (188)_chunk_0 01.wav	ফর আরও ফর। পিন অর্ড সবুজ। উদ্ভিতের পুষ্টি সুধানর জন্য কুনটি প্রয়োজন? মা একটু উদ্ভিতের পুষ্টি সুধানর জন্য কুনটি প্রয়োজন? মা একটু মুহুস্ত করা।	Kishoreganj	test	24	আল, ফ, ফ, আল, ফ। নাকি নাকি নাকি। উদ্ভিত পুষ্টি শ্রদের জন্য কোনটি পরিজন মাটি। উদ্ভিত পুষ্টি শ্রদের জন্য কোনটি পরিজন মাটির মাছ।	1	0.4785714149
test_kishorega nj (95)_chunk_00 1.wav	ফইরাই তাল মাইরালা। এহ? নামাজ ফইরাই মছিদো তাল মাইরালা। কেহ? মছিদ কুলা তয় না	Kishoreganj	test	14	ফরে এই তালমাইরা। হাঁ? নামাজ। ফরে এই মসজিদো তালমাইরা। কারে? মসজিদ কুলাতাইনা।	1	0.407407403
test_kishorega nj (145)_chunk_0 02.wav	কীটনাশক, হাসপাতালের বর্জ্য, গৃহস্থালীর মানে <> বাড়ির আবর্জনা	Kishoreganj	test	7	কীটনাশক। হাসপাতাল বর্জ্য। গৃহস্থালি, মানে নামতাইন, বাড়ির আবাজন না, হে।	1	0.400000006
test_kishorega nj (103)_chunk_0 01.wav	প্রানির উপর নির্ভরশীল। নির্ভরশীল টিকাছে। উদ্ভিদ কুনতা আর প্রানি কুনতা এইজ ক?	Kishoreganj	test	12	পানির উপর? নির্ভরশীল। নির্ভরশীল, টিকাছে। এ উদ্ভিদ কুনা আর পানি কুনা যে ক?	1	0.2763157785
test_kishorega nj (136)_chunk_0 01.wav	কাডাল, আম, তরমুজ। হাঁ, তরমুজ, পেঁপে। কলা।	Kishoreganj	test	7	কাডা আম তরমুজ। হাঁ তরমুজ? পেঁপে। কলাকাড	1	0.2380952388
test_kishorega nj (83)_chunk_00 0.wav	আম চুকা,	Kishoreganj	test	2	আম, চুকা।	1	0.200000003

Figure 8.3: Kishoreganj Inferences

test_rangpur (75)_chunk_001.wav	মেয় সরাই দিবে, ওইয়ে আসুলা যেমন কইল বলে আমার অদি পরায় পরিশোধ। তা আমার মতন মাইনসের কি হইসে? <> কিছু কিছু মাংসা আসে বেটি, তোমার টেকা হইলো ওই সেজনের পর মেয় অনেক।	Rangpur	test	31	মেয়। সবাই দিবে। ওই যে আসলে যেমন কইল বলে হামারি পডায় পরিষদ। না, সেটা ঠিক আছে। কিছুটা আছে। তা হামার এটা মাইনসের কী হইচ? কিছুটা আছে বেটি। তোমার টেকা হইলো ওই সেজনের পর মেয় অনেক।	0.75	0.4223602414
test_rangpur (29)_chunk_001.wav	নিবো হয় এটে পার এটে ভাজা, ভাজাও কম লাগিল হয়, এদি কাছ ইট। গীরা বাকারের এটা।	Rangpur	test	16	মানে হয় এতে ভাজা, এতে ভাজাও কম লাগে নয়। যদি কাছ ইট, কারাবাজারের এতে।	0.75	0.3421052694
test_rangpur (25)_chunk_002.wav	মজবুত হয়। <> মজবুত হয় সেটা ঠিক আসে কাম তো কামের গজিত চলবার নয় খালি মজবুত হওয়ার সাথে খাইল হইবে।	Rangpur	test	19	লে মজিবুত হয়। হ, মজবুত হয় সেটা ঠিকনাই। কামতো কামে গজিত চলা নাপবার নয়। মজিব হীসা খাকল হইবে।	0.75	0.340206176
test_rangpur (84)_chunk_000.wav	যোরতো হোয়া সেলা মেজাজ খারাপ। বলে সগ পোড়া দিলে তোর পোয়াত ভুতি নাপি ছ্যাচো নাপস? হা,	Rangpur	test	16	যোর তো হোয়া সেলা মেজাজ খারাপ। বলে শোক পুরা যেহেলা তোর উয়াত ভুতি নাপি, ছেচো নাপসে। হ্যা?	0.75	0.3095238209
test_rangpur (78)_chunk_001.wav	চর মর শুলা এখনো আসে। <> অইল্লা আবাদ করে না? <> নাহ, অহেনা ভুট্টা গাজসে, এমারা এইয়ে তায়ত গাইসে। ওই ভুট্টা গাজিও পানি নেয়।	Rangpur	test	22	চর মর শুলা এখনো আসে। এ নাবাত করে না? না, ওনা ভুট্টা গাজছে, এমনা ওই যে কায়াত গাজছে, ওই ভুট্টা গাজি ও পানি নাই।	0.75	0.2561963466
test_rangpur (94)_chunk_000.wav	হামার বাড়িত পসিল সেদিনকা। <> পেইস? যান্না কি কয় পরে? <> যান্না তো মোক খোজ ভাইতো আন্দুলার বাপ নাই,	Rangpur	test	17	হামার বাড়িত পেছিল সেদিন সময় করি। পেছিল? যা কি কয় করে? যান্না তো ওমাক খোজ, তান্না জাহুলরাক নয়।	0.7368420959	0.3636363745
test_rangpur (69)_chunk_001.wav	<> যান্না যে মসজিদত যাবার চায়। অয়ে মসজিদত আর বাড়িত খাওয়া এক হইলো? আর ওই যে তোমার মোজ টা।	Rangpur	test	18	তায় যে মসজিদত যাবার চায়। কয় মসজিদো তার বাড়িত খাওয়া ইয়াক হইলো। আর এই যে ওমার রাজা।	0.7368420959	0.2758620679
test_rangpur (87)_chunk_001.wav	দিবে মনে করা সবায় কালেক্ট কইরেবে টাকা বুজেন নাই। যাই যতো মেয় পঞ্চাস, একশো, তেরশো, দুইশো। দুইশো ওটে দিয়া।	Rangpur	test	19	দেবে মনে করা সবাই কালেক্ট করবে টাকা। বুজেনা এই যাই যত যান্না পঞ্চাস একশো তেরশো, দুইশো, দুইশো টাকা দিয়া	0.7368420959	0.2427184433
test_rangpur (8)_chunk_000.wav	ভালো মন খালি চায়। মাছ-পোস্ত খাবার চায় খালি?খালি মাছ-পোস্ত খাবার চায়। এহেন মাছ-পোস্তর যে এলা দামা যে দামা এলা মনে করা	Rangpur	test	22	ভালো মন খালি চায়। মাছ পোস্ত খাবার চায় খালি। খালি মাছ পোস্ত খাবার চায়। আর মাছ পোস্ত যে এলা দাম। যে দাম। এলা মনে করা না।	0.7368420959	0.1293103397
test_rangpur (118)_chunk_000.wav	কি এটে? <> হ কইলক শুক্ক। <> কুশুদা পেরাছি আবোরা। <> টেকা চাইশো টাকা ওইয়ে	Rangpur	test	12	<> হ কইল শুক্ক। <> পশুদ একবার আছিল। আরা চাইশো টাকা	0.7333333492	0.5625
test_rangpur (95)_chunk_000.wav	হামরা ফির রেডি হই। এটে আসমা। বোরকা পিন্দিছত, ময়নাক কাপড় চোপড় পিন্দি দিছত। এলা আসমা।	Rangpur	test	15	হামরা ফির এডি হইসে এটে আসু। বোরকাপিন দিসেম, ময়নাক কাপিসোপাকিন দিসেম। হুম।লা আসু।	0.7333333492	0.3647058904

Figure 8.4: Rangpur Inferences

Benchmarking Performance

As indicated, the comparative analysis shows that the proposed Tugstugi-Whiper-Medium model outperformed all baseline models across the regional datasets, showing that it was the top-performing model for all datasets. The performance evaluation compared the proposed Tugstugi-Whiper-Medium model against the best-performing versions of the baseline models, indicating consistently better performance across the regional datasets compared to all baseline models. Moreover, while trained on the region specific dataset on Whisper-small, it gave us better results than the training on the merged dataset; which led us to make the mux model training 3 models individually on specific regions.

Models	Habiganj		Kishoreganj		Rangpur	
	WER	CER	WER	CER	WER	CER
Whisper-Small(Raw Data)	0.882	0.523	1.048	0.887	0.832	0.549
Whisper-Small(Region-wise)	0.79	0.4333	0.9503	0.5896	0.8769	0.4778
Whisper-Small(Split Data)	0.77	0.3549	0.9565	0.5911	0.8545	0.4612
Google ASR(Split Data)	0.9091	0.7021	1.0	0.9621	0.9692	0.8462
Bangla ASR(Split Data)	1.22	0.5875	1.52	0.7665	0.9682	0.7053
Tugstugi(Proposed)	0.6154	0.2406	0.8	0.444	0.7101	0.3437

Table 8.9: Benchmarking Performances

In the table 8.9, For Habiganj, the proposed model has provided excellent results with a WER of 0.6154 (61.54%) and CER of 0.2406 (24.06%). In contrast to the baseline performances of the best-fared Whisper-Small (Split Data) at WER of 77% and CER of 35.49%, the proposed model provides a 20.06% improvement in WER and 32.21% improvement in CER which is a substantial leap in the previous best configuration.

For Kishoreganj, the proposed model performs remarkably with WER of 0.8 (80%) and CER of 0.444 (44.4%). Compared to the best baseline, with Whisper-Small (Region-wise Data), with WER of 95.03% and CER of 58.96%, the proposed model can show a 15.82% improvement in WER and 24.70% improvement in CER. These improvements are particularly significant given the extreme difficulty that is inherent to identifying regional dialects.

Lastly for Rangpur, the suggested model continues to have good performance with WER of 0.7101 (71.01%) and CER of 0.3437 (34.37%). In comparison, the best performing baseline, Whisper-Small (Split Data), produced a WER of 85.45% and CER of 46.12%. The model displayed a 16.90% reduction in WER and a 25.46% reduction in CER.

Cross-Model Comparison: Significantly, while Google ASR and Bangla ASR may show competitive performance on some metrics, they also always exhibited different results across regions. The proposed model exhibited a stable and better performance across all tested regions, indicating its robustness and generalizability for the different Bengali dialects and variety of speech styles. The consistent improvements across all measures indicate the benefits of the proposed architectural enhancements and training practices.

8.2.2 Region Detection

Region	Precision	Recall	F1-Score	Accuracy
Habiganj	0.9655	0.9803	0.9729	0.9803
Kishoreganj	0.9688	0.9652	0.9670	0.9652
Rangpur	0.9614	0.9536	0.9575	0.9536

Table 8.10: Region Detection of Audio Files using CNN

The credential based region detection model built on CNN Mel Spectrogram exhibits extraordinarily strong performance across all three regional datasets. Habi-

ganj shows the highest performance: accuracy 98.03%, precision 96.55%, recall 98.03%, F1-score 97.29%. Kishoreganj is similarly strong with accuracy 96.52%, precision 96.88%, recall 96.52%, F1-score 96.7. Rangpur shows strong performance at accuracy 95.36%, precision 96.14%, recall 95.36%, F1-score 95.75. The model demonstrates extraordinary consistency with only 2.67% variation in accuracy over regions, ranging from 95.36 to 98.03%. F1-scores for all regions are over 95%, demonstrating a good balance between precision and recall. The results provide assurance that CNN architecture with Mel Spectrogram features are appropriate for accurately recognizing and characterizing their existing acoustic qualities for Bengali regional dialects, and provides a strong base for automated dialect classification systems.

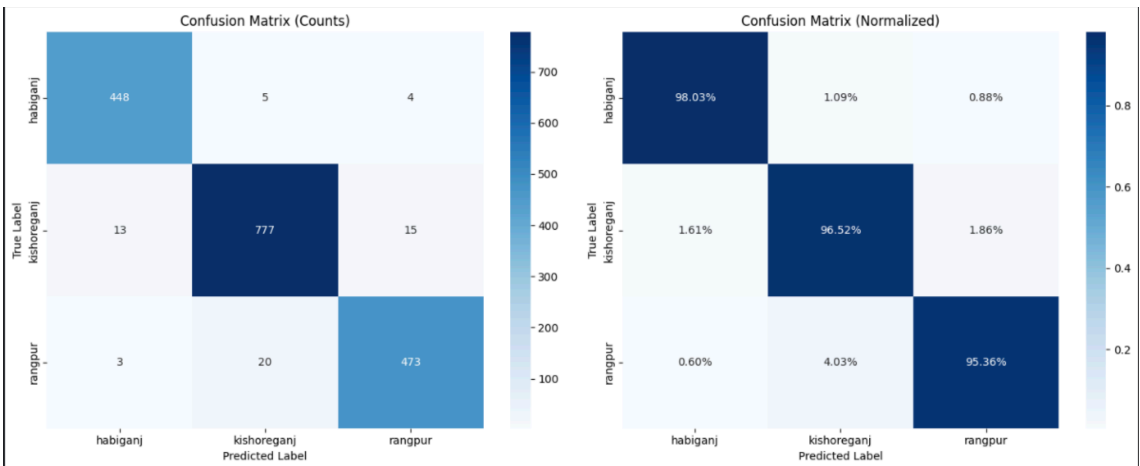


Figure 8.5: Confusion Matrix

Chapter 9

User Interface

9.1 Developing The BanglaMUX System

Our Bengali Dialect Transcription and Translation System has a straightforward design that makes it easy to use. It's built for Bengali speakers and seeks to connect different regional dialects with Standard Bengali using smart speech-to-text transcription and text-to-text translation technology.

9.1.1 Frontend Stack and UI Implementation

The system's frontend is built with HTML, CSS, and JavaScript. It's designed to be user-friendly, so even those who aren't tech-savvy can use it easily. The design keeps things accessible and clear, working well on different devices. Key UI features include:

- **Input Field for Text Translation:** You can type in Bengali dialect in the text box. After you submit it, the system recognizes your region with mBERT and translates your text into Standard Bengali.

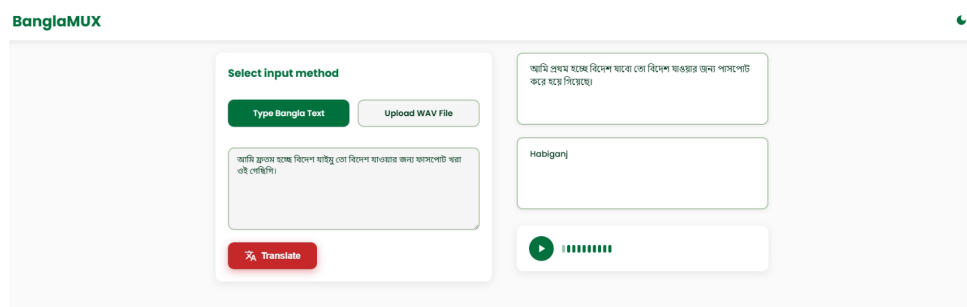


Figure 9.1: BanglaMUX UI For Translation (Light Mode)

- **Audio File Upload for Transcription:** You can upload WAV audio files to the system. Once you do that, it'll process the audio and transcribe it. If you want to translate the transcription into standard Bangla, just copy and paste it into the translation input box.

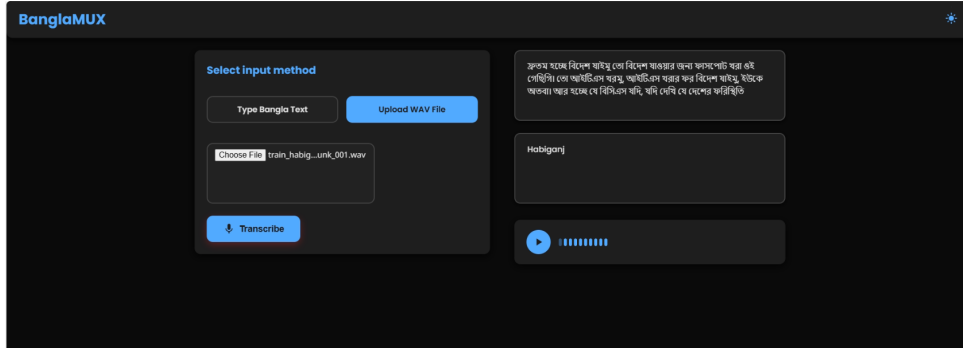


Figure 9.2: BanglaMUX UI For Transcription (Dark Mode)

- **Feedback Mechanisms:** The UI shows error messages when users input something wrong, so they know if the text or audio file isn't right for processing.
- **Responsive Design:** The interface shows error messages when there are invalid inputs, so users know if their text or audio file can't be processed. It also adjusts to different screen sizes and devices, making it easy to use on both desktops and mobile.

9.1.2 Backend Integration with AI Models

The system's backend is set up with Flask, acting as the main server to handle user requests. When a user uploads an audio file or types in text, the system processes it through different machine learning models that are made for specific jobs, as explained below:

- **Audio Processing for Dialect Identification and Transcription:** After uploading, backend starts by pulling out some features from audio file. The CNN-based audio classification model figures out the regional dialect. After identifying the dialect, the right Whisper model is used to turn the audio into Bengali text.
- **Text Translation:** To handle Bengali dialects, the system starts by using an mBERT-based classifier to figure out the region. Then, it uses the mT5 model, which is trained for each regional dialect, to translate the text into Standard Bengali.
- **APIs:**
The backend exposes two main RESTful APIs: one for translation (/translate) and another for transcription (/transcribe). Both APIs take in POST requests and process the input, giving back either translated text or transcribed audio.
- **Error Handling and Performance Optimization:** The system takes care of errors like missing files or unsupported file types with error messages. The backend is set up for better performance with smart memory use and quick model loading, so the system can deal with multiple requests at the same time without big slowdowns.

Chapter 10

Limitations, Failed Experiments and Future Works

10.1 Limitations

Throughout our research and experimentation with models, we encountered a number of challenges and limitations that impacted the overall functioning of the system. From audio impairments and transcription mistakes to model limitations such as token limits, each of these limitations cascaded, impacting system performance. Below we highlight the main limitations faced throughout the research with the rationale of their impacts.

10.1.1 Poor Audio Quality in the Dataset

A leading challenge faced during this research was the poor quality of the audio data sourced from the dataset. The audio was very noisy, containing background noise, such as ambient conditions and static, and overlapping speech from multiple speakers, which did not help the performance of the ASR model, such as Whisper. ASR systems like Whisper are very sensitive to noise, and noise is detrimental to their performance once they are trained. ASR systems require distinctly clear, noise-free audio for an adequate transcription. Background noise causes misunderstanding, failure to hear all words spoken, and therefore mistakes in transcription due to ambiguity. The overlapping voices also made it difficult for even the ASR model to decide which speaker was speaking and, thus, only served to compound the issues noted above. It can be understood that the transcript often sounded garbled or unrecognizable, with specific passages unrecognizable, and this negatively affected the quality of the overall system.

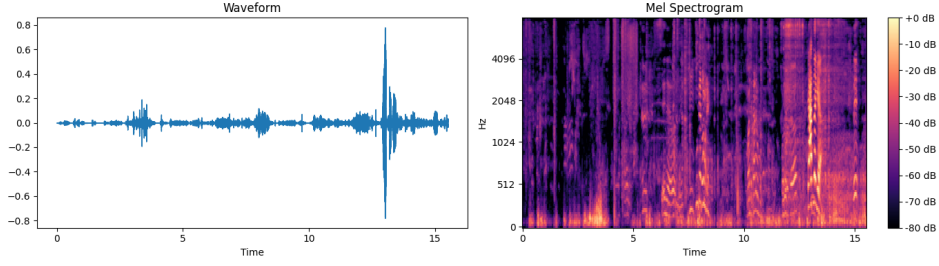


Figure 10.1: Sample waveform and Spectrogram of a Kishoreganj Audio

10.1.2 Substandard Transcription Quality

The poor quality of audio data also led to multiple errors in transcription as derived from speech. While the ASR model is a powerful model, producing accurate readable text was difficult given that the accuracy and readability relies on the quality of the input. Given the amount of noise and unclear speech represented in the dataset, the transcription outputs regularly included numerous errors, mistakes in word identification, sentence structures, and incomplete or parts of phrases. The transcripts of such poor quality reduced the capacity to use the transcriptions for any further tasks, such as regional identification or translation. In short, transcription was the agent of influence, we could not continue any of the subsequent or downstream steps of the pipeline because each downstream task (such as translations) rested on the fidelity of the transcription.

```
<> নামটা জানি কিতা? <> মেস্বারিও গাইছে এখবার। <> আজকে শিউলিরে দেইক্কা মনে ওইছে।  
রূপবানু? ওয়। ও আইছা, ই বেটির ফুরি বুঝি? আমি দি ছিনি না। ই বেটির বাফের বাড়ি ওইলো  
<>
```

Figure 10.2: Substandard Transcription Quality

10.1.3 Incompatibility of the Transcription Pipeline with the Translation Pipeline

As a result of issues with transcription quality mentioned previously, the transcription's integration into the translation pipeline was not as successful as it could have been. It is important to have high-quality transcriptions in order to produce accurate translations, particularly when dialects or terms that are region-specific are included in the transcription. As a result of receiving flawed transcription from the speech recognition system, the final output from the translation system also suffered. When there is poor quality in a transcription, regional identification can easily be incorrect and terms not translated properly, among other issues. Because of the lack of reliable, accurate, high-quality transcriptions that were provided to the translators, the whole translation process was very limited, the translation system was left to assist with the inaccurate transcriptions that were given to them.

10.1.4 Token Length Limitation in Whisper ASR Model

Another major limitation encountered during this research was the restriction of token length implemented in the Whisper ASR model. Whisper and other deep-learning based ASR models are designed to operate on input audio in chunks defined by the maximum token characteristic. For Whisper, it is 448 tokens. Given the relatively large audio datasets we worked with, the token limitation became a problem, because long audio chunks had to be truncated into smaller-sized chunks along the way. This was an important issue because elements of context, and all other important elements in speech are lost in the process of chunking due to the token limitation. The token length limitation ultimately reduced the model's precision and resulted in recognition loss of what could have been cognitive elements details for accurate speech recognition. Additionally, the implementation of a system that cannot recognize long audio sequences in one system pass and must be truncated down to smaller segments, added confusion for maintaining contextual coherence across the entire audio file.

10.2 Failures and Observations

10.2.1 Learnings from trying to use Wav2vec2 to transcribe Bangla dialect audio

Initially Wav2vec2 was the model of choice for transcription for the task of transcription of Bangla dialects as it had no limitation regarding audio-transcription lengths and our speech data consisted of multiple lengthy conversations. However, trying to train it on the initial dataset yielded very bad results, with WER and CER being close to 1.0 during evaluation, seemingly demonstrating that the model was incapable of learning Bangla dialects at any significant rate. Reasons for this could include the fact that the Wav2vec2 model was pretrained on clean and formal speech datasets, unlike our noisy and dialectic speech dataset. It is also known to be not particularly good with multiple speakers speaking at the same time, and our dataset has multiple overlapping speakers. Wav2vec2 also has an architectural drawback of being only an encoder, requiring separate decoders which do not quite capture the nuances of Bangla transcripts, while other models such as OpenAI's Whisper model, contain both encoder and decoder parts.

10.2.2 Learnings from trying to use Whisper-Small

Next we picked Whisper-Small as our model of choice for the task of transcription. While having the architectural edge over Wav2vec2, it still had a hardcoded sequence length limit of 448 tokens per transcription, requiring the rest of the transcript to be truncated. This was a problem as much of our training data was useless as the model would not learn very much. This sequence length limit resulted in very poor performance on inferences. Leading us to try multiple ways to get around the 448 token limit problem. At first we thought enlarging the embedding dimensions and mapping the inputs to the new embedding space would lift the maximum capacity

limit of 448 tokens, but even after doing so, Whisper-small seems to be hardcoded to detect input sequences that exceed 448 tokens and raise errors. Thus this method was a dead end. On careful analysis it was revealed that the WhisperTokenizer would generate many more tokens for the same number of Bangla words than it would for other languages like English, as Indic languages are more nuanced and the tokenizer generates multiple subword tokens to capture its complexity in full, easily crossing the 448 token limit per transcription, even though Whisper can process around 30 second audio files. Trying to use a custom tokenizer also did not work. Current tokenizers for Bangla are not very efficient. All of this is to say that Whisper generated poor results. Better than Wav2vec2 but not nearly good enough.

10.2.3 Getting rid of punctuations to generate less tokens

To reduce the number of tokens generated and keep more learnable information in the token limit, we removed all punctuations and special symbols like “<>” which was used to mean unclear audio. We discovered that the outputs were not significantly better, even if more of the 448 tokens were contextual information.

10.2.4 Learnings from trying to split speech audio using Voice Activity Detection

Next we chose to split the audio into smaller chunks, around 7 to 15 seconds long each, in an attempt to generate transcripts that would be within the 448 token limit. However, this required the usage of a Voice Activity Detection (VAD) algorithm, as splitting the audio midway through a word would cause wrong transcriptions to be learned by the model. The theory was that the VAD would generate audio files perfectly based on gaps in the speech. But upon closer inspection of the split files, it was revealed that many audio files existed where the split was made midway through a word. Reasons include the noisiness of the audio dataset and lack of clarity of voice, multiple speakers speaking simultaneously along with high amount of background noise. Listening to some of the audio revealed that clearer audio files were in fact split correctly and unclear ones were split very poorly. Aside from the splitting of the dataset, another problem that had arisen was the fact that the transcriptions were now not aligned to the audio files and needed some sort of alignment.

10.2.5 Failures in trying to use Forced Aligners to align text

Forced aligners are rule-based softwares that align text and audio to match with each other. Many of them generate timestamps given an input speech audio and an input text transcription. No forced aligner we found supported Bangla as a language. We tried using Aeneus, a rule-based software that converts the speech audio to a spectrogram and uses another software named Espeak, which generates phonemes from graphemes from its internal dictionary, to generate timestamps for audio-transcript datasets at a word level chunksize. However, the Espeak software library lacked a Bangla phoneme-grapheme dictionary, resulting in Espeak not being

able to generate any timestamps for our Bangla dialect audio-transcript split dataset.

10.2.6 Failures in trying to use Whisper-generated word counts to align text

After failing at forced aligners, we thought of another way to align our split dataset: using word counts from whisper generated transcripts from the split audio to count words from the true transcript and split the dataset. The idea was that even if Whisper generated messy words for speech audio, it would have the right number of words which could be then matched to split the original audio. However, Whisper-generated transcripts often had more or less words than the transcriptions given in the dataset, often because there were actually more or less words in the audio in the background, highlighting the transcription quality problem within the given dataset. Many times the Whisper model would not understand the given audio and would fall back on transcribing the same word or phrase in a loop, resulting in erroneous word counts, while also making the word count idea fall apart.

10.2.7 Dataset problems discovered while manually trying to align text

We then tried to manually align every audio chunk transcription manually, by listening to the audio ourselves and curating the text transcriptions. The huge dataset consisting of data from three regions: Habiganj, Kishoreganj and Rangpur, required many days of listening and curation. In doing so we realized that the audio was very noisy, had multiple overlapping speakers and many segments of speechless noise, making us realize that the dataset was itself not fit for training ASR models very well in many cases. We also realized that Whisper-Small was picking up the speech from children and women much more clearly than male speech, indicating that higher frequencies play a role in speech detection in ASR models.

10.2.8 Learnings from training Whisper-Small on curated-split dataset

Training Whisper-Small on the split and now aligned dataset yielded much better results with much lower median CER rates during inference. This proved that the preprocessing and curation steps increased the model’s learning.

10.2.9 Learnings from training Tugstugi on curated dataset

Training Tugstugi, a pretrained Whisper-Medium model on the curated-split dataset yielded much better results than Whisper-Small, indicating that more parameters in the model yielded better learned features with the best prediction so far, thus this pretrained model after being fine tuned on the split dataset became our proposed model. It too, however, is not perfect, and will often generate wrong results, being

far from the goal we set out to achieve in Bangla ASR technology.

10.2.10 Transcription Dataset and Translation Dataset misalignment problem

The two datasets we had had more or less the same transcripts, taken from the same audio files. However, while the transcripts in the transcription dataset were complete transcriptions of the audio file they corresponded to, the translation dataset had the transcriptions broken down into sentences and their translations. As such, sentences that would repeat if a larger transcription was broken down were not repeated in the translation dataset. In trying to create a pipeline which would need a true label for complete transcriptions instead of sentence-wise chunks, we faced a lack of any such complete translations.

10.2.11 Failure to merge Translation Dataset transcripts to align with Transcription Dataset

The first method we tried was using algorithms to automatically align and merge multiple lines of translations for each whole and complete transcription. But the main problem we faced was that searching and matching Bangla text had many issues including normalization problems and zero length character issues. Besides this, constructing a translation dataset that had the whole translation for an entire transcription from the transcription dataset proved impossible due to how sentences that were repeated were sometimes repeated in the translation dataset but sometimes not, along with the fact that sometimes the transcription data in the two sets would not match one to one. Thus we found no way to automatically align the two datasets or generate ways to calculate the performance of the intended pipeline.

10.2.12 Failure to measure intended pipeline's Performance

Due to the lack of alignment between the datasets, after building the pipeline, we had no way of determining CER from inferences, leading to manual checks and no standardized methods to measure the performance of the pipeline. Due to merging translation data by hand we had some data we could use to infer and measure performance for testing the pipeline on Kishoreganj data but they were very small in number.

10.2.13 Lack of quality in translations generated by Intended Pipeline

Due to inaccurate transcriptions generated by the ASR part of the pipeline, and the lack of training on such inaccurate transcripts by the translation model, the errors in output generated by the final translations by the pipeline was of very low quality, leading to this pipeline being something we intend to work on and improve upon in

the future. For now, we built BanglaMUX which consists of two separate pipelines inside.

10.2.14 Challenges in making a usable frontend user interface for the BanglaMUX pipeline

Creating a user-friendly user interface created a new set of challenges since the results from the pipeline might sometimes be very erroneous as the ASR model would loop and generate huge numbers of repeating words or phrases while the translation model would sometimes translate into words not having much semantic meaning. These are things we intend to improve upon in the near future.

10.3 Future Works

The Research has been successful in providing information about the obstacles and potential paths forward for Bangla speech recognition, transcription and translation systems. There are many avenues to improve and extend the findings of this research. What follows are the key directions of work, which focus on expanding the system and improving its overall performance.

10.3.1 Addressing Token Length Limitations with IPA

One future approach for alleviating the token length limitation with the Whisper ASR model can be done by utilizing the International Phonetic Alphabet (IPA) during preprocessing. Instead of leveraging audio to become transcription, the audio could first be converted to an IPA representation. IPA takes the token space, and limits it by compressing the phonetic structure of the speech, and therefore the number of tokens for the model to process the input. After the audio is converted to IPA symbols, then the ASR model can create the transcription from the IPA representation. In this method, it will address the problem of token length areas, enabling the model to manage longer audio sequences with less data dilution.

10.3.2 Splitting Audio More for Token Optimization

To better leverage the use of lower token models and increase performance, an additional enhancement would be to split the audio dataset into much shorter sections. Long audio files can be split to make segments shorter and therefore more manageable so that ASR models with lower token capacities can operate on segments without having token truncation problems. Smaller chunks would also increase transcription accuracy since the model could pay more attention to smaller and more discrete audio snippets with less distraction and more clarity over where contextual information in the audio would begin and end. Furthermore, smaller segments of speech will have less possibility of losing context, which would assist in increase

transcription correctness. Finally, splitting audio like this would entail using intelligent chunking algorithms, and it is routine for a computer to chunk the data while meaning and coherence is communicated through each of the splits.

10.3.3 Complete Speech-to-Translation Pipeline

A primary future goal will be the full pipeline process the entire flow from dialectal audio input, region detection, transcription to standard Bangla translations. After a dialect is detected and a transcription is generated, the system would be capable of automatically constructing a standard form of Bangla from transcription. This would be a seamless approach to convert spoken Bangla from different regions to a written, standardized form. It will be crucial that the transcription and translation models work together as a system that is accurate at each processing stage and to develop improved dialect detection mechanisms and train more robust regional models are critical to achieving system objectives.

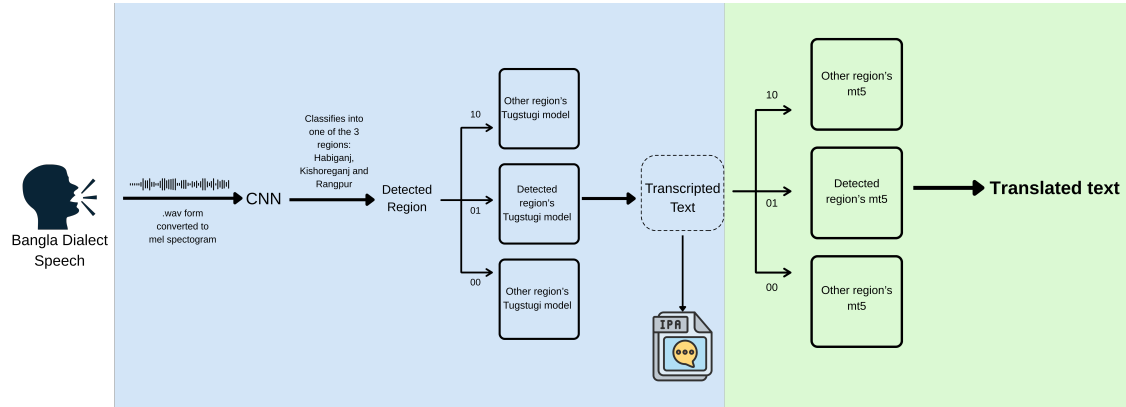


Figure 10.3: Intended Pipeline

10.3.4 Developing a Speech-to-Speech and Translation System

An ultimate aspiration for future work may be to construct an entire speech to speech corpus which takes audio (in Bangla), recognizes the dialect and region, transcribes speech to standard Bangla, translates Bangla to English, and ultimately produces speech in English. This way, the system could perform a more generalized transformation from spoken-Bangla, in the dialect of the region, to English-speech. A system such as this could be applied in a variety of instances of multilingual communication in real time by benefiting from the potential to remove language barriers and improve access. The greatest challenges for this type of system are providing a speech-to-speech synthesis system, providing contextually accurate translation, and maintaining the naturalness of speech in the complete pipeline.

Chapter 11

Conclusion

The aim of this study is to compare the low resource language dialects so that they can be accurately translated into their standardized versions which emphasizes the importance of appreciating the regional sociolinguistic variation particularly in Bangla and its implications within the communicative context. In order to analyze dialect speech and establish effective translation models, we carried out a technical literature review of various processing methods of low resource languages including dialect handling. These key references, selected from the mostly cited works and the most current articles, made up the background literature that formed our background investigation. In order for the readers to understand the details of our studied topics, we provided the background knowledge and literature review in our paper. To facilitate the understanding of our research progress, we provided a Gantt chart showing the overall schedule of all the planned activities. In the next steps of our research, we will focus on using the latest technologies from the field of Machine Learning as well as NLP to formulate and implement the system with better accuracy and efficiency to analyze and translate various regional Bangla dialects to its standard form. Thus, we expect that our study will be useful for NLP and network science communities as the foundation for advancing the retrieval of low-resource languages for translation purposes and improving the information exchange within multicultural societies.

Bibliography

- [1] T. T. Showrav, “An Automatic Speech Recognition System for Bengali Language based on Wav2Vec2 and Transfer Learning”, arXiv preprint arXiv:2209.08119, 2022, [Online]. Available: <https://arxiv.org/abs/2209.08119>.
- [2] F. T. J. Faria, M. B. Moin, A. A. Wase, M. Ahmmed, M. R. Sani, **and** T. Muhammad, “Vashantor: A Large-scale Multilingual Benchmark Dataset for Automated Translation of Bangla Regional Dialects to Bangla Language”, arXiv preprint arXiv:2311.11142, 2023, [Online]. Available: <https://arxiv.org/abs/2311.11142>.
- [3] 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”, arXiv preprint arXiv:2411.10879, 2024, [Online]. Available: <https://arxiv.org/abs/2411.10879>.
- [4] A. K. Paul, D. Das, **and** M. M. Kamal, “Bangla Speech Recognition System using LPC and ANN”, 2009 Seventh International Conference on Advances in Pattern Recognition, pp. 171–174, IEEE, Feb. 2009.
- [5] M. A. Ali, M. Hossain, **and** M. N. Bhuiyan, “Automatic Speech Recognition Technique for Bangla Words”, International Journal of Advanced Science and Technology, vol. 50, pp. 51–60, Jan. 2013.
- [6] N. J. Kalita, **and** B. Islam, “Bengali to Assamese Statistical Machine Translation using Moses (Corpus Based)”, arXiv preprint arXiv:1504.01182, 2015, [Online]. Available: <https://arxiv.org/abs/1504.01182>.
- [7] A. Gutkin, L. Ha, M. Jansche, K. Pipatsrisawat, **and** R. Sproat, “TTS for Low Resource Languages: A Bangla Synthesizer”, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pp. 2005–2010, May 2016.
- [8] P. Godard, G. Adda, M. Adda-Decker, J. Benjumea, L. Besacier, J. Cooper-Leavitt **and others**, “A Very Low Resource Language Speech Corpus for Computational Language Documentation Experiments”, arXiv preprint arXiv:1710.03501, 2017, [Online]. Available: <https://arxiv.org/abs/1710.03501>.
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez **and others**, “Attention is All You Need”, *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [10] J. R. Saurav, S. Amin, S. Kibria, **and** M. S. Rahman, “Bangla Speech Recognition for Voice Search”, *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pp. 1–4, IEEE, Sep. 2018.

- [11] K. Abe, Y. Matsubayashi, N. Okazaki, **and** K. Inui, “Multi-dialect Neural Machine Translation and Dialectometry”, *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, 2018.
- [12] M. A. Hasan, F. Alam, S. A. Chowdhury, **and** N. Khan, “Neural Machine Translation for the Bangla-English Language Pair”, *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, IEEE, Dec. 2019.
- [13] S. Siddique, T. Ahmed, M. R. A. Talukder, **and** M. M. Uddin, “English to Bangla Machine Translation using Recurrent Neural Network”, arXiv preprint arXiv:2106.07225, 2021, [Online]. Available: <https://arxiv.org/abs/2106.07225>.
- [14] M. Zampieri, P. Nakov, **and** Y. Scherrer, “Natural Language Processing for Similar Languages, Varieties, and Dialects: A Survey”, *Natural Language Engineering*, vol. 26, no. 6, pp. 595–612, 2020.
- [15] H. R. Milon, S. N. U. Sabbir, A. Inan, **and** N. Hossain, “A Comprehensive Dialect Conversion Approach from Chittagonian to Standard Bangla”, *2020 IEEE Region 10 Symposium (TENSYP)*, pp. 214–217, IEEE, Jun. 2020.
- [16] A. Bhattacharjee, T. Hasan, K. Samin, M. S. Islam, M. S. Rahman, A. Iqbal, **and** R. Shahriyar, “BanglaBERT: Combating Embedding Barrier in Multilingual Models for Low-Resource Language Understanding”, arXiv preprint arXiv:2101.00204, 2021, [Online]. Available: <https://arxiv.org/abs/2101.00204>.
- [17] M. Kowsher, M. J. Uddin, A. Tahabilder, N. J. Prottasha, M. Ahmed, K. R. Alam, **and** T. Sultana, “BnVec: Towards the Development of Word Embedding for Bangla Language Processing”, *Int. J. Eng. Technol*, vol. 10, no. 2, p. 95, 2021.
- [18] C. Yi, J. Wang, N. Cheng, S. Zhou, **and** B. Xu, “Applying Wav2Vec2.0 to Speech Recognition in Various Low-Resource Languages”, arXiv preprint arXiv:2012.12121, 2020, [Online]. Available: <https://arxiv.org/abs/2012.12121>.
- [19] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant **and others**, “mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer”, arXiv preprint arXiv:2010.11934, 2020, [Online]. Available: <https://arxiv.org/abs/2010.11934>.
- [20] S. Sultana, M. S. Rahman, **and** M. Z. Iqbal, “Recent Advancement in Speech Recognition for Bangla: A Survey”, *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, 2021.
- [21] S. Sarker, “BNLP: Natural Language Processing Toolkit for Bengali Language”, arXiv preprint arXiv:2102.00405, 2021, [Online]. Available: <https://arxiv.org/abs/2102.00405>.
- [22] A. Bhattacharjee, T. Hasan, W. U. Ahmad, K. Samin, M. S. Islam, A. Iqbal **and others**, “BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla”, arXiv preprint arXiv:2101.00204, 2021, [Online]. Available: <https://arxiv.org/abs/2101.00204>.

- [23] S. K. Sahoo, B. K. Mishra, S. Parida, S. R. Dash, J. N. Besra, **and** E. V. Tello, “Automatic Dialect Detection for Low Resource Santali Language”, *2021 19th OITS International Conference on Information Technology (OCIT)*, pp. 234–238, IEEE, Dec. 2021.
- [24] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale **and others**, “ByT5: Towards a Token-Free Future with Pre-Trained Byte-to-Byte Models”, *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022.
- [25] M. R. I. Tomal, T. Kader, A. K. M. Masum, **and** M. K. A. Chy, “Bangla Language Dialect Classification using Machine Learning”, *2022 4th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, pp. 1–4, IEEE, Dec. 2022.
- [26] P. S. Hossain, A. Chakrabarty, K. Kim, **and** M. J. Piran, “Multi-Label Extreme Learning Machine (MLELMs) for Bangla Regional Speech Recognition”, *Applied Sciences*, vol. 12, no. 11, p. 5463, 2022.
- [27] A. Chowdhury, K. T. Deepak, **and** S. M. Prasanna, “Machine Translation for a Very Low-Resource Language-Layer Freezing Approach on Transfer Learning”, *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*, pp. 48–55, Oct. 2022.
- [28] A. Bhattacharjee, T. Hasan, W. U. Ahmad, **and** R. Shahriyar, “BanglaNLG and BanglaT5: Benchmarks and Resources for Evaluating Low-Resource Natural Language Generation in Bangla”, arXiv preprint arXiv:2205.11081, 2022, [Online]. Available: <https://arxiv.org/abs/2205.11081>.
- [29] O. Kuparinen, **and** Y. Scherrer, “Dialect Representation Learning with Neural Dialect-to-Standard Normalization”, *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pp. 200–212, May 2023.
- [30] S. Badawi, “A Transformer-Based Neural Network Machine Translation Model for the Kurdish Sorani Dialect”, *UHD Journal of Science and Technology*, vol. 7, no. 1, pp. 15–21, 2023.
- [31] S. M. Islam, S. Ahmmed, **and** S. H. Mustakim, “Transcribing Bengali Text with Regional Dialects to IPA using District Guided Tokens”, arXiv preprint arXiv:2403.17407, 2024, [Online]. Available: <https://arxiv.org/abs/2403.17407>.
- [32] Bengali.AI, “ASR for Regional Dialects – Ben10 Competition Dataset”, <https://www.kaggle.com/competitions/ben10/data>, 2024, Accessed: 2025-06-18.