Automatic Subtitle Generation for Bengali Multimedia Using Deep Learning

by

Ehsanur Rahman Rhythm
22241163
Shafakat Sowroar Arnob
20101129
Rajvir Ahmed Shuvo
20141003

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 and Engineering

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
September 2023

Declaration

It is hereby declared that

1. The thesis submitted is 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:

Rhythm

Ehsanur Rahman Rhythm
22241163

Arnob

Shafakat Sowroar Arnob
20101129

Shuvo

Rajvir Ahmed Shuvo
20141003

Approval

The thesis titled "Automatic Subtitle Generation for Bengali Multimedia Using Deep Learning" submitted by

1. Ehsanur Rahman Rhythm (22241163)
2. Shafakat Sowroar Arnob (20101129)
3. Rajvir Ahmed Shuvo (20141003)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on September 21, 2023.

Examining Committee:

Supervisor:
(Member)

Sifat

Sifat E Jahan
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Annajiat Alim Rasel
Digitally signed by Annajiat Alim Rasel
DN: cn=Annajiat Alim Rasel, o=Brac University, ou=CSE Department, email=annajiat@bracu.ac.bd, c=BD
Date: 2023.01.14 23:04:00 +06'00'

Annajiat Alim Rasel
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

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

Head of Department:
(Chair)

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

Abstract

For audio or video material to be more inclusive and accessible, automatic subtitle generation is essential. Nevertheless, implementing this technology into Bengali presents significant challenges due to scarce resources and linguistic difficulty. In this study, a new deep learning based system for creating Subtitles for Bengali multimedia automatically is introduced. The suggested approach makes use of the Wav2vec2 and the Common Voice Bengali Dataset, a large collection of Bengali audio recordings. This study uses the Common Voice Dataset Bengali to train and tune the Wav2vec2 model in order to accurately convert Bengali audio into text. Current automatic speech recognition approaches are combined with Bengali language-specific factors in the created system to give accurate and reliable transcription works. The transcribed text is synced with the matching audio parts throughout the subtitle production process. The produced subtitles are enhanced using post-processing approaches, similar to capitalization and punctuation restoration, to ensure readability and consistency. The findings of this study might greatly improve Bengali language media's usability and availability across a range of sectors. The created subtitles may enhance the watching experience for Bengali multimedia by easing greater understanding, and expanding availability. The study demonstrates the potential of using deep learning and ASR methods to get over the difficulties of automated subtitle production in the Bengali language, advancing multimedia availability and inclusion.

Keywords: Automatic Subtitle Generation, Bengali Audio, Deep Learning, Common Voice Dataset, Wav2Vec2, Automatic Speech Recognition, Natural Language Processing

Dedication

In loving memory of Tanjib Ahmed (2001-2022), whose brilliance and warmth continue to inspire us.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ADSM	Acoustic Data-Driven Subword Modeling
API	Application Programming Interface
ASG	Automatic Subtitle Generation
ASR	Automatic Speech Recognition
BPE	Byte Pair Encoding
CER	Character Error Rate
CS	Code-Switching
CTC	Connectionist Temporal Classification
GPU	Graphical Processing Unit
HMM	Hidden Markov Models
LSTM	Long Short-term Memory
MLE	Maximum Likelihood Estimation
NLP	Natural Language Processing
PASM	Pronunciation-Assisted Subword Modeling
RNN	Recurrent Neural Network
SaaS	Software as a service
SOTA	State-of-the-Art
SRT	SubRip Text
WAV	Waveform Audio File Format
WER	Word Error Rate
XLSR	Cross-Lingual Speech Representations

Chapter 1

Introduction

1.1 Problem Statement

With the increasing amount of audio and video content being produced in Bengali, there is a growing need for accurate and efficient automatic subtitle generation as the tedious process of manually creating subtitles is time-consuming. Notably, 34.6% of Bangladesh's population (49.2 million) is affected by hearing loss, making automated subtitles a daily necessity [6]. Additionally, to date, there is no research in the field of Automatic Subtitle Generation for Bengali content. Moreover, the unique phonetic and linguistic characteristics of Bengali further complicate the development of robust machine-learning models for automatic subtitle generation. The goal of this research is to develop an effective deep learning-based system for automatic subtitle generation for Bengali language content.

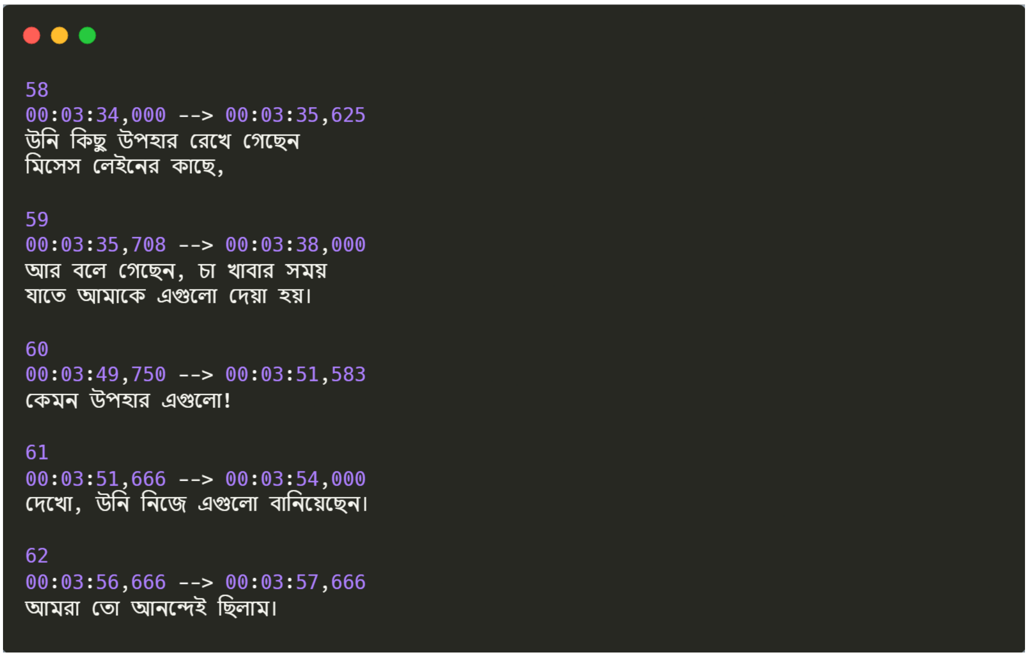
1.2 Background Information

The process of automatically creating subtitles for audio or video is referred to as Automatic Subtitle Generation. A transcript of the audio being played is presented in the form of text that is projected over the video and shown as subtitles. Persons who are deaf or hard of hearing may utilize them, as well as persons who are learning a new language. It has grown more significant in today's digital age, which is characterized by the predominance of audiovisual information across a variety of platforms, including films, television programs, internet videos, and resources for e-learning. They make it possible to localize the content, which opens up audiovisual material to a wider audience in a variety of locations and languages.

This Automatic Subtitle Generation makes use of sophisticated algorithms of deep learning to properly recognize audio and position subtitles independently at the appropriate places in the video. This technology enables users to save time and effort by automatically creating subtitles, and it also gives them the ability to update and translate those subtitles as necessary.

The application of these systems to the Bengali language has not yet reached its full potential, even though significant progress has been achieved in the creation of automated subtitle production systems for some other languages. The country of Bangladesh uses Bengali as its official language, while India recognizes Bengali as one of its 23 official languages. Bengali is one of the languages that is spoken the most in the Indian subcontinent. Bengali-specific automated subtitle-creation technologies are few, despite the enormous number of Bengali speakers and the cultural relevance of the Bengali language. This presents a substantial barrier to access and participation for Bengali speakers.

There are several challenges associated with automatic subtitle generation for Bengali audio. One of the major challenges in developing an effective Automatic Subtitle Generation system for Bengali language content is the lack of large-scale datasets of Bengali audio and subtitles. Datasets are essential for training and assessing deep learning models because they provide the necessary examples and patterns for the models to learn from. However, Bengali audio data is scarce compared to other commonly spoken languages. Without access to large-scale datasets, the development and enhancement of automatic Bengali subtitle generation systems that perform well on a wide variety of Bengali audio content may be hampered.



```
58
00:03:34,000 --> 00:03:35,625
উনি কিছু উপহার রেখে গেছেন
মিসেস লেইনের কাছে,

59
00:03:35,708 --> 00:03:38,000
আর বলে গেছেন, চা খাবার সময়
যাতে আমাকে এগুলো দেয়া হয়।

60
00:03:49,750 --> 00:03:51,583
কেমন উপহার এগুলো!

61
00:03:51,666 --> 00:03:54,000
দেখো, উনি নিজে এগুলো বানিয়েছেন।

62
00:03:56,666 --> 00:03:57,666
আমরা তো আনন্দেই ছিলাম।
```

Figure 1.1: Sample Bengali Subtitle.

The complexity of the Bengali language presents an extra challenge to developers who are tasked with developing an automatic subtitle-generation system for material written in Bengali. The word combinations, tense systems, and case systems as well as the sentence structures of this language are quite complex. There are many different linguistic variants in Bengali, and each of them conveys a distinct and subtle meaning. When it comes to writing a suitable subtitle, having a firm grasp on the correct verb form is necessary.

In addition, the Bengali language has a vast array of variants as a result of factors such as accent, dialect, and mood. Words may be pronounced differently based on a variety of circumstances, such as the speaker's regional or cultural background. Word pronunciations may change depending on which of Bengali's various dialects a person employs. As a result, producing subtitles in Bengali of good quality may be rather difficult.

Although study into the development of automated subtitles has been studied for several languages that are widely spoken, the possible relevance of such research to Bengali has not been examined to the same extent. Therefore, researchers in this field generally have to start from the beginning when doing their own groundwork rather than drawing on the work of others in the field. Because there hasn't been much work done in this area, developing an efficient system for automatically generating subtitles for Bengali-language videos may be more difficult.

Despite these obstacles, there has been some improvement in the area of automatic speech recognition in the Bengali language, which lays the groundwork for the creation of an efficient system for the generation of automatic subtitles for material written in the Bengali language. We can now create subtitles with a level of accuracy that is satisfactory thanks to the development of a variety of Speech Recognition systems that are based on deep learning.

The end objective is to improve accessibility and inclusion for Bengali speakers in many fields such as the media, education, and entertainment, while also laying the groundwork for more study into the automated production of subtitles for other languages with little resources.

There is still a lot of work to be done in the field of research that is relatively new, which is automatic subtitle production. There is currently no one working on the process of automatically generating subtitles in the Bengali language. This is a big hole in the literature since Bengali is a widely used language that is spoken by more than 260 million people all over the globe.

1.3 Research Objectives

The primary purpose of this thesis is to create an automated subtitle creation system for Bengali audio that is both efficient and accurate, and it will do so with the use of deep learning methods. More specifically, the study hopes to accomplish the following goals:

1. Explore and analyze available resources and datasets for Bengali audio and subtitles, with a focus on using the Common Voice Dataset Bengali for training and evaluating the subtitle generation model.
    - Look into different sources and collections of Bengali audio and subtitles that researchers have used in the past.
    - Check how good these datasets are in terms of size, diversity, and whether the audio and subtitles match up.
    - Dive into the Common Voice Dataset Bengali, a big collection of various languages, and see how it can be useful for training and testing the subtitle generation model.
    - Investigate any issues or difficulties that might come up when using the Common Voice Dataset Bengali, and come up with possible solutions.
2. Find the best ASR models to accurately transcribe Bengali audio into text.

- Read up on the latest research about automatic speech recognition (ASR) models and techniques that are specifically designed for processing Bengali.
- Identify and evaluate the most advanced ASR models that have proven to be accurate in converting Bengali audio into text.
- Consider both traditional models and newer deep learning-based models, like recurrent neural networks (RNNs) or transformer models, and compare how well they perform in creating subtitles.

3. Create a deep learning-based system for automatic subtitle generation, using techniques like speech recognition, natural language processing, and sequence-to-sequence modeling.

   - Develop a step-by-step process for generating subtitles automatically, starting from the audio input and ending with the final subtitle output.
   - Explore and choose the right natural language processing techniques that can handle Bengali text effectively, like breaking it into words (tokenization), finding root words (stemming), or labeling the parts of speech.
   - Experiment with different sequence-to-sequence models, such as encoder-decoder setups with attention mechanisms, to generate subtitles based on the transcribed audio.
   - Evaluate and fine-tune the subtitle generation model using measures like WER or CER to make sure it's accurate and does a good job.
   - Take into account any challenges that might come up during the subtitle generation process, such as dealing with multiple speakers, handling background noise, or dealing with words that are not commonly used.

Chapter 2

Background & Literature Review

2.1 Automatic Speech Recognition

Figure 2.1 illustrates how Automatic Speech Recognition over the past century has evolved and highlights the increasing dominance of Transformer-based models in the last decade. Even though Bengali is one of the most spoken languages in the world, very little research has been done in this domain. One of the more recent works was done by Showrav [41]. He worked with a dataset that contained 399 hours worth of samples from 19,817 contributors. He used a pre-trained model named IndicWav2Vec [31]. It is a Wav2Vec 2.0 model which was fine-tuned from the "facebook/wav2vec2-large-960h-lv60-self" model. He achieved a Levenshtein Mean Distance of 3.819 in the test dataset.

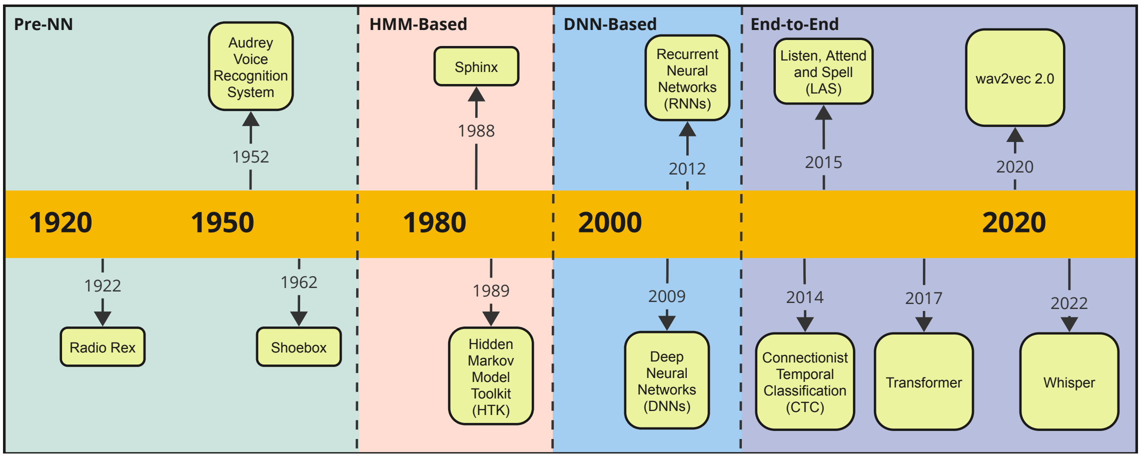

Pre-NN
HMM-Based
DNN-Based
End-to-End
1920
1950
1980
2000
2020
1922
Radio Rex
1952
Audrey Voice Recognition System
1962
Shoebox
1988
Sphinx
1989
Hidden Markov Model Toolkit (HTK)
2009
Deep Neural Networks (DNNs)
2012
Recurrent Neural Networks (RNNs)
2014
Connectionist Temporal Classification (CTC)
2015
Listen, Attend and Spell (LAS)
2017
Transformer
2020
wav2vec 2.0
2022
Whisper

Figure 2.1: Evolution of Automatic Speech Recognition over the past century.

Sayan et al. used a CTC-based CNN-RNN model for the Bengali Automatic Speech Recognition task [24]. Two CNN blocks were suggested: a two-layered Block A and a four-layered Block B, with the first layer being a 7x3 kernel and the succeeding levels all being 3x3 kernels. They benchmarked and assessed the Bengali ASR task performance of seven deep neural network topologies with varied degrees of complexity and depth using the publicly available Large Bengali ASR Training data set. With Block B, our best model has a WER of 13.67, which is 1.39% lower than similar models using bigger convolution kernels of sizes 41x11 and 21x11.

Back in the day, Hidden Markov Models (HMM) were the most prevalent models in the domain of speech recognition. Later came, the Connectionist Temporal Classification (CTC) model. However, the frame-independence assumption made by HMMs [1] and CTC [4] proved to be a problem in certain scenarios. Seq2Seq on the other hand is a general framework for neural machine translation and other sequence generation tasks. It can be used for speech recognition and it removes the unreasonable frame-independence assumption made by the earlier models [13]. Even though Seq2Seq is quite a competent model for speech recognition it suffers from drawbacks like slow training speed and requires huge computational resources. In regards to that, a no-recurrence sequence-to-sequence model was introduced by Linhao et al. which obtained a 10.92% WER after 1-day training [13].

Moreover, several recent research studies have been made regarding ASR in the past. A study by Park et al. presents a data augmentation method called SpecAugment, which is specifically designed for ASR tasks [17]. The method is based on warping the time and frequency axes of the audio signal and applying masking and frequency masking to the signal. The authors show that this method can significantly improve the performance of ASR systems, especially for low-resource languages and noisy environments. Using 1D time-channel separable convolutions, Kriman et al. present QuartzNet, a deep neural network architecture for ASR tasks [16]. This architecture is shown to be computationally efficient while still achieving state-of-the-art performance on a variety of ASR benchmarks, as demonstrated by the authors.

Some researchers have proposed end-to-end deep learning models that map the raw audio to the text without resorting to a phoneme or word recognition step first. These models include Listen, Attend, and Spell (LAS) and Connectionist Temporal Classification (CTC). These models have proven to be especially useful in situations where there are limited resources available, such as when dealing with a local dialect or a noisy environment.

Deep learning techniques have significantly improved the performance of Automatic Speech Recognition (ASR) systems in recent years. Recurrent neural networks (RNNs) and long short-term memory (LSTM) networks are frequently employed because they are effective at modeling sequential data, such as speech. Convolutional neural networks (CNNs) are also a viable option for extracting features from unprocessed audio data.

The sequence-to-sequence has been implemented using LSTM and GRU but Yu Zhang et al. on the other hand came up with very deep convolutional networks to come up with significantly improved results [10]. Their best model achieved a WER of 10.53% without any language model which was significantly better than other models at that time. They used the Wall Street Journal ASR dataset for their experiment.

A very common problem with speech recognition is the quality and availability of the dataset and the training time. Some languages do not have enough resources in such cases, for example, the language that we are focusing on that is, Bengali. Sun et al. in their research worked around ways to improve the efficiency of low-resource speech recognition [28]. Without any transfer learning or data augmentation methods and with only more or less 10 hours of data, they achieved an 8% improvement in CER compared to the baseline models.

Speaking of datasets, Das et al. contributed to creating a dataset for the Bengali language. Their dataset was divided into two sections based on age groups [3]. One included people ranging from 20-40 years old and another from 60-80 years old. They used a toolkit very popular back in the day known as the Hidden Markov Model Toolkit for aligning the dataset. The result of the aged people was worse than that of the young ones.

Cho et al. also had contributions to the low-resource speech recognition area. Seq2Seq as a model even though have been providing very good results but has an issue with the huge amount of data it requires [12]. They used transfer learning for 4 BABEL languages from the 10 languages they used to build a Seq2Seq model. It proves to be a significantly better approach. On the other hand, during decoding, they also used RNNLM with the model they built earlier on. This integration also provides a significantly better result. Their target languages were Assamese, Tagalog, Swahili, and Lao with the best results in Lao with a %WER of 57.9.

Dutta et al. addressed the task of post-editing in Automatic Speech Recognition (ASR) systems, specifically focusing on correcting phonetic and spelling errors [37]. The proposed approach utilizes a pre-trained sequence-to-sequence model called BART, which is adaptively trained to function as a denoising model for error correction. An augmented dataset is used for adaptive training, which includes synthetically induced errors and actual errors from an existing ASR system. Additionally, a word-level alignment-based approach is proposed for rescoring the outputs. Experimental results on accented speech data demonstrate the effectiveness of the strategy in rectifying ASR errors and improving the Word Error Rate (WER) compared to a competitive baseline. However, it is noted that the proposed model has limitations in capturing a wider context, as evidenced by a negative result obtained in the grammatical error correction task for the Hindi language.

Gulati et al. introduced the Conformer, a model that combines the strengths of both transformer and CNN models for ASR tasks. It effectively captures both local and global dependencies in audio sequences. According to the authors, the Conformer achieved state-of-the-art accuracy on the LibriSpeech benchmark, with a WER of 2.1%/4.3% without a language model and 1.9%/3.9% with an external language model on test/test_other. Its ability to extract robust and accurate features from speech samples improves the performance of voice cloning systems, and its efficient design also reduces computation time [23].

The system proposed by Moritz et al. employs triggered attention in the encoder and time-restricted self-attention in the decoder, it is an end-to-end ASR system based on transformers, specifically designed for streaming ASR applications where a result needs to be produced as quickly as possible after each spoken word [25]. The system's word error rate (WER) is 2.8% on the "clean" test data and 7.3% on the "other" test data, representing state-of-the-art performance on the LibriSpeech benchmark. The proposed streaming transformer architecture could be used as an ASR component of the system to transcribe the speech of a target speaker in real-time. The proposed system would work well in applications where real-time transcription is needed. Additionally, the triggered attention mechanism of the transformer-based architecture could be used to improve the robustness of the system to different speaking styles and variations in the speech samples.

Open-source technologies have paved the way for easier access for everyone to proceed with their research in the speech recognition field. Of them the most popular and one of the most recent ones is the Pytorch-Kaldi Toolkit [18]. This open-source toolkit gives us a lot of freedom in this domain. It is possible to plug in our models, tune the hyperparameters, also use our custom dataset, plug in our features, and many more. Kaldi is written in C++ and to use it in Python, a wrapper named PyKaldi has been developed.

Ankit et al. worked with 3 languages of the same Indo-Aryan family for their research one of which is Bengali [33]. They used the Kaldi model and Py-Torch Kaldi Toolkits for their entire research. Two different techniques were combined in their research to get an optimal result in speech recognition. The two techniques are transfer learning and semi-supervised learning. Their proposed architecture was SincNet-CNN-LiGRU. The best WER for Bengali they reported was 11.20% whereas for Hindi it was below 6%. The acoustic model they suggested had a 25.65% performance improvement for the Hindi language. The RNNLM did not work quite well for Bengali but worked quite exceptionally for the other two languages.

Hannun et al. came up with a new approach that did not have any concept of "phoneme" or use any phoneme dictionary which is the Deep Speech model [5]. Their model had a 16% error rate on the full test data set. The Deep Speech model also contributed to handling noisy environments better than the other systems that existed back then.

Deep Speech 2, the improved version of Deep Speech marked an important milestone in the development of end-to-end speech recognition systems based on deep learning techniques. Deep Speech 2 was enhanced using much more advanced techniques using bidirectional RNNs, CNN, and CTC loss. Amodei et al. in 2016 used Deep Speech 2 for end-to-end speech recognition in Mandarin and English [7]. They achieved a remarkable result of 1% WER on the WSJ-Eval92 dataset.

Zhao et al. address the speaker mismatch problem in automatic speech recognition (ASR) using transformer models [29]. A speaker-aware training approach is proposed, where speaker knowledge is embedded into the transformer encoder using a persistent memory model. Speaker information is represented by static speaker i-vectors, which are concatenated to speech utterances at each self-attention layer of the encoder. This creates a persistent memory that carries speaker information throughout the encoder depth. The model captures speaker knowledge through self-attention between speech and the persistent memory vector. Experimental results on ASR tasks such as LibriSpeech, Switchboard, and AISHELL-1 demonstrate significant word error rate (WER) reductions of 4.7% to 12.5% compared to other models with the same objective. The proposed model also outperforms the first persistent memory model used in ASR, achieving WER reductions of 2.1% to 8.3%.

Zeyer et al. presented a Transformer encoder-decoder-attention model for end-to-end speech recognition, which achieves competitive results with less training time compared to an LSTM model [21]. The Transformer training is observed to be more stable but prone to overfitting and generalization issues. Incorporating two initial LSTM layers in the Transformer encoder improves positional encoding. Data augmentation using a variant of SpecAugment enhances both the Transformer and LSTM models by 33% and 15% respectively. Various pretraining and scheduling schemes are analyzed, leading to improved performance. Additional convolutional layers are added to enhance the LSTM model. Experiments conducted on LibriSpeech, Switchboard, and TED-LIUM-v2 datasets demonstrate state-of-the-art performance on TED-LIUM-v2. The paper provides practical comparisons, limited training on LibriSpeech, and offers the code and setups for reproducibility.

Yue et al. focused on end-to-end automatic speech recognition (ASR) for code-switching (CS) speech, specifically addressing the challenges posed by low-resourced acoustic data [20]. The proposed ASR pipeline aims to improve the transcription of Frisian-Dutch code-switched speech archives. Two key techniques are employed: a multi-graph decoding approach that creates separate search spaces for monolingual and mixed recognition tasks to maximize the use of available textual resources, and language model rescoring using a recurrent neural network trained with cross-lingual embedding and adapted with limited in-domain CS text. Experimental results demonstrate the effectiveness of these techniques in enhancing the recognition performance of the low-resourced E2E CS ASR system.

Wav2Vec is a very popular and recent model designed for this domain which incorporates CNN. It was 1st introduced in 2019 by Schneider et al. [19]. They used unsupervised learning to improve supervised learning. Their model improved the WER by up to 36% compared to other base models. It proved to perform better than Deep Speech 2 with a WER of 2.43%.

In a research by Radford et al., they focused on improving the performance of Whisper, a speech recognition system, particularly in lower-resource languages [38]. It identifies the lack of training data as a major hurdle and proposes that increasing the amount of data for these languages can significantly enhance speech recognition accuracy. The authors suggest studying fine-tuning as a means to further improve results, especially in domains with high-quality supervised speech data. They emphasize that fine-tuning allows for better comparisons with prior work and can lead to additional enhancements. The paper also raises the question of whether Whisper's robustness is primarily due to its strong decoder, which is an audio-conditional language model. To investigate this, the authors propose conducting experiments where various components of Whisper are ablated or examining the performance of other speech recognition encoders when combined with a language model. Additionally, the authors consider the possibility of incorporating unsupervised pre-training or self-teaching methods, which are common in recent state-of-the-art speech recognition systems. While not essential for achieving good performance in Whisper, these techniques might yield further improvements.

The exponential growth of online child exploitation material poses a significant challenge for European Law Enforcement Agencies (LEAs). To address this issue, a next-generation AI-powered platform is proposed to process audio data from online sources in a timely and practical manner. The platform utilizes speech recognition and keyword-spotting techniques to transcribe audiovisual data and identify keywords related to child abuse. Two neural-based architectures, Wav2vec2.0 and Whisper, known for their accuracy, are employed for model development. Extensive testing is conducted across different languages and scenarios. To protect the sensitive data obtained from LEAs, federated learning is explored as a privacy-preserving approach to enhance system robustness. The proposed models achieve word error rates ranging from 11% to 25%, depending on the language, and demonstrate high true-positive rates (82% to 98%) in recognizing spotted words. Furthermore, federated learning strategies prove effective in maintaining or improving system performance compared to centralized trained models. The proposed AI-powered platform and the use of federated learning provide a foundation for automated audio analysis in forensic applications related to child abuse, with privacy considerations at the forefront [42].

Zweig et al. introduced advancements in CTC-based all-neural speech recognizers by proposing a new symbol inventory, an iterated CTC method, and stabilization and initialization techniques [11]. Evaluations on the NIST 2000 conversational telephony test set demonstrate superior performance compared to previous systems, both with and without external language models. The ReLU-RNN system achieves a state-of-the-art error rate of 15%, surpassing previous character-based CTC systems. While current neural-only systems have yet to match the logic of conventional approaches, the results highlight the rapid improvement in the performance of all-neural systems.

Zhu et al. explored the development of a streamable end-to-end multilingual automatic speech recognition system based on the Transformer Transducer [30]. Several techniques are proposed to adapt the self-attention architecture based on language identification. The authors analyze the trade-offs of each method in terms of quality improvements and the introduction of additional parameters. Experiments conducted on a real-world task involving five languages show relative gains of approximately 8% to 20% compared to the baseline multilingual model. Overall, the study focuses on enhancing the performance of multilingual speech recognition systems through architectural adaptations and demonstrates promising results.

Zhou et al. addressed the disparity between the objective function (maximum likelihood) and the performance metric (word error rate) used in end-to-end speech recognition models trained with Connectionist Temporal Classification (CTC) [14]. To overcome this mismatch, the authors propose joint training using both maximum likelihood and policy gradient. By leveraging policy learning, they optimize directly on the performance metric. Experimental results demonstrate that this joint training approach improves relative performance by 4% to 13% compared to maximum likelihood training alone. The proposed model achieves impressive word error rates of 5.53% on the Wall Street Journal dataset and 5.42% and 14.70% on the Librispeech test-clean and test-other sets, respectively.

The need for a fully acoustic-oriented Subword modeling approach in end-to-end automatic speech recognition (ASR) has been addressed by Zhou et al. The proposed method, called Acoustic Data-Driven Subword Modeling (ADSM), combines the advantages of various text-based and acoustic-based subword techniques [35]. ADSM utilizes a fully acoustic-oriented label design and learning process to generate acoustic-structured subword units and acoustic-matched target sequences for ASR training. Experimental evaluations on the LibriSpeech corpus demonstrate that ADSM outperforms existing approaches such as byte pair encoding (BPE) and pronunciation-assisted subword modeling (PASM) in all cases. ADSM achieves more logical word segmentation and balanced sequence lengths, making it suitable for both time-synchronous and label-synchronous ASR models. The paper also briefly discusses how ADSM can be used for acoustic-based subword regularization and handling unseen text segmentation.

Similarly, Shahgir et al. fine-tuned wav2vec 2.0 to recognize and transcribe Bengali speech [40]. They used the Bengali Common Voice Speech Dataset to train it. After undergoing 71 epochs of training, the validation set consisting of 7747 elements yielded a training loss of 0.3172 and a WER (word error rate) of 0.2524. In addition, this result was accomplished using only 17.84% of the Bengali Common Voices Dataset, which suggests that it is quite likely that even better results can be accomplished by making use of the complete dataset.

Rakib et al. also worked on the same Bengali Common Voice Speech Dataset and achieved better results than the SOTA pre-trained Bengali ASR models by optimizing a pre-trained wav2vec2 model [39]. Their fine-tuned model achieved a WER of 4.66% and a CER of 1.54% on the validation set of the common voice speech dataset.

2.2 Automatic Subtitle Generation

Deep Learning and Natural Language Processing (NLP) algorithms are used in the process of automatic subtitle synthesis, which is a method that creates subtitles for videos automatically. This may be beneficial for those who are deaf or hard of hearing, as well as those who wish to better comprehend accents or languages that they are not already fluent in [9].

The use of an ensemble-based technique is one strategy for automatically generating subtitles. This method integrates many distinct natural language processing (NLP) algorithms to increase the performance of subtitle production and video summarization [15]. Generating subtitles from audio samples is also possible with the use of machine learning strategies and high-performance computer methods, such as graphics processing units (GPUs).

In addition, there is a wide variety of software and online services accessible for the development of automated subtitles. For example, Nova A.I. is designed to create subtitles automatically, allowing users to hardcode subtitles on top of their videos or download them as SRT, VTT, or TXT files [44]. Animaker's Automatic Subtitle Maker is another AI-powered tool that recognizes speech, generates subtitles, and adds them to videos [43].

In addition to these tools, researchers have explored different techniques for improving automatic subtitle generation. For instance, a study compared various speech recognition engines in a real-time use case, finding that DeepSpeech had the lowest Word Error Rate (WER) of 26%, but CMU Sphinx was a better overall engine for the given use case when considering system resource usage [27].

Similarly, with the help of RNN, Kiran et al. generated video subtitles and indexed the scenes accordingly [32]. They discussed preprocessing audio data and extracting features as well. Che et al. generated subtitles for their lecture with the help of Automatic Speech Recognition, Sentence Boundary Detection, and Machine Translation [8]. Through their research, they found that with the help of their Automatic Lecture Subtitle generator can help human subtitle producers 54% of the time.

Their model can generate English subtitles quite accurately. They also proposed a framework to prepare bilingual subtitles (English to Chinese) without declining the quality.

In 2009, Boris Guenebaut conducted research on Automatic Subtitle Generation for Sound in Videos [2]. Although the results were not entirely successful, he made significant findings in his study. The author acknowledged that his proposed system does not present enough stability to be widely used. However, it proposes one interesting way that can certainly be improved. Similarly, Patel et al., in their research, used CMUSPHINX4 java API to recognize the extracted audio from the media, and generate subtitles from it [26]. Singh et al. proposed an automated approach to generating audio and video transcripts in multiple Indian languages [34]. Their method involves converting media speech to text and subsequently translating it.

In summary, automatic subtitle generation is an important and growing field that leverages machine learning and NLP algorithms to create subtitles for videos. There are various approaches and tools available for this purpose, and ongoing research aims to improve the accuracy and efficiency of these techniques.

Despite the growing importance of Bengali as a global language, there has been a lack of research in the field of Automatic Subtitle Generation for Bengali content. This gap in the literature is significant as it limits the accessibility of Bengali media to a wider audience. By conducting research in this area, we can improve the representation and reach of Bengali language and culture. Therefore, it is very necessary to carry out this study to satisfy this need.

Chapter 3

Dataset

3.1 Gathering Dataset

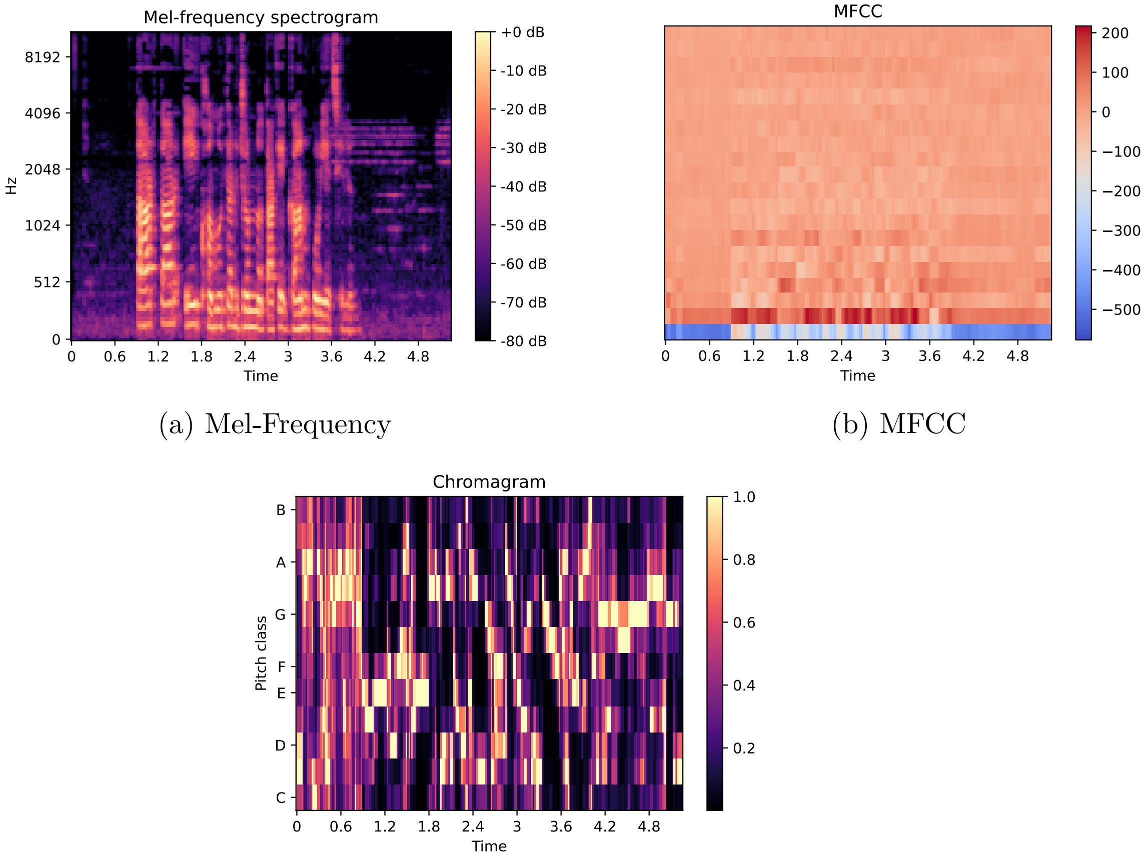
The gathering of an applicable dataset to use in the training and evaluation of our automated subtitle-generating system for Bengali audio was the first stage in the process of conducting our study. To accomplish this thing, we made use of the Bengali Common Voice Dataset that was made available by Mozilla [36]. This particular dataset was named because it includes a different range of speakers, accents, and speech styles, which enables it to be an accurate reflection of factual Bengali speech.

The Bengali Common Voice Dataset is a vast collection of multilingual speech data that was donated by volunteers who recorded their voices while reading audibly from specified textbooks. These recordings are included in the Bengali Common Voice Dataset. The dataset contains abstracts in Bengali that correlate to the audio recordings that are included in it. To train our Automatic Speech Recognition (ASR) system, the aligned audio and textbook pairs that are handed here serve as significant resources.

To get access to the dataset, we went to the official website of the Common Voice project developed by Mozilla [22]. We downloaded the dataset that was applicable to the Bengali language. Because the dataset is intimately accessible and certified under an open license, it may be used for academic study.

Because it covers such a wide variety of subjects and situations, the Bengali Common Voice Dataset is an excellent choice for training an ASR system that may be used for a variety of purposes. The data collection includes speakers of the Bengali language who are of all periods, genders, and geographical origins. This helps to ensure that the Bengali language is directly represented.

The Bengali Common Voice Dataset comprises roughly audio recordings from 22,817 contributors and has a total storage capacity of 1267 hours' worth of speech data (53 hours validated by one or more users). Because of the large size of the dataset and the wide range of voices represented, it's an excellent resource for tutoring and testing our automated subtitle-generating technique.



(a) Mel-Frequency

(b) MFCC

(c) Chromagram

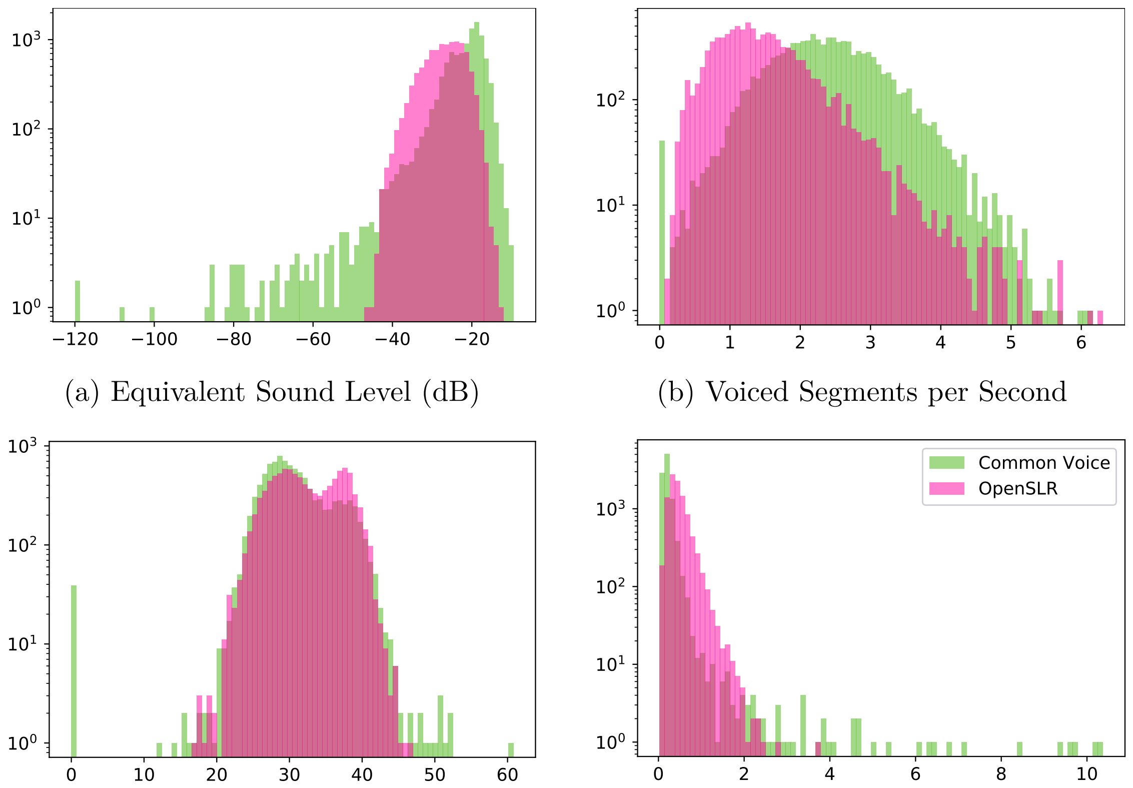
Figure 3.1: Sample Spectograms of train audio in the dataset.

3.2 Dataset Comparison

The two largest datasets for Bengali speech recognition tasks that are publicly available are the OpenSLR dataset and the Common Voice dataset by Mozilla. We chose the Common Voice dataset as when compared that was the more suitable dataset for our task. The OpenSLR dataset contains a large Bengali ASR training dataset which contains about 196000 utterances. On the other hand, the Bengali Common Voice dataset has about 1267 hours worth of data.

The Common Voice dataset demonstrates a greater diversity of features which we visualized in Figure 3.2. Alongside, histogram 3.2 exhibits a wider range of sound levels for the Common Voice Dataset.

The OpenSLR dataset has a minimum of 6.31 voiced segments per second with a median of 1.36 whereas the Common Voice dataset shows a wider range, with a minimum of 0 and a median of 2.35. This indicates that OpenSLR tends to contain speech with more consistent vocalization, possibly due to its professional or studio-based recordings, but on the other hand, common voice includes a more diverse range of speech, including instances with lower vocalization rates which might be because of more participants. Our task benefits from the diverse range of speeches of the Common Voice dataset. OpenSLR, because of studio-based recordings or controlled recording environments has a better balance to the sound levels.



(a) Equivalent Sound Level (dB)

(b) Voiced Segments per Second

(c) Mean F_0 semitone from 27.5Hz

(d) Mean Unvoiced Segment Length (Sec)

Figure 3.2: Feature Histogram comparisons between Common Voice Dataset & OpenSLR [36].

A lower dynamic range in sound levels has been exhibited by OpenSLR with an upper and lower limit of -12dB to -46 dB. On the contrary, Common Voice displays a wider range, with an upper limit of -9 dB to -120 dB. This adds to the previous reasons that we pointed out for the difference in the sound levels. In terms of pauses in the audio files, Common Voice samples have shorter pauses during vocalization. It has higher average voice segments per second because of the time constraint.

So, because of the above insights that we provided, the Common Voice dataset by Mozilla was the more suitable choice for our research purpose as it provides a lot more variety for us to work with.

3.3 Preprocessing

In our project utilizing the Mozilla Common Voice dataset, we undertook a series of critical data preprocessing techniques and steps to ensure the data's quality, consistency, and readiness for speech recognition. The dataset initially comprised various columns, including clientid, path, sentence, upvotes, downvotes, age, gender, accents, and locale.

3.3.1 Data Quality Assessment

It goes without saying the importance of a good quality of a dataset. That is why our priority was to assess and enhance the quality of the dataset. To achieve this, we considered the approach of checking the votes provided by reviewers of the public dataset, considering the upvotes (indicating favorable assessments) and downvotes (indicating unfavorable assessments). We implemented the following steps:

Vote-Based Filtering

We kept only the parts of the dataset where more people liked them than disliked them. If there were parts with an equal number of likes and dislikes or no votes at all, we didn't include them. This careful filtering based on votes made the dataset more trustworthy and better for training our speech recognition model.

Audio Data Normalization

To make sure all the audio sounded the same and could be easily compared, we used a technique called normalization. This process adjusted the volume of the audio so that it fell within a specific range, usually between -1 and 1. This adjustment was important because it helped keep the audio signals consistent and made the training process for our models more reliable and successful.

Enhanced Dataset for Speech Recognition

Through these steps, we made the Mozilla Common Voice dataset even better for speech recognition work. By using votes to filter the data, we kept only the parts that people liked and trusted. We also made sure all the audio was in a similar range, which is like keeping everything on the same page. This careful process means we now have a clean dataset ready to be used. It's perfect for training and testing speech recognition models, and it makes them work more accurately and effectively.

3.3.2 Empowering Speech Recognition Tasks

The extensive data preparation we conducted has resulted in a stronger Mozilla Common Voice dataset, ready to support speech recognition tasks effectively. This dataset improvement process, achieved through careful filtering based on votes, audio normalization, and segmenting audio into chunks, brings several important benefits:

Enhanced Reliability: We filtered the dataset based on votes to keep mostly the parts that reviewers found reliable. This step helped us remove less trustworthy or potentially incorrect data, making our training data more dependable.

Consistency and Uniformity: By normalizing the audio, we ensured that all audio waveforms were adjusted to fit within a consistent range. This consistency in how the audio data is presented is crucial for training models effectively, as it keeps things stable and helps the model learn more efficiently.

Improved Data Handling: Converting videos to MP3 format made it easier to work with audio content. MP3 audios are usually more compressed than the initial

WAV format that we received from the public dataset. It also requires less bandwidth, can be processed faster, and is computationally more efficient for us in this task. Additionally, dividing the audio into smaller, well-documented chunks, each with timecodes, made it much simpler to handle and analyze the audio, especially when generating subtitles from it.

3.3.3 Ready-to-Use Dataset

The dataset we've worked on, after going through careful preparation, has become an invaluable resource for training and testing speech recognition models. This cleaned-up dataset brings several advantages:

Accuracy: We made sure the dataset contained mostly high-quality audio recordings with accurate transcriptions. This boosts the accuracy of the speech recognition models we train with this data.

Efficiency: We made the dataset more efficient to work with. Normalizing the audio and breaking it into smaller chunks made it easier for computers to process and for models to learn from. This approach simplifies both training and analysis.

Versatility: The enhanced dataset is adaptable. This dataset can now be used in many speech recognition tasks as well and the one we are going for that is the subtitle generation task. Depending on the researcher this dataset is now a valuable asset for its versatility.

In the end, after putting in a lot of effort to make sure our data was top-notch, we've transformed the Mozilla Common Voice dataset into a rock-solid and dependable tool for speech recognition work. This improved version of the dataset after all this preprocessing that we ended up with will not only help us in this specific task but also other researchers as well.

Chapter 4

Model Backgrounds & Architecture

In this part, we describe the model that was chosen, which is called Wav2Vec2, along with an overview of the architecture that was used for the system that generated automated subtitles for this thesis.

4.1 Wav2Vec2 Model

The Wav2Vec2 model was created by Facebook AI Research as a state-of-the-art architecture for automated speech recognition (ASR) workloads that need end-to-end processing. It has achieved exceptional levels of success across a wide range of languages and fields. The Wav2Vec2 model has been through multiple versions, and each successive version has shown significant improvements in terms of both its performance and its efficiency. Along with base models, wav2vec2 has some pre-trained versions that are trained on large-scale multilingual corpora, which makes it applicable to a variety of languages, including Bengali.

In addition to the models that have already been trained, Wav2Vec2 also offers basic models that may be used as a foundation for further modification and adaptation to meet the requirements of certain downstream jobs. These foundational models have not been trained on any one particular dataset or language; instead, they provide the flexibility to be fine-tuned using data that is relevant to a certain domain.

In addition, Facebook AI Research has developed a multilingual version of Wav2Vec2 known as XLSR (cross-lingual speech representations). XLSR was developed to learn speech representations that are applicable across different languages. After that, the XLS-R (XLM-R for Speech) models were made available to the public. These models made use of unsupervised pre-training on over half a million hours' worth of audio data, and they were able to recognize 128 different languages. The XLS-R models are available in a variety of sizes, with the maximum number of parameters ranging anywhere from 300 million to two billion. They have been pre-trained on an enormous quantity of data from several languages, which enables them to recognize a variety of speech patterns and improves their capacity to transmit knowledge across languages.

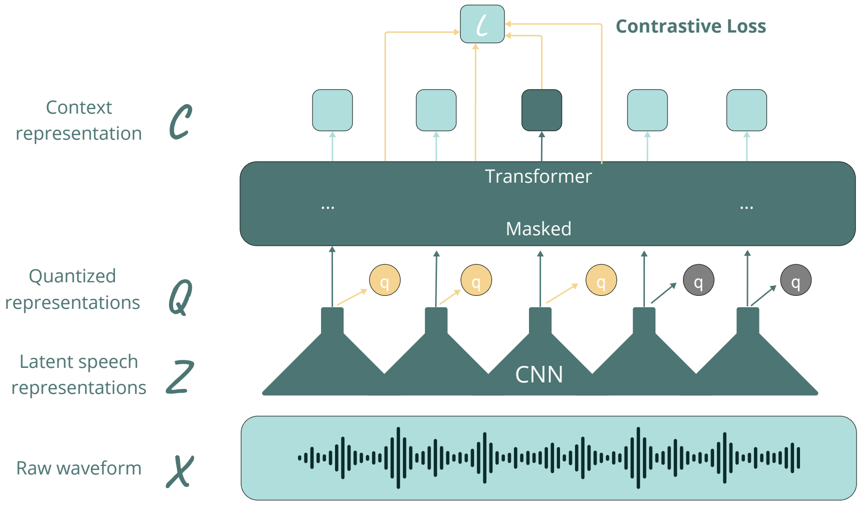


Figure 4.1: Wav2Vec 2.0 Model Architecture.

During self-supervised pre-training, XLS-R randomly masks feature vectors to prepare them for passing on to a transformer network. This process is analogous to the masked language modeling aim that BERT seeks to achieve. By going through this procedure, the model is given the ability to learn contextualized speech representations. A single linear layer is put on top of the network that has already been pre-trained to facilitate the fine-tuning of certain tasks, like voice recognition, speech translation, and audio categorization. Because of this, the model may be trained on labeled data, which enables it to be adapted to the downstream applications that are needed.

The Wav2Vec2 model is made up of two primary components, which are feature extraction and a transformer-based encoder:

1. **Feature Extraction:** The Wav2Vec2 model needs the raw audio waveform to be transformed into a high-dimensional feature representation, and here is where the feature extraction module comes into play. The procedures in this approach are intended to extract educational elements from the audio input.

   The module first performs a process known as filterbank analysis on the raw audio waveform. This entails utilizing a bank of filters to divide the audio stream into several frequency bands. The model is able to capture the spectral content and fluctuations in the spoken signal by breaking down the audio into various frequency components.

   A logarithmic compression method is used to increase the extracted features' ability to discriminate. Due to the signal's dynamic range being compressed, lower energy parts are better represented but the relative disparities between higher energy regions are preserved. This logarithmic adjustment makes it easier to properly capture both low- and high-intensity speech elements.

   Additionally, the logarithmic filterbank characteristics are normalized for mean and variance. This stage normalizes the characteristics across the audio samples, ensuring that they have a zero mean and unit variance. Any biases or variances in the feature representation brought on by variations in the recording environment or microphone characteristics may be reduced with the use of normalization.

The characteristics are further enhanced using a convolutional neural network (CNN) after being processed via filterbank analysis, logarithmic compression, and normalization. CNN uses a number of convolutional techniques to make use of its capacity to capture local dependencies and patterns in the input. To capture various levels of abstraction in the audio signal, these procedures include sliding filters over the feature maps, extracting local features, and learning hierarchical representations.

The feature extraction module converts the raw audio waveform into a rich and useful feature representation by integrating the filterbank analysis, logarithmic compression, mean and variance normalization, and CNN-based processing. The Wav2Vec2 model is able to learn contextualized representations and carry out precise automated speech recognition thanks to this feature representation, which is used as the input to the following transformer-based encoder.

2. **Transformer Encoder:** The Wav2Vec2 model, which analyzes the collected features and trains contextualized representations, has a crucial component called the transformer encoder. As a result, automated speech recognition tasks are performed better because the model is better equipped to recognize complex language patterns and relationships in the audio input.

   Transformer layers are stacked to create the transformer encoder. Self-attention and feed-forward neural networks are the two primary elements of each layer. The links between various components of the input sequence are crucially captured by self-attention systems. The model can successfully simulate long-distance dependencies and comprehend the contextual information included in the voice data by paying attention to various points in the sequence.

   In the self-attention process, the model learns to give each place in the input sequence a distinct weight or attention score. These attention scores specify how much weight should be given to each location during input processing. The model can recognize relationships between far-off pieces and comprehend the entire context of the voice data by paying attention to pertinent portions of the sequence.

   The self-attention method also enables the model to detect local and global relationships in the audio data. This is accomplished by using a multi-head attention mechanism, in which the model runs self-attention operations many times concurrently, each action paying to a separate input subspace. The model can capture both high-level and fine-grained linguistic patterns by focusing on many sequence segments at once, which enhances comprehension of speech context.

   The transformer encoder includes self-attention as well as feed-forward neural networks in each layer. These networks handle attention-weighted representation processing and non-linear transformations. The feed-forward networks' several layers of fully linked networks provide the model the ability to learn elaborate mappings and identify complex correlations in the audio data.

   The Wav2Vec2 model can efficiently extract the dependencies, linguistic patterns, and contextual information from the audio data by merging the self-attention and feed-forward networks in the transformer encoder. This makes it possible for the model to do precise automated voice recognition, making it a valuable tool for uses like automatic subtitle production.

In this study, the employment of the Wav2Vec2 model makes it possible to support correct and reliable automated subtitle synthesis for Bengali audio material. In the following sections, we will go into further detail on the architecture as well as the process of fine-tuning and subtitle generation.

4.2 Model Fine-Tuning and Post-Processing

During the process of fine-tuning the Wav2Vec2 model for Bengali automated subtitle production, a number of different strategies and components are used to maximize the model's performance and guarantee that appropriate transcription results are produced. These aspects are essential to the process of molding the model to the particular specifications of the endeavor. Let's get into more depth about each one of them.

1. **CTC Loss:** Throughout the process of fine-tuning, the Connectionist Temporal Classification (CTC) loss will be used as the training target. Taking into account the temporal dimension of voice data, this loss function enables the model to acquire the knowledge necessary to align the predicted text with the ground truth labels. When the model is optimized by the use of CTC loss, it becomes more capable of managing the sequence-to-sequence nature of speech recognition tasks. These tasks include situations in which the alignment between audio and text is not one-to-one.

2. **Phonetic Dictionary:** A phonetic lexicon is added to the model to further increase the performance of the model. This dictionary provides a mapping from individual words to the phonetic representations of those words. It does so by taking into account the distinctive linguistic qualities and variances of Bengali pronunciation. When phonetic information is included in the model, it becomes more adept at handling ambiguous or out-of-vocabulary terms, which ultimately leads to higher transcription accuracy.

3. **Acoustic Models:** To capture fine-grained acoustic nuances and fluctuations in voice input, the Wav2Vec2 model makes use of acoustic models throughout the fine-tuning process. These models are educated using vast audio datasets, which enables the model to acquire acoustic characteristics that are essential for effective speech recognition. The Wav2Vec2 model is able to better match the extracted characteristics with the associated textual representations as a consequence of the incorporation of these acoustic models into the process of fine-tuning, which ultimately results in more accurate transcriptions.

4. **Word Sequence Search (n-Gram Model):** After the phase of fine-tuning, an exhaustive word sequence search is carried out to provide correct subtitles. In this stage, a language model, such as an n-gram model, is used to search for the word sequences that are most probable given the transcribed audio. The language model takes into account the statistical features and linguistic

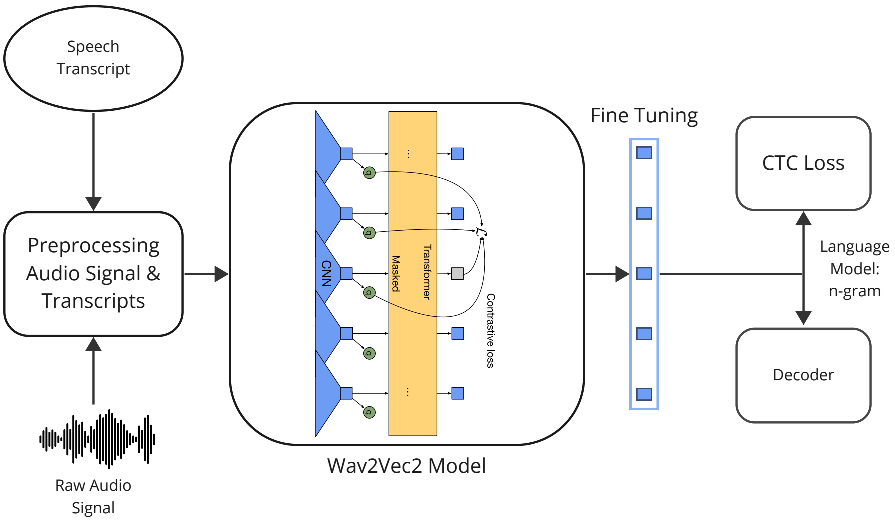


Figure 4.2: Automatic Speech Recognition Pipeline with Wav2Vec 2.0.

patterns of the Bengali language to improve the fluency and coherence of the subtitles that are created. This results in an output that is more natural and contextually appropriate.

Wav2Vec2 is a model that can be refined to provide accurate and dependable transcription outcomes for Bengali automated subtitle synthesis if these components are included in the process of fine-tuning. The model is able to accurately capture the intricacies of the Bengali language and provide high-quality subtitles that are closely aligned with the audio material as a result of the combination of the CTC loss, phonetic dictionary, acoustic models, and word sequence search.

In general, the process of fine-tuning, together with the following post-processing procedures, plays a vital role in fitting the Wav2Vec2 model to the particular needs of automated subtitle synthesis. This is because the Wav2Vec2 model is used to generate subtitles for videos. Because of this optimization procedure, the transcription will be correct, which will improve the general accessibility and usage of Bengali-language media.

4.3 Architecture Overview with Subtitle Generation

The architecture for the automated production of Bengali subtitles contains a set of components and procedures that are related to one another and work together to generate subtitles that are correct and synced with the audio material they accompany. The ASR, which employs a finely tuned version of the Wav2Vec2 model, the creation of a timecode, the rendering and assessment of subtitles, and the evaluation of the system are the essential components of the design.

1. **ASR using Fine-tuned Wav2Vec2:** The Automatic Speech Recognition (ASR) system is at the fundamental heart of the design. This system makes use of the fine-tuned Wav2Vec2 model. This model has been trained using audio data in Bengali, and after going through the process of fine-tuning, it has been adapted to transcribe spoken words into written representations of

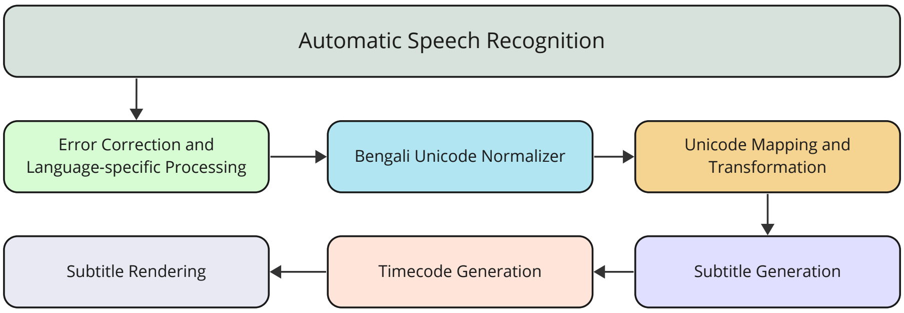

Automatic Speech Recognition
Error Correction and Language-specific Processing
Bengali Unicode Normalizer
Unicode Mapping and Transformation
Subtitle Generation
Timecode Generation
Subtitle Rendering

Figure 4.3: Automatic Subtitle Generation Pipeline.

those words. The ASR system performs a precise conversion of the audio input into text, which is then used as the foundation for the development of subtitles. It is able to do this by using its capability to capture auditory features as well as contextualized linguistic patterns. This allows it to achieve the desired goal.

2. **Timecode Generation:** Creating a timecode is a vital step in the process of matching the transcribed text with certain timestamps in the audio stream, which is the next stage in the process. Timecodes are produced by a procedure that involves analyzing the audio waveform and using the precise time information that is obtained. When the video is played again, these timecodes indicate when each subtitle should appear and when it should stop appearing. This technique ensures that the audio and the corresponding subtitles are synchronized with one another, which enhances not only the viewing experience but also the viewer's capacity to understand what they are seeing on the screen.

3. **Subtitle Rendering:** Following the generation of the timecodes, the transcribed text and the timecodes themselves become inputs for the subtitle rendering component, which then uses both of these inputs to generate the final subtitle files. The subtitles are prepared throughout the process of rendering in line with the predetermined visual standards. These criteria include suggestions for the size of the font, the color of the typeface, and the positioning of the subtitles. After the subtitles have been created, they are either embedded within the video itself or sent as separate subtitle files, depending on whatever option was chosen during production. This enables users to see the text in concert with the audio content that is being presented.

4. **Evaluation:** An evaluation process has been included in the architecture of the system from the very beginning. This was done to ensure that the subtitles that are produced are of the highest possible quality and accuracy. This involves assessing whether or not the transcribed text and the audio are in sync with one another, and also determining whether or not the subtitles are grammatically correct and readily accessible. Additionally, this includes determining whether or not the subtitles are in sync with one another. Evaluation metrics and qualitative assessments are used to track the progress of the automated subtitle creation system and highlight areas that may benefit from further development in the near or distant future.

In general, the architecture for the automated synthesis of Bengali subtitles involves an automatic speech recognition (ASR) system that makes use of the fine-tuned Wav2Vec2 model. Additionally, the design includes the generation of timecode, the rendering of subtitles, and evaluation. Accessibility, usability, and the overall viewing experience of Bengali-language media are all improved because of an integrated method that makes it feasible to create subtitles that are in time with the audio and accurate. Integrating cutting-edge ASR approaches, precise timecode generation, and fast subtitle rendering, the architecture provides a complete solution for the automatic synthesis of Bengali subtitles. This is accomplished via the use of the architecture. Because of this, it is now feasible to create Bengali subtitles in a way that is both timely and accurate.

Chapter 5

Methodology

The purpose of this study is to investigate the use of deep learning methods in the production of an automated system for generating subtitles for Bengali audio. This can be accomplished by following the sequential stages specified in the proposed work plan that is shown in figure 5.1 below.

5.1 Feature Extraction & Transformer Encoding

The accurate representation of audio data is essential for the successful training of our system that generates automated subtitles, and feature extraction plays a vital part in this. In this part, we will discuss the styles of feature extraction that were used in order to dissect the preprocessed audio data that was attained from the Bengali Common Voice Dataset.

1. **Acoustic Features:** In order to get useful aural information from the preprocessed audio data, we used the feature extraction system that's described in the following judgment:

   (a) **Mel Frequency Cepstral Coefficients (MFCCs):** MFCCs are a kind of acoustic characteristic that's used considerably in a variety of speech-processing operations, including automatic speech recognition. They do this by mapping the power spectrum of short-time frames onto the mel scale, and also performing a separate cosine transform later. This allows them to get the spectral features of the audio stream.

2. **Additional Features:** In addition to MFCCs, we contemplated the preface of fresh rudiments that would improve the representation of the audio data. This was done in trouble to make the audio data more accurate. In spite of this, we conducted a number of trials and thorough evaluations and came to the conclusion that the MFCCs supplied our ASR system with an acceptable quantum of discriminational information. As a result, we decided to concentrate the majority of our efforts on MFCCs for the definition of our features.

Python's Librosa module, which offered styles that were both effective and versatile for audio analysis and processing, was used in the implementation of the feature extraction procedure.

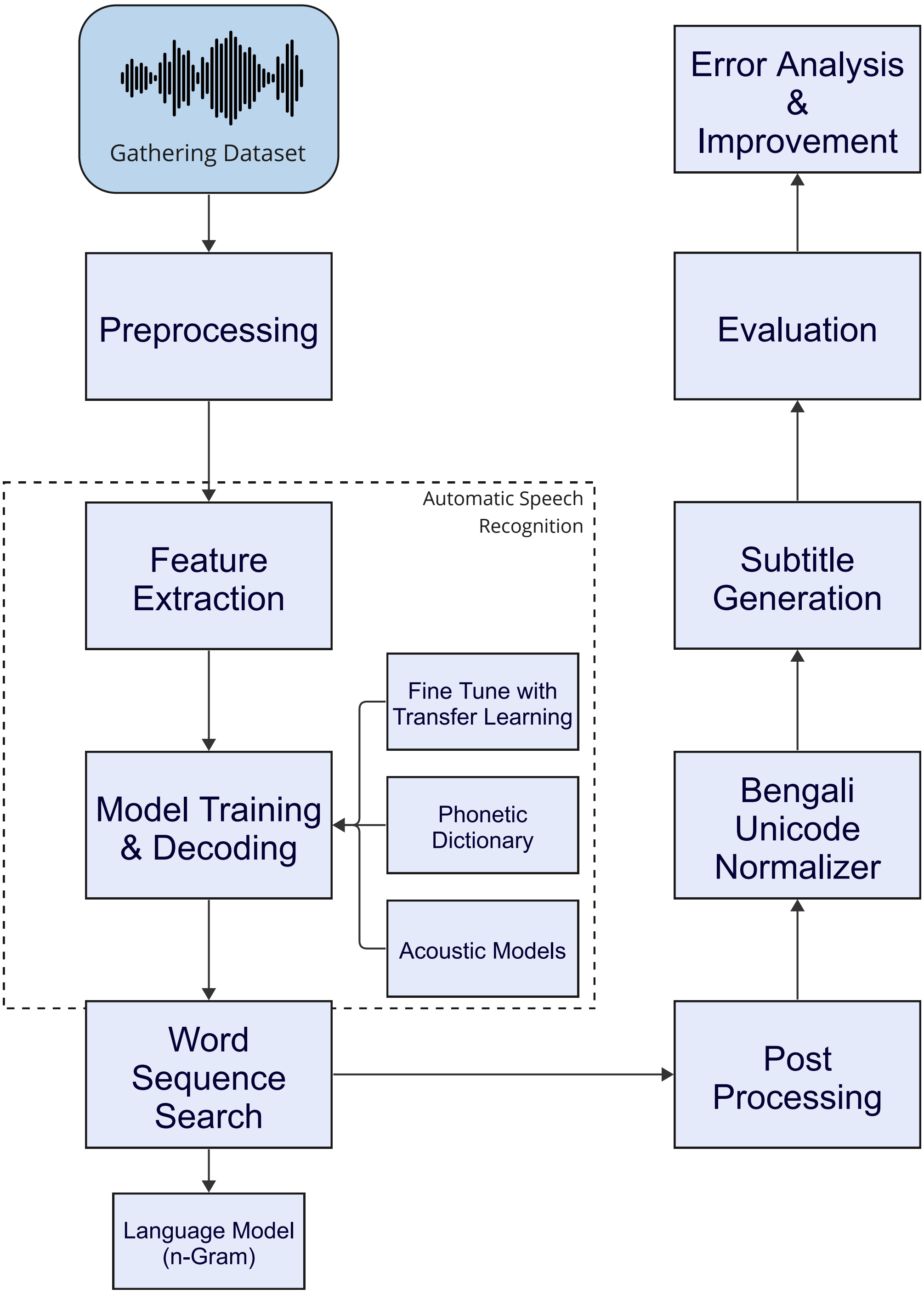


Figure 5.1: Proposed Methodology for Automatic Subtitle Generation for Bengali Audio Using Deep Learning.

In the coming part, we will talk about the procedures of model training and decoding. In these phases, the extracted features are used to train our ASR system and produce subtitles for Bengali audio.

5.2 Model Training & Decoding

In this part, we will walk you through the stages of training our automated subtitle-generating system to work with Bengali audio. These phases include decoding, as well as training our models to work with the audio. We made use of a wide variety of strategies in order to increase the ASR system's perfection and overall performance.

1. **Fine-Tune with Transfer Learning:** We decided to use a method called fine-tuning in conjunction with transfer learning so that we could make the utmost of the eventuality of models that had formerly been trained and speed up the training process. To be more specific, we used the Wav2Vec2 model, which had preliminarily been pre-trained on a massive multilingual corpus, as the starting model for our automatic speech recognition (ASR) system. During the process of fine-tuning, the model that had been trained was adapted so that it could do the unique job of automatically generating subtitles in Bengali by making use of the Bengali Common Voice Dataset.

   During the process of fine-tuning, we started training on the Bengali Common Voice Dataset while also continuing to initialize the Wav2Vec2 model with the pre-trained weights. With the help of this strategy, our model was suitable to make use of the information gained from the multilingual training data and more directly capture the material acoustic aspects of Bengali speech.

2. **Phonetic Dictionary:** We created a phonetic wordbook that's unique to the Bengali language in order to prop in the process of mapping auditory units to their phonetic representations. The phonetic wordbook offered a mapping between different words and the phonetic recap that corresponded to each word. This made it possible for the ASR system to deal with terms that weren't in its wordbook, which improved the delicacy of the subtitles that were created.

   Expertise in a number of different languages as well as hand reflection were needed for the construction of the phonetic wordbook. In order to guarantee that the phonetic representations of Bengali words are correct, we made use of the verbal materials that were formerly available, similar to wordbooks and verbal exploration. The phonetic wordbook was included in the decoding process so that the alignment of acoustic units could be improved, and the delicacy of the transcription was increased.

3. **Acoustic Models:** During the training phase of our ASR system, we made use of aural models to get an understanding of the connection that exists between the aural characteristics that were derived from the preprocessed audio data and the textual representations that correspond to those features. The audio models were constructed using neural networks. More precisely, they included a direct layer, which was also activated using a softmax function.

   During the training process, the audio models were enhanced by optimizing them using the Connectionist Temporal Classification (CTC) loss function. Because of this, the models were suitable to learn how to rightly match the acoustic characteristics with the applicable textual representation, which enabled proper transcription of the Bengali audio data.

Deep learning architectures, similar to PyTorch were used in the perpetration of the model training and fine-tuning procedures. These frameworks give effective calculation and optimization strategies.

Decoding the acoustic characteristics and producing the final subtitles both need the use of several word sequence search algorithms, one of which is the language model. These will be discussed in the coming section.

5.3 Word Sequence Search

Word sequence search algorithms were used by our team in order to match the acoustic characteristics with the word sequences that were most likely to be spoken in the Bengali audio we were assigned with transcribing. This allowed us to produce accurate and cohesive subtitles for the Bengali audio. This section provides an explanation of the language model (n-gram) approach that's used while doing word sequence quests.

1. **Language Model (n-Gram):** An automatic speech recognition (ASR) system isn't complete without a language model, which is an essential element that calculates the liability of certain word sequences being in a specific language. In our system, we made use of a verbal model known as an n-gram, which calculates the liability of a word grounded on the environment of the (n-1) words that came before it. This allowed us to make accurate predictions.

   In order to train the language model, we made use of a huge corpus of Bengali textbook data. This corpus comported of a wide variety of textual sources, including books, papers, and material set up online. Tokenization, Sentence Segmentation, and a variety of other verbal factors were addressed throughout the preprocessing of the corpus.

   We created an n-gram language model using the preprocessed corpus, using different values for n to capture different degrees of environment for the various situations of the environment. This needed the use of maximum likelihood estimation (MLE) or other smoothing approaches similar to add-k smoothing or Kneser-Ney smoothing in order to estimate the probability of word sequences.

   When it came time to decode the communication, the language model was put to use to assign points to the various word sequences that were produced by the ASR system. The scores generated by the acoustic model were coupled with the chances generated by the language model in order to establish the word sequence that's most likely to correspond to a particular collection of audile characteristics. By taking into account the audile substantiation as well as the environment of the language, this integration supported the process of producing subtitles that were more accurate and coherent.

   The n-gram language model was developed using libraries and tools similar to

NLTK or KenLM, which offer effective ways for training and querying language models. These libraries and tools were used in the implementation.

In the coming part, we will talk about the post-processing methodologies that were used to enhance the quality of the created subtitles, making them easier to read and ensuring that they were accurate.

5.4 Post-Processing

The post-processing stages are necessary for enhancing the automatically produced subtitles and assuring their quality, readability, and consonance for Bengali audio. In this part, we will discuss the post-processing styles that were used to ameliorate the quality of the subtitles that were generated by our ASR system.

1. **Error Correction and Language-specific Processing:** We used mistake correction styles in addition to language-specific processing approaches so that we might increase the verbal correctness of the subtitles that were created. These approaches include:

   (a) **Spell-checking:** During the process of subtitle development, we made use of an n-gram language model for word sequence search, as was covered in Section 5. This language model was used as a kind of post-processing for the Automatic Speech Recognition (ASR) system. We were suitable to estimate the word sequences that were most likely to do by integrating the chances from the language model with the scores from the aural model. This allowed us to take into account both the audio evidence and the environment of the language. This decoding phase helped improve the delicacy and consistency of the subtitles that were created. This is how the textbook that was created was checked for spelling errors in the Bengali language, using a wordbook of duly spelled terms to compare the textbook against and determine which words demanded to be corrected.

   (b) **Grammar and Language-specific Error Correction:** In the coming section, we will talk about the system of normalizing Bengali textbooks in Unicode, which is going to be used as a later step in the post-processing way. The purpose of this normalization is to address character encoding and other normalizing concerns that are unique to Bengali textbooks. By standardizing the representation of Bengali characters inside Unicode, we assure consistency and harmony across systems, which contributes to the accurate and applicable display of the translations. Thus, language-specific error correction strategies were used in order to resolve frequent verbal problems as well as inconsistencies that were present in the Bengali subtitles. In order to identify and correct particular problems, these styles used rule-based approaches, pattern matching, and language analysis.

2. **Formatting and Readability Enhancement:** We employed several formatting and readability enhancement strategies, similar to the following, in order to enhance the overall readability and balance of the subtitles that were created.

   (a) **Punctuation Insertion:** The character "|" seems to be the last one in the vast majority of the authentic samples. As a consequence, a "|" character has been added to the end of the text which was anticipated. Also, punctuation essentials similar to commas, periods, and question marks have been added to the subtitles in the proper places in order to enhance the grammatical structure and clarity of the subtitles.

(b) **Formatting and Styling:** The subtitles that were created were formatted duly by making use of the right line breaks, indentation, font styles, and sizes. This was done to guarantee that the generated subtitles were presented in a visually charming manner and adhered to the criteria that have been set for subtitle formatting.

5.5 Bengali Unicode Normalizer

Character encoding in Bengali text may vary very frequently, which might affect several Unicode representations being used for the same Bengali letters. In order to remedy this situation, we included a Bengali Unicode normalization procedure in the post-processing portion of our workflow. The purpose of the normalization procedure was to make the Unicode representation of Bengali characters harmonious and compatible across all platforms. This was fulfilled by the normalization of the Bengali character set.

1. **Character Encoding Analysis:** An investigation into the character encoding discrepancies that were discovered in the Bengali subtitles that were produced by our ASR system came first; this was followed by the use of the normalizing procedure. The results of this investigation helped to identify the numerous Unicode representations that are employed for the same Bengali characters. These representations include compound characters and alternate forms.

2. **Normalization Algorithm:** On the base of the exploration of character encoding, we constructed a normalization technique that's especially acclimated for Bengali text. The system addressed a variety of challenges related to normalization, including the following:

   (a) **Composite Character Decomposition:** Numerous Bengali characters may be recast as a mixture of their introductory characters and the diacritic marks that are associated with them. In order to guarantee harmonious representation across the data, the normalization algorithm broke down these compound characters into their individual element factors.

   (b) **Normalization of Alternative Forms:** Some Bengali characters may have alternate forms or variations in their Unicode representations. These forms and variations are referred to as the normalization of indispensable forms. This diversity of representations was analyzed by the algorithm, which also mapped them all into a common format for the sake of consistency.

3. **Unicode Mapping and Transformation:** In order to normalize Bengali using Unicode, we first had to develop a mapping table that connected the

various variants of Unicode representations with their separate normalized forms. During the process of transformation, the mapping table was applied to translate the various representations into their separate standardized Unicode forms.

4. **Integration into the Post-processing Pipeline:** The procedure for normalizing Bengali text to the Unicode standard was included in the post-processing channel of our ASR system. Following the phases of mistake correction, formatting, and readability enhancement, the produced subtitles were subjected to the normalization process. This procedure ensured that Bengali letters were represented in Unicode in a manner that was harmonious and compatible with other Unicode representations.

The procedure of normalizing Bengali text using Unicode was veritably important for conserving the authenticity of the Bengali language and ensuring that the subtitles were rendered correctly across a variety of devices and apps.

In the coming part, we will examine the last phase of the process of creating subtitles, which is the actual development of the subtitles based on the processed text and the matching audio timestamps. This step takes place after the processing of the text has been completed.

5.6 Subtitle Generation

The actual creation of subtitles, determined by the text that has been processed and the audio timestamps that correspond to it, is the last stage of our automated system for the development of subtitles. In this section, we will provide an overview of the methodology and processes that were used in the process of generating Bengali subtitles that are accurate and in sync with the audio. These methods and procedures were utilized in the process of producing correct and in sync with the audio Bengali subtitles. From figure 5.2, we can see how the transcribed texts are segmented within the audio reference.

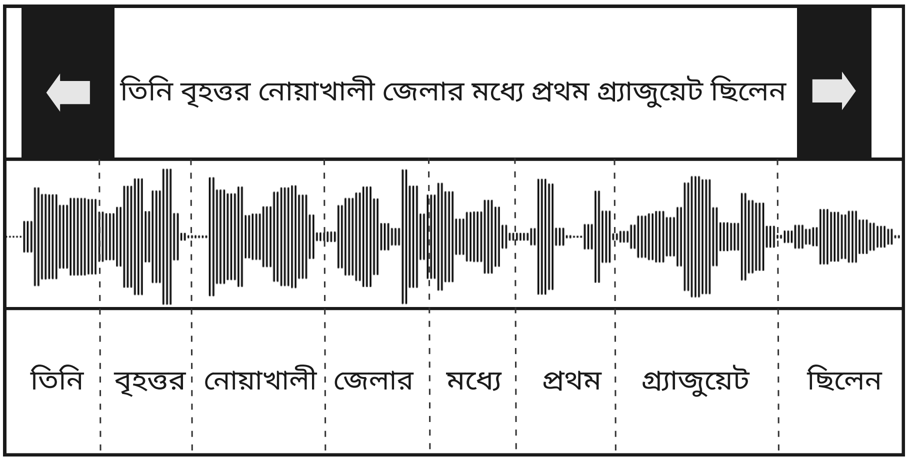


Figure 5.2: Segmented Audio with Transcribed Text by ASR.

1. **Audio Data Preparation for Subtitle Generation:** To generate the subtitles from videos, we need to extract the audio from the video first. Then, we

need to process that audio:

(a) **Converting to WAV Format:** We transformed the video files into WAV format. WAV is a lossless audio format, which preserves the original audio quality without compression and does not discard any audio data during encoding. WAV files provide greater flexibility for preprocessing and feature extraction.

(b) **Audio Chunking:** We broke down the audio into smaller pieces, like dividing a cake into slices. Each piece had a specific timecode, like a timestamp, which helped us keep track of where each part belonged in the video. This made it much simpler to work with the audio because we could deal with smaller sections at a time, like solving one piece of a puzzle before moving to the next. Also, it helped us to work with the computational resources that we had.

2. **Alignment of Text and Audio:** In order to achieve precise synchronization between the created subtitles and the corresponding audio segments, we made use of a strict alignment technique. This allowed us to achieve our goal. A time code was used in order to successfully complete this task. In order to correctly align the generated text with the audio waveform, this operation needed the use of complex alignment algorithms or methods. Finding the correct time intervals in the audio that fit each portion of the subtitles was the first step in getting everything aligned properly. Because of this, it was possible to bring the segments into the correct alignment with one another.

3. **Timecode Generation:** After making sure that the text and the audio were exactly synchronized with one another, the next step was to produce the correct timecodes for each segment of the subtitles. These timecodes contain the starting and ending timestamps for each subtitle, which indicates the length of time that the subtitle should be illustrated on-screen. Additionally, the timecodes indicate the period of time that the subtitle should be displayed.

4. **Subtitle Formatting:** When we created the subtitles for the video, we ensured that they were formatted correctly so that they were easier to read and presented in a more aesthetically pleasing manner. Utilizing these techniques, you are able to personalize not only the position of the subtitles on the screen but also their font style, size, and color, as well as the placement of the subtitles on the screen. In addition, we emphasized specific words or phrases by using formatting characteristics such as italics and boldface where it was appropriate to do so, and we indicated who the speakers were by using boldface and italics.

5. **Subtitle Rendering:** After the timecodes and layout were finalized, we produced the final subtitle output by using a rendering engine for the subtitles. This allowed us to generate the highest-quality subtitles possible. This engine generated a complete subtitle file that was suitable for use with standard subtitle formats such as WebVTT (.vtt) and SubRip (.srt). The processed text was combined with time codes and guidelines for formatting, which enabled it to accomplish this goal.

Our automatic subtitle creation system reached its peak in the generation phase, which resulted in the production of synchronized and visually pleasing subtitles for Bengali audio content. These subtitles were generated as a direct result of the generation stage of the system.

In the next section, we will discuss the evaluation technique that was carried out in order to analyze the performance and effectiveness of both our ASR system and our subtitle generation pipeline. This will be done by comparing and contrasting the two systems' respective strengths and weaknesses.

5.7 Evaluation

It is necessary to do an analysis of our automated subtitle production system in order to determine its efficacy, accuracy, and overall performance. This section provides an overview of the evaluation procedure that was used to assess the efficacy of the subtitles that were developed for Bengali audio material.

1. **Evaluation Metrics:** In order to determine whether or not our strategy was successful, we relied on a mix of quantitative and subjective evaluation criteria. The accuracy, synchronization, readability, and overall quality of the subtitles that were developed were judged based on these criteria. The following are the metrics of assessment that we used:

   (a) **Word Error Rate (WER):** The word-level error rate (WER) is determined by comparing the reference transcripts to the automatically generated subtitles. WER levels that are lower are indicative of greater accuracy.

   (b) **Character Error Rate (CER):** CER calculates the proportion of character-level discrepancies between the reference transcripts and the produced subtitles. Lower CER values indicate more accuracy, similar to WER.

   (c) **Synchronization Accuracy:** This statistic measures how well the produced subtitles and related audio segments align. It counts the proportion of time codes or subtitle lengths that are appropriately aligned.

   (d) **Subjective Evaluation:** The overall effectiveness, readability, and coherence of the created subtitles were evaluated by a team of human reviewers. Based on their interpretation and comprehension of the subtitled material, they offered arbitrary scores and comments.

2. **Evaluation Data:** We created a second assessment dataset containing Bengali audio recordings and manually transcribable reference subtitles for testing purposes. To provide complete assessment coverage, this dataset included a wide variety of speech patterns, themes, and speakers.

3. **Evaluation Procedure:** The following stages made up the assessment process:

   (a) **Data Preparation:** The evaluation dataset underwent preprocessing to ensure that it met the specifications for our automated method for creating subtitles' input format.

(b) **Subtitle Generation:** For the evaluation dataset, subtitles were created using the system.

(c) **Evaluation Metric Calculation:** To determine the assessment metrics, such as WER, CER, and synchronization correctness, the produced subtitles were compared to the reference subtitles.

(d) **Subjective Evaluation:** A selection of the produced subtitles was checked by human evaluators who gave their opinions on their quality, readability, and coherence.

4. **Findings and Assessment:** The examination of the evaluation findings was carried out in order to determine how successful our automated technique of subtitle generation is. The objective measurements, which include synchronization accuracy, WER, and CER, give information that can be quantified on the capabilities of the system in terms of accuracy and synchronization. The responses to the subjective assessment questions offered qualitative ratings on the overall quality and readability of the subtitles that were prepared.

5.8 Error Analysis and Improvement

The phase of our automatic subtitle production system that is dedicated to the study of errors and the pursuit of ways to enhance the system's overall performance is of the utmost significance. This phase is responsible for understanding what factors led to the errors that were produced and developing potential solutions to correct those factors. In this section, we will analyze the analysis of faults that were discovered during the evaluation process and propose ways to boost the precision and standard of the subtitles that are created.

1. **Error Analysis:** During the course of the evaluation, we conducted a comprehensive inquiry into the errors that were generated by our software, which is responsible for the generation of automatic subtitles. The objective of the inquiry was to get an understanding of the myriad of errors that were made, the patterns that were most often seen among them, and the factors that could have contributed to their occurrence. In order to categorize the errors that we discovered, we used the following major categories:

   (a) **Word Errors:** Errors occurred as a result of poor transcribing or recognizing words, which eventually resulted in disparities being present between the captions that were made and referenced transcripts.

   (b) **Timing Errors:** The fact that the subtitles are not synchronized with the audio that corresponds to them is the root cause of the errors that are seen. Timing discrepancies inside the subtitles are one of these problems. For example, the start or end timestamps can be wrong.

   (c) **Phonetic Errors:** The improper analysis of phonetic information may lead to inaccuracies, which, in turn, can lead to faults in the subtitles that are formed as a consequence of such interpretation.

   (d) **Formatting Errors:** Errors in the presentation and layout of the subtitles, comprising difficulties such as incorrect alignment, discrepancies in the font used, or missing aesthetic components.

2. **Improvement Strategies:** In order to improve the precision and overall quality of the subtitles that are generated, we have come up with a number of potential solutions that depend on the error trends and causes that we have uncovered. These include the following:

   (a) **Model Refinement:** Improving the performance of the Wav2Vec2 model when applied to Bengali audio by systematically fine-tuning it via the use of additional training data or specific approaches. Improving the hyperparameters, determining the possibilities of various designs, or using advanced transfer learning techniques are all examples of things that might fall under this category.

   (b) **Data Augmentation:** The training dataset will see a rise in both its depth and breadth as a direct consequence of the use of a number of data augmentation procedures. This method has a chance to assist in minimizing the effects of insufficient training data and boost the system's capacity to generalize its findings, both of which would be beneficial.

   (c) **Language-Specific Acoustic Models:** The development of language-specific acoustic models that are adapted to the distinctive phonetic features and pronunciation patterns of Bengali. This may result in better identification accuracy and fewer phonetic mistakes.

   (d) **Post-Processing Techniques:** Exploring more complex post-processing approaches, such as language modeling, with the goal of further honing the subtitles that are created. In this way, the subtitles' coherence and naturalness may be improved by using language models that are based on n-grams or more sophisticated neural language models.

   (e) **Error Correction Algorithms:** The investigation of error correction algorithms that are able to identify and fix frequent mistakes in created subtitles. In order to recognize and fix certain mistake patterns, these algorithms may make use of statistical methodologies, deep learning, or rule-based approaches.

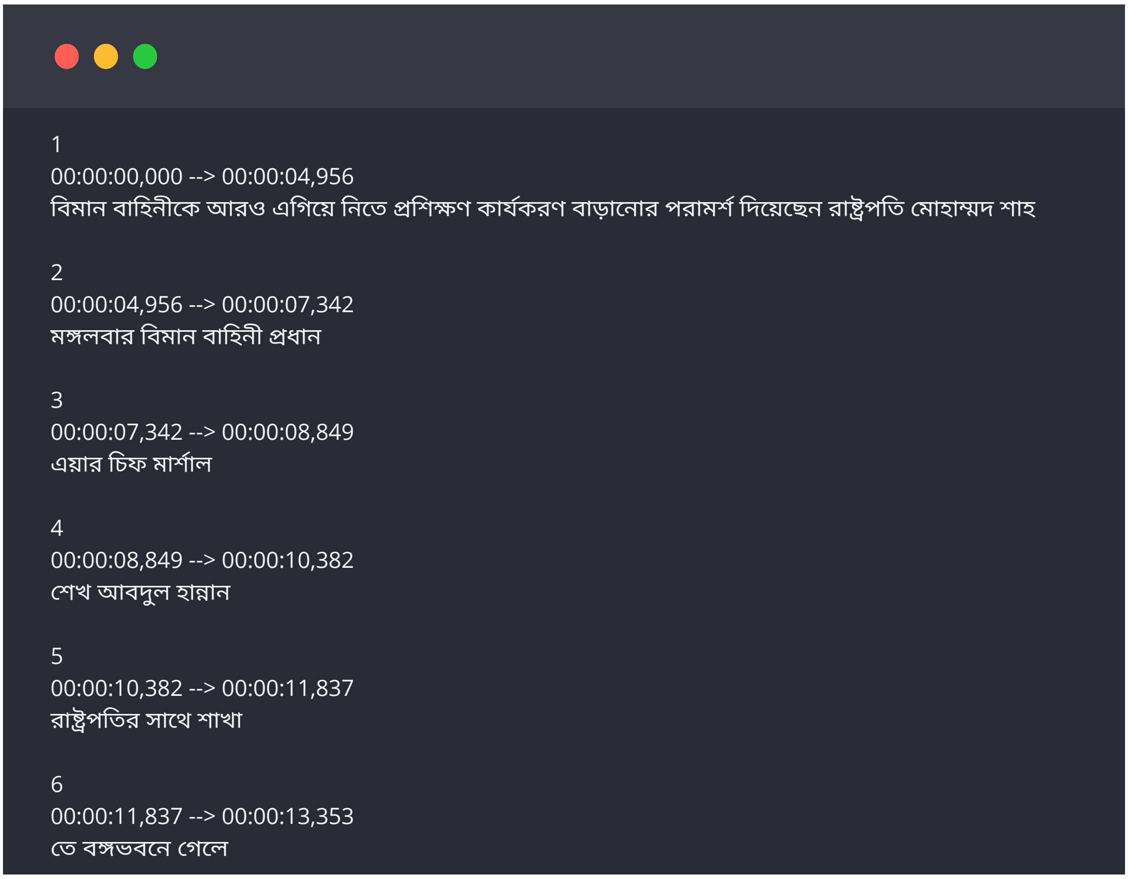
3. **Iterative Evaluation and Refinement:** We are going to carry out iterative assessments of the improved system so that we can determine how successful the recommended improvement tactics are. This entails carrying out the review procedure once again with the improved methodology and then contrasting the outcomes with the prior version. We will evaluate the effects of the adjustments by analyzing how the assessment measures, such as WER and CER, synchronization accuracy, and subjective evaluations, have changed.

We want to fix the mistakes that have been discovered and improve the accuracy, synchronization, and overall quality of the automatically produced subtitles by repeatedly reviewing and developing our system for the development of automated subtitles.

Chapter 6

Result & Analysis

A number of different approaches and strategies were used in order to achieve the objectives of the research. The Wav2Vec2 architecture, a state-of-the-art automated speech recognition (ASR) model, was used in the research. The model was fine-tuned using a substantial dataset consisting of Bengali audio recordings. The model was modified with the use of transfer learning strategies so that it would be applicable to the particulars of the Bengali language. In addition, the system contained both a phonetic lexicon and acoustic models in order to increase the accuracy of the transcription.



```
1
00:00:00,000 --> 00:00:04,956
বিমান বাহিনীকে আরও এগিয়ে নিতে প্রশিক্ষণ কার্যকরণ বাড়ানোর পরামর্শ দিয়েছেন রাষ্ট্রপতি মোহাম্মদ শাহ

2
00:00:04,956 --> 00:00:07,342
মঙ্গলবার বিমান বাহিনী প্রধান

3
00:00:07,342 --> 00:00:08,849
এয়ার চিফ মার্শাল

4
00:00:08,849 --> 00:00:10,382
শেখ আবদুল হান্নান

5
00:00:10,382 --> 00:00:11,837
রাষ্ট্রপতির সাথে শাখা

6
00:00:11,837 --> 00:00:13,353
তে বঙ্গভবনে গেলে
```

Figure 6.1: Generated Subtitle for a Bengali News Media (.srt)

In order to produce the subtitles, it was necessary to conduct an order of words search by making use of a language model, and more specifically, n-gram models that had been modified to work with the Bengali language. Post-processing techniques, such as the Bengali Unicode normalization, were used with the goal of bringing about an improvement in the consistency and quality of the subtitles that were generated. The accuracy of the ASR, the accuracy of the timecode generation, the standard of the output of the subtitles, and the overall utility of the system were some of the factors that were considered while evaluating the system's performance. Figure 6.1 shows the SRT file generated from the video by our proposed Subtitle Generation Architecture for Bengali Language.

By performing an analysis of the results obtained from these applicable methodologies and approaches, this section will provide insights into the efficacy and usefulness of the created automatic subtitle-generating device for Bengali audio content. These insights will be provided via the presentation of data.

6.1 ASR Performance

In order to evaluate how well the automatic recognition of speech (ASR) system performed, we employed the following methods:

1. **Character Error Rate (CER):** Evaluation of the accuracy of ASR systems often makes use of a statistic called the Character Error Rate, which is commonly abbreviated as CER. For the purpose of determining it, all that is required of us is to conduct a simple count of the number of alterations, such as replacements, omissions, and additions, that must be made from the text that represents the ground truth (also referred to as the source text) to the text that represents the result of the ASR system. The CER is often expressed using the formula that is as follows:

$$CER = (S + D + I)/N \tag{6.1}$$

   where S stands for the number of substitutions, D for the number of deletions, I for the number of insertions, and N for the total number of characters in the source text. The CER value represents the percentage of the source text's characters where the ASR system provided an incorrect prediction. Lower CER values are indicative of better performance, with 0 standing for a performance that is faultless.

2. **Word Error Rate (WER):** Another key indicator for assessing ASR systems is the Word Error Rate (WER), which is particularly useful in situations in which the transcription comprises paragraphs or sentences of words that have significance.

$$WER = (S_w + D_w + I_w)/N_w \tag{6.2}$$

   WER acts on the word level and counts the number of word changes, such as insertions, deletions, and replacements, that are necessary to turn one phrase into another. WER is calculated using the same algorithm as CER, however, it acts at the word level instead of the character level. In most cases, the value of the WER is greater than the value of the matching CER.

Good CER and WER values are dependent on the particular use case, and these values might change depending on variables such as the kind of material and the level of difficulty of the activity. For printed text, standards derived from extensive digitization projects conducted by Australian newspapers show that CER values of 1% to 2% indicate high accuracy, values between 2% and 10% are deemed to be average, and values over 10% indicate poor accuracy. In situations featuring handwritten writing that contains a variety of information, a CER value of about 20% may be deemed acceptable as a reasonable level.

Score	Fine-Tuned	Fine-Tuned + Bengali Unicode Normalizer
CER	0.098	0.093
WER	0.310	0.283

Table 6.1: CER & WER results of Fine-Tuned Wav2Vec2
and Fine-Tuned Wav2Vec2 with Bengali Unicode Normalizer

The wav2vec2 model has feature extraction and Transformer encoding in it. We trained and fine-tuned the model on the Bengali Common Voice Dataset by adding the CTC loss function, Phonetic Dictionary, Acoustic Models, and Word Sequence Search (n-Gram Model). After that, we post-processed and added Bengali Unicode Normalizer to get greater ASR performance. For both cases, we generated subtitles and found good differences in the accuracy.

On Bengali audio data, the fine-tuned Wav2Vec2 model demonstrated good performance, resulting in a CER score of 0.098 and a WER score of 0.310. In addition, the CER score was lowered to 0.093 after using the Bengali Unicode Normalization on top of the fine-tuned Wav2Vec2 model. The WER score also decreased to 0.283. These findings provide evidence that we can get more enhanced ASR performance on top of the fine-tuned model if we add post-processing and Bengali Unicode Normalizer with it.

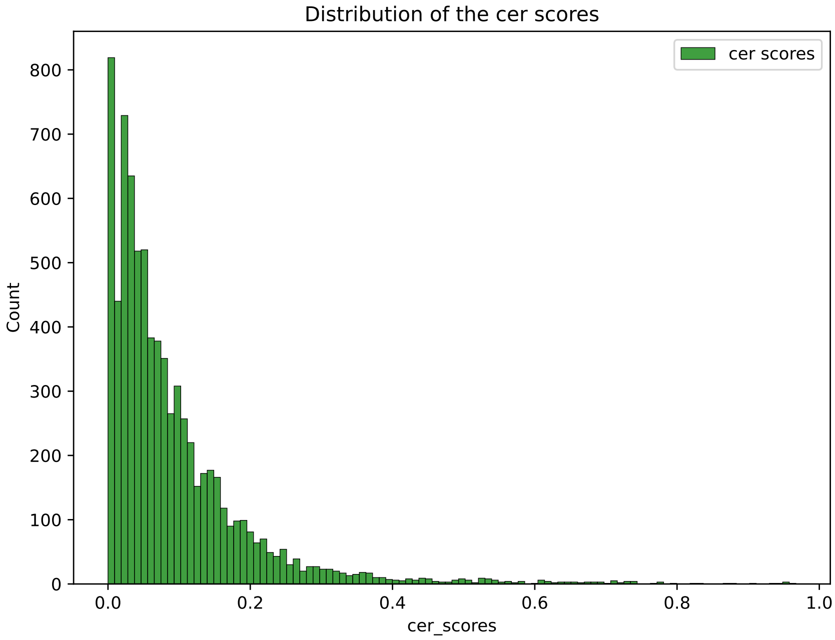


Figure 6.2: Distribution of CER scores.

In figure 6.2 & 6.3, we can see the Distribution of the CER scores which is a plot of count vs CER scores, and the CER score of the ASR Model which is a plot of CER scores of the whole validation dataset from 0 to 8000 samples. From this, we can visualize how high or how low the CER score went in our findings. It also shows how many samples have these scores in the Distribution of the CER scores graph. We can see that, the CER score range is 0 to 1 and 0.093 is the best CER score we achieved.

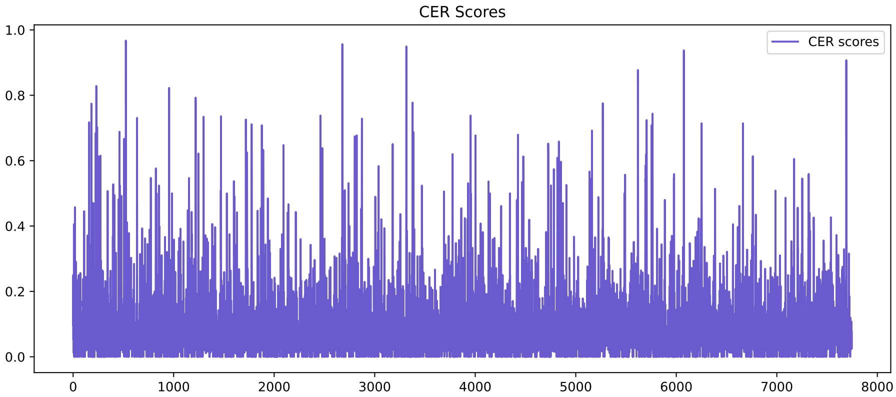


Figure 6.3: CER Scores of the ASR Model.

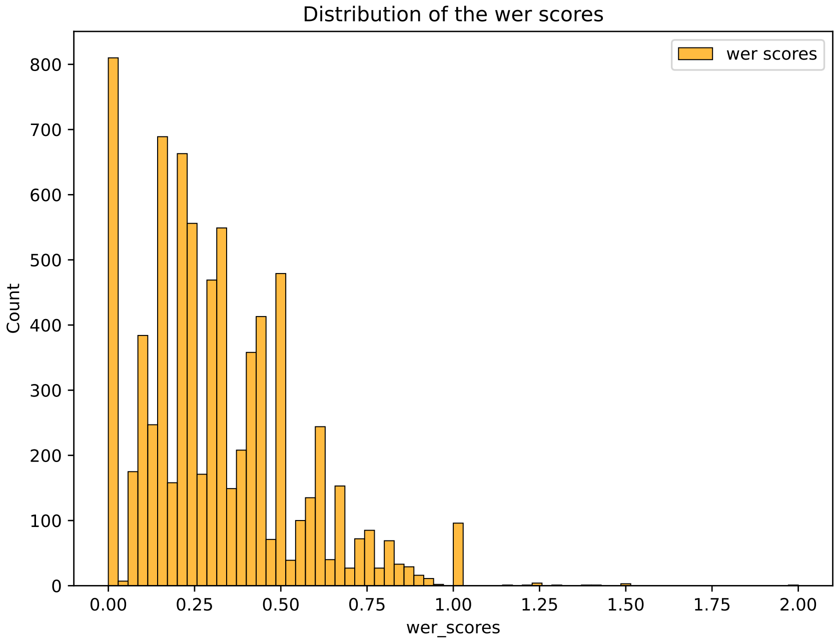


Figure 6.4: Distribution of WER scores.

On the other hand, in figure 6.4 & 6.5, we can see the Distribution of the WER scores, which is a plot of count versus WER scores, and the WER score of the ASR model, which is a plot of WER scores for the whole validation dataset, ranging from 0 to 8000 samples. This provides a visual representation of how high or low the WER score was in our results. In addition, the number of samples that have these scores is shown in the Distribution of the WER scores graph. We can see that the range of possible WER scores is 0 to 2 and the best score we were able to attain was 0.283.

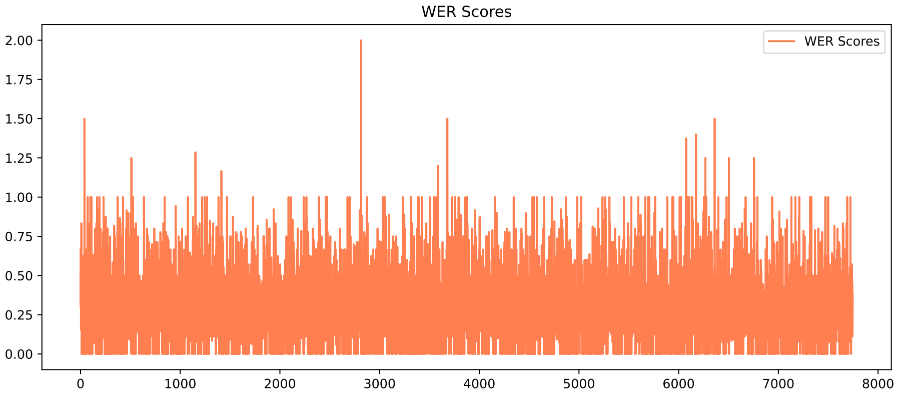


Figure 6.5: WER Scores of the ASR Model.

Predictions	References	Pred len	Ref len	CER Score	WER Score
টেস্ট খেলাগুলোর বাইরে ব্যাপক সংখ্যায় রান তোলেন।	টেস্ট খেলাগুলোর বাইরে ব্যাপকসংখ্যায় রান তুলেন।	7	6	0.0435	0.5
তিনি হিন্দু দেবদেবীদের সম্মান করেন না	তিনি হিন্দু দেবদেবীদের সম্মান করেন না।	6	6	0.0263	0.167
কিন্তু এসব জায়গার অধিকাংশ এখন সাধারণ মানুষের ভোগ দখলে।	কিন্তু এইসব জায়গার অধিকাংশ এখন সাধারণ মানুষের ভোগ দখলে।	9	9	0.0182	0.112
এবং তিনি ব্যক্তিগতভাবে তাকে মৃত্যুদণ্ড দেন।	এবং তিনি ব্যক্তিগতভাবে তাকে মৃত্যুদণ্ড দেন।	6	6	0.0	0.0
এতে রয়েছে যুদ্ধের কিছু দৃশ্য	এতে রয়েছে যুদ্ধের কিছু দৃশ্য।	5	5	0.0344	0.2
এই সম্মেলনে কুয়েতে কোনো প্রতিনিতি উপস্থিত ছিলেন না।	এই সম্মেলনে কুয়েতের কোনো প্রতিনিধি উপস্থিত ছিলেন না।	8	8	0.0384	0.25

Table 6.2: Predicted vs Reference Result.

Table 6.2 shows the Predicted vs Reference Results. In this, we can find that Our ASR model has predicted the Audio samples very accurately though some of them have spelling errors and grammatical errors. The predicted word length in a sentence is very close to the reference word length which depicts the performance level of our ASR system. However, we can find good CER and WER scores for the predicted samples by our fine-tuned wav2vec2 which outperforms most of the publicly available ASR systems.

6.2 TimeCode & Subtitle Generation

Notably, the ASR system obtained levels of accuracy and effectiveness when it came to the generation of transcriptions for audio in Bengali. A high degree of accuracy was seen in the process of our approach and turning it into text by the low CER and WER scores. However, we found that CER and WER scores is varying whenever the predicted sentence length is not similar to the reference sentence length. On the other hand, automated timecodes is varying from manual timecode annotations due to the ASR we got.

After perfecting the ASR system for our goal, we approached generating TimeCode for subtitles. We used pydub and moviepy libraries for generating the subtitles. After getting the videos, we convert them to audio and did segmentations in between the silences to make chunks of small audios for that particular file. We kept 500 milliseconds of silence at the beginning and end of each chunk for synchronizing the subtitle later. Then, we generated the start time code and end time code for each chunk. Lastly, we generated text for all the chunks through our ASR pipeline and build a SRT file with the time codes and texts accordingly.

The following metrics were used in order to assess the level of precision in the timecode and subtitle generation:

1. **Absolute Timecode Error (ATE):** The absolute time difference between the timecodes that were created by the system and the timecodes that were annotated by hand is what ATE measures. It offers an overall assessment of accuracy as well as a calculation of the time variation in milliseconds for each timecode.
2. **Relative Timecode Error (RTE):** The relative timing error (RTE) is calculated by comparing the manually annotated times with the automatically produced times. It computes the percentage of time that is off for each timecode and provides an indicator of the relative accuracy of the timecodes that have been created.
3. **Synchronization Accuracy:** The precision with which the subtitles are aligned with their respective audio tracks is referred to as ”synchronization accuracy.” It analyzes the temporal accuracy of the subtitles and determines how well they coincide with the words that are being spoken.

Lower figures of both ATE and RTE are indicative of a higher degree of precision and accuracy in the creation of the timecode. The performance of the timecode creation module was assessed using the Bengali audio dataset, with the goal of making additional improvements. The purpose of the module was to dependably give proper timecodes that were automatically synchronized with the audio sections.

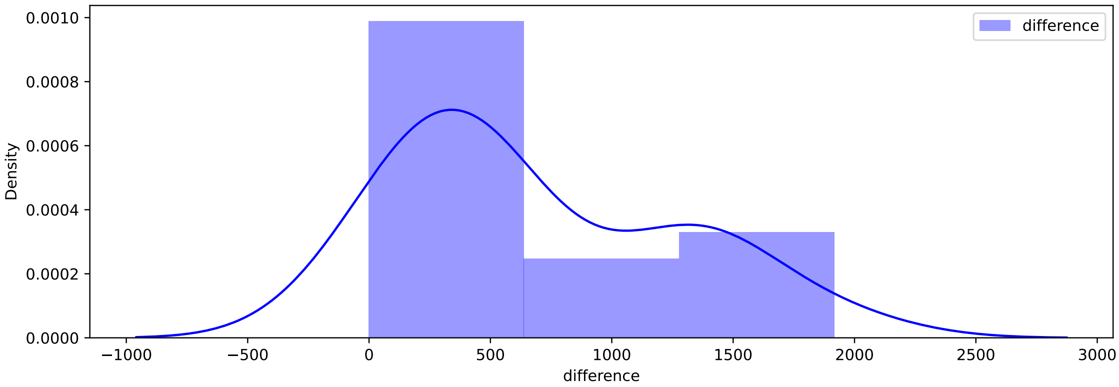
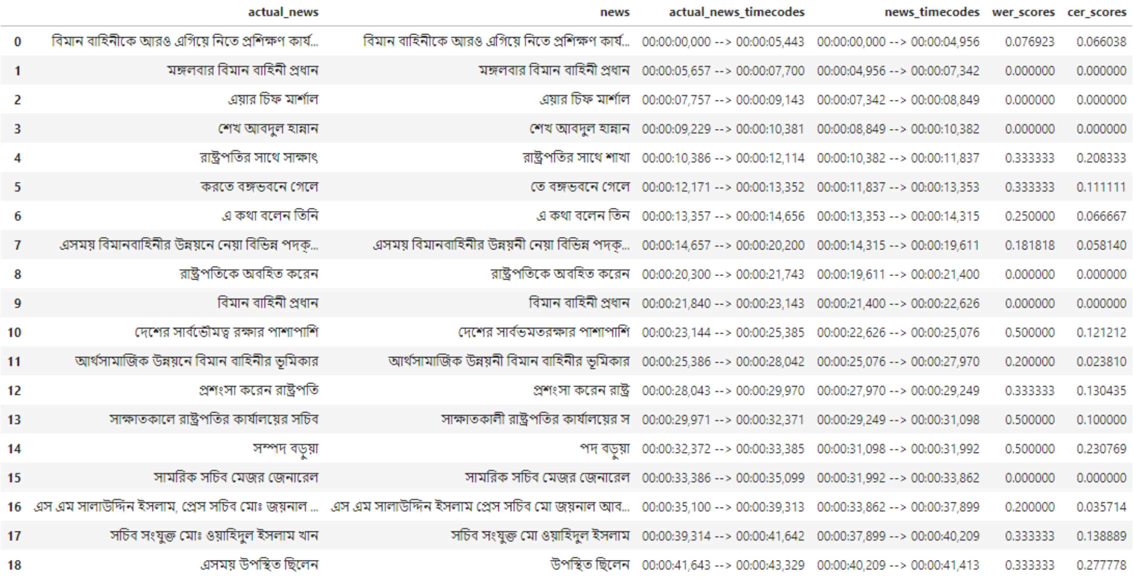


Figure 6.6: Distribution of Time difference between the Manual Subtitle vs Generated Subtitle Time Code.

In our investigation, we compared the efficacy of the automatically produced subtitles to that of the hand-crafted subtitles by determining the amount of time that elapsed between the two sets of time codes for manually made and automatically generated subtitles. Because of this, we were able to measure the inaccuracy that existed among both sets of subtitles and evaluate the effectiveness of the deep learning model in terms of its ability to generate correct subtitles. In figure 6.6, we see a visualization of the distribution of errors between subtitles that were made manually and those that were generated by an automated system. The histogram shows how often different error values occurred in our data, providing insight into the overall accuracy of the machine-generated subtitles. The figure and assessment results revealed that the module obtained a low Absolute Timecode Error (ATE) and Relative Timecode Error (RTE). The fact that the automatically produced timecodes were quite similar to the manually annotated timecodes as the difference between them is very low, demonstrated a high degree of accuracy in the process of matching the timecodes with the audio segments.

When it comes to synchronization, such as when displaying subtitles or indexing audio, having an accurate timecode generation is very necessary. Timecodes that may be relied on provide for precise navigation and synchronization of audio and text-based material. According to the results, the timecode-generating module may have the capability of improving the effectiveness and precision of a wide variety of applications, particularly those that are dependent on synchronized time-based data.



	actual_news	news	actual_news_timecodes	news_timecodes	wer_scores	cer_scores
0	বিমান বাহিনীকে আরও এগিয়ে নিতে প্রশিক্ষণ কার্য...	বিমান বাহিনীকে আরও এগিয়ে নিতে প্রশিক্ষণ কার্য...	00:00:00,000 --> 00:00:05,443	00:00:00,000 --> 00:00:04,956	0.076923	0.066038
1	মঙ্গলবার বিমান বাহিনী প্রধান	মঙ্গলবার বিমান বাহিনী প্রধান	00:00:05,657 --> 00:00:07,700	00:00:04,956 --> 00:00:07,342	0.000000	0.000000
2	এয়ার চিফ মার্শাল	এয়ার চিফ মার্শাল	00:00:07,757 --> 00:00:09,143	00:00:07,342 --> 00:00:08,849	0.000000	0.000000
3	শেখ আবদুল হান্নান	শেখ আবদুল হান্নান	00:00:09,229 --> 00:00:10,381	00:00:08,849 --> 00:00:10,382	0.000000	0.000000
4	রাষ্ট্রপতির সাথে সাক্ষাৎ	রাষ্ট্রপতির সাথে শাখা	00:00:10,386 --> 00:00:12,114	00:00:10,382 --> 00:00:11,837	0.333333	0.208333
5	করতে বঙ্গভবনে গেলে	তে বঙ্গভবনে গেলে	00:00:12,171 --> 00:00:13,352	00:00:11,837 --> 00:00:13,353	0.333333	0.111111
6	এ কথা বলেন তিনি	এ কথা বলেন তিন	00:00:13,357 --> 00:00:14,656	00:00:13,353 --> 00:00:14,315	0.250000	0.066667
7	এসময় বিমানবাহিনীর উন্নয়নে নেয়া বিভিন্ন পদক্...	এসময় বিমানবাহিনীর উন্নয়নী নেয়া বিভিন্ন পদক্...	00:00:14,657 --> 00:00:20,200	00:00:14,315 --> 00:00:19,611	0.181818	0.058140
8	রাষ্ট্রপতিকে অবহিত করেন	রাষ্ট্রপতিকে অবহিত করেন	00:00:20,300 --> 00:00:21,743	00:00:19,611 --> 00:00:21,400	0.000000	0.000000
9	বিমান বাহিনী প্রধান	বিমান বাহিনী প্রধান	00:00:21,840 --> 00:00:23,143	00:00:21,400 --> 00:00:22,626	0.000000	0.000000
10	দেশের সার্বভৌমত্ব রক্ষার পাশাপাশি	দেশের সার্বভমতরক্ষার পাশাপাশি	00:00:23,144 --> 00:00:25,385	00:00:22,626 --> 00:00:25,076	0.500000	0.121212
11	আর্থসামাজিক উন্নয়নে বিমান বাহিনীর ভূমিকার	আর্থসামাজিক উন্নয়নী বিমান বাহিনীর ভূমিকার	00:00:25,386 --> 00:00:28,042	00:00:25,076 --> 00:00:27,970	0.200000	0.023810
12	প্রশংসা করেন রাষ্ট্রপতি	প্রশংসা করেন রাষ্ট্র	00:00:28,043 --> 00:00:29,970	00:00:27,970 --> 00:00:29,249	0.333333	0.130435
13	সাক্ষাতকালে রাষ্ট্রপতির কার্যালয়ের সচিব	সাক্ষাতকালী রাষ্ট্রপতির কার্যালয়ের স	00:00:29,971 --> 00:00:32,371	00:00:29,249 --> 00:00:31,098	0.500000	0.100000
14	সম্পদ বড়ুয়া	পদ বড়ুয়া	00:00:32,372 --> 00:00:33,385	00:00:31,098 --> 00:00:31,992	0.500000	0.230769
15	সামরিক সচিব মেজর জেনারেল	সামরিক সচিব মেজর জেনারেল	00:00:33,386 --> 00:00:35,099	00:00:31,992 --> 00:00:33,862	0.000000	0.000000
16	এস এম সালাউদ্দিন ইসলাম, প্রেস সচিব মোঃ জয়নাল ...	এস এম সালাউদ্দিন ইসলাম প্রেস সচিব মো জয়নাল আব...	00:00:35,100 --> 00:00:39,313	00:00:33,862 --> 00:00:37,899	0.200000	0.035714
17	সচিব সংযুক্ত মোঃ ওয়াহিদুল ইসলাম খান	সচিব সংযুক্ত মো ওয়াহিদুল ইসলাম	00:00:39,314 --> 00:00:41,642	00:00:37,899 --> 00:00:40,209	0.333333	0.138889
18	এসময় উপস্থিত ছিলেন	উপস্থিত ছিলেন	00:00:41,643 --> 00:00:43,329	00:00:40,209 --> 00:00:41,413	0.333333	0.277778

Figure 6.7: Comparison of Generated Subtitle & Actual Subtitle.

According to our work plan, we applied a machine-learning model to generate subtitles for a Youtube video from a prominent news channel in Bangladesh. To evaluate the performance of our model, we also manually transcribed the video to obtain the actual subtitles. We then created a data frame to compare the machine-generated and manually-created subtitles. Figure 6.7 presents a snapshot of this DataFrame, where the actual_news column represents the manually transcribed subtitle lines and the news column represents the corresponding lines generated by our model for the given timecodes. By comparing these two columns, we were able to assess the accuracy of our machine-generated subtitles and identify any discrepancies with the manually created subtitles. From this, we also can see that our whole system is getting great WER and CER scores for this new Youtube video testing and the time codes are also very similar to each other achieving higher level of accuracy.

6.3 Overall Analysis & Improvement Opportunities

The results of the study are shown in the form of a pairplot in figure 6.8, which illustrates how the CER scores vary depending on the length of sentences that are included in the subtitles. The pairplot offers insightful comparisons of the degrees

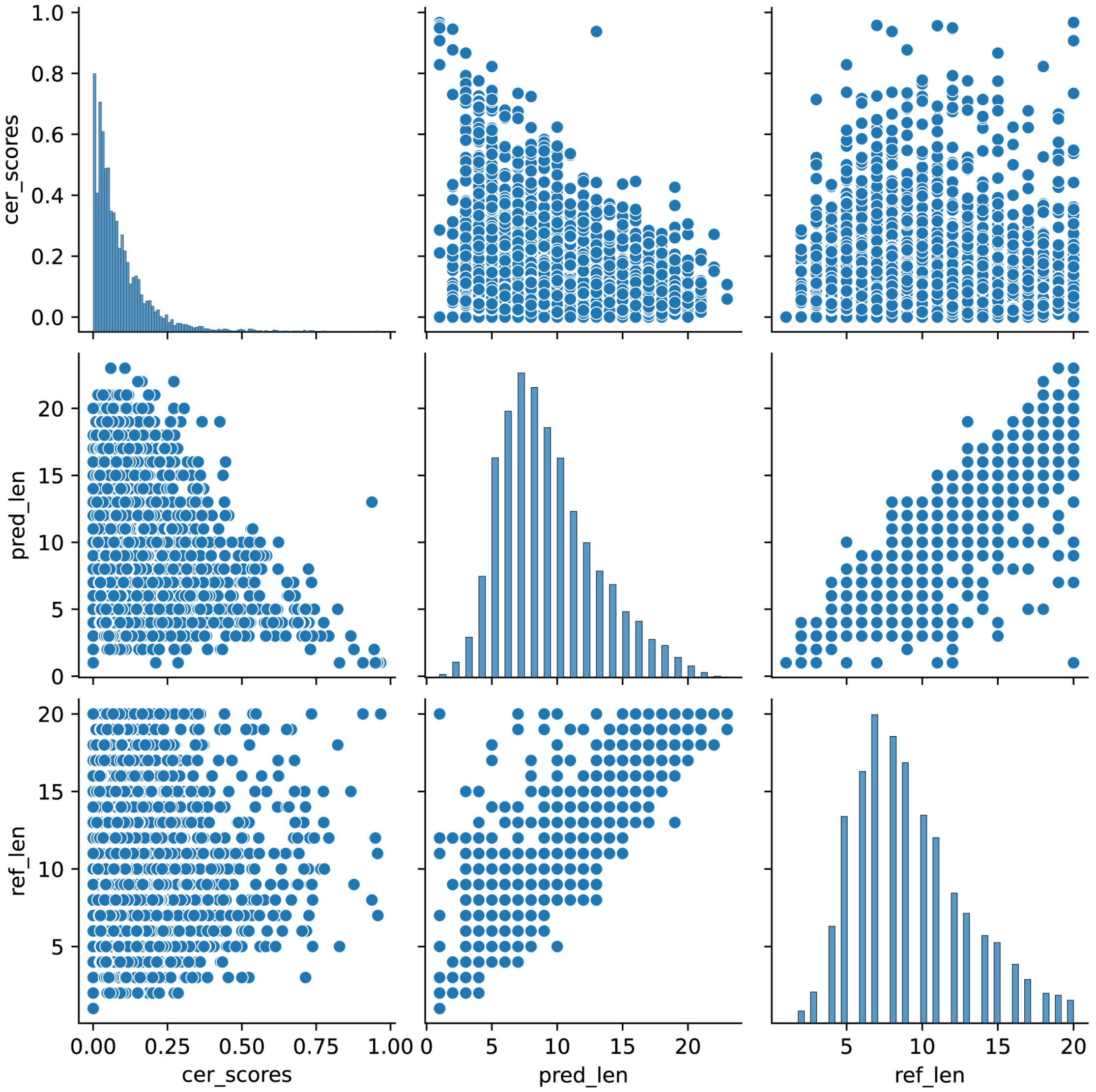


Figure 6.8: Pairplot showing the relationship between CER scores and the number of words in sentences for both machine-generated (Prediction) and human-made (Reference) subtitles. The plot illustrates how CER scores vary with the length of sentences for both types of subtitles, providing insights into the accuracy of machine-generated subtitles compared to human-made subtitles.

of accuracy achieved by machine-generated subtitles and those achieved by subtitles created by humans.

When we take a look at the pair plot in figure 6.8, we can see how the CER scores change depending on whether the amount of words in the sentences increases or decreases for both types of subtitles. Due to the nature of the pair plot, this is something that is open to us as an option. As a result of this research, we are able to have a better grasp of the effectiveness of machine-generated subtitles in contrast to the subtitles that were made by humans. It allows us to identify any trends or patterns in the CER scores based on the length of the sentence and evaluate the accuracy of the automatically produced subtitles over a range of sentence challenges. In addition to this, it enables us to determine the appropriate length for sentences.

This pairplot is an illustration of the link between CER scores and the length of sentences, which offers a more in-depth knowledge of the accuracy and usefulness of subtitles that were generated by a computer. It provides valuable insight into the advantages and disadvantages of the ASR system in accurately transcribing subtitles and reveals locations where adjustments may be made to raise the level of quality and precision of the machine-produced subtitles. In addition, it highlights places where modifications may be made to boost the speed at which the subtitles are created. In addition to this, it offers insightful information on the capabilities of the ASR system, as well as its shortcomings, in terms of effectively transcribing subtitles.

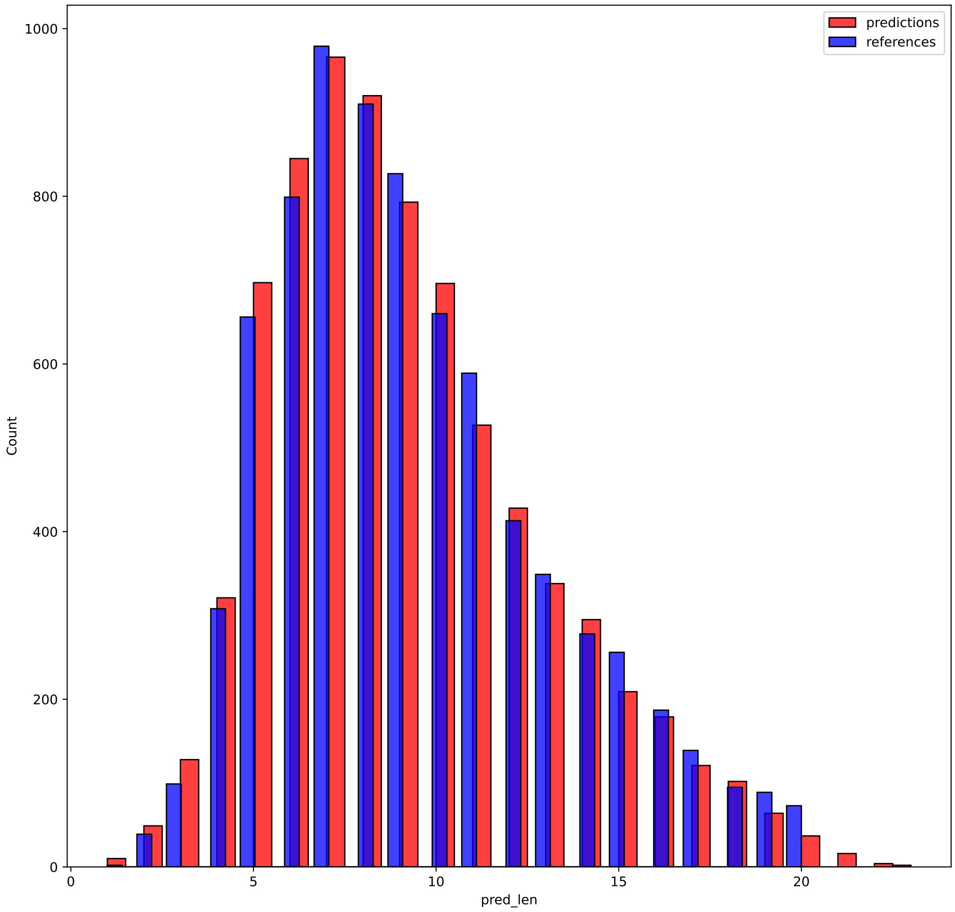


Figure 6.9: Distribution of the number of words in the sentences.

From the 6.9 graph, the Distribution of the number of words in the sentences is visualized which depicts that our predicted length has much higher accuracy in reference to the actual length.

The analysis of the ASR system, the timecode generation module, and the subtitle rendering module have offered significant insights into the performance of these components and areas in which they may be improved. After applying Bengali Unicode Normalization, the fine-tuned Wav2Vec2 model showed high performance on the Bengali audio dataset, with a CER score of 0.093 and a WER score of 0.283. These scores were derived from the model's analysis of the Bengali audio dataset. However, when the projected sentence length was different from the reference sentence length, different CER and WER scores were detected. Additionally, owing to the output of the ASR, inconsistencies were identified between the automated and human timecode annotations. These disparities were caused by the ASR. The timecode generating module demonstrated accurate alignment with minimal Absolute Timecode Error (ATE) and Relative Timecode Error (RTE) values; however, further comparison with human timecode annotations would offer a more thorough assessment. Opportunities for development include linguistically specific fine-tuning of the ASR model, investigation of post-processing approaches for improved transcription, and enhancement of timecode alignment. In addition, improving the resilience of the modules to changes and checking the accuracy of the manual timecode annotations would also help the general improvement of the system.

Chapter 7

Conclusion

In this thesis, we have successfully developed and evaluated a deep learning-based approach to automatically generate accurate and synchronized subtitles for Bengali audio content. Our approach involved gathering and preprocessing a large dataset of Bengali audio and corresponding subtitle data, designing and implementing a deep learning-based approach that incorporates techniques such as speech recognition, natural language processing, and sequence-to-sequence modeling, and evaluating the accuracy, synchronization, and readability of the generated subtitles.

Primarily, we followed a systematic work plan to automatically generate Bengali subtitles using deep learning techniques. We used Bengali Common Voice Dataset to train our automatic speech recognition model which is the foundation of building the automatic subtitle generator. We used several preprocessing methods, including noise reduction, silence removal, punctuation removal, and text cleaning, to guarantee the accuracy and consistency of the data we gathered.

Based on our findings, we can confidently assert that our methodology is capable of generating high-quality, perfectly timed Bengali subtitles for audio content. Several obstacles to creating automatic subtitles for Bengali audio were conquered with the help of cutting-edge deep-learning techniques and thorough data pretreatment. Our method is also very adaptable, making it suitable for use in a wide variety of languages and fields.

In conclusion, a system for the automatic generation of Bengali subtitles has been built through this study. Accurate subtitle transcription and synchronization were accomplished using deep learning techniques and the Bengali Common Voice Dataset. The suggested approach has the potential to enhance usability and accessibility in many fields, including the media, education, and entertainment industries. Improving the precision and timing of the generated subtitles may need further development of deep learning models and the incorporation of new data sources.

7.1 Challenges and Limitations

An admirable effort has been made to develop an automatic subtitle production system for Bengali audio, however, this endeavor has not been without its fair share of difficulties. The availability of data, especially training data that is of a high quality and is varied in terms of language, has continued to be a difficult obstacle to overcome. The many regional differences in accent, dialect, and pronunciation of Bengali made it challenging to develop a model that was correct over the language's whole territory. In addition, the complexity of the Bengali alphabet as well as the linguistic features of Bengali, such as ligatures and diacritics, have contributed to the difficulty of transcription. Variability in sentence lengths as well as the occasional unpredictability of the Character Error Rate (CER) and the Word Error Rate (WER) measures have also needed attention. Due to the possibility of inaccuracies in the manual timecode annotations that are used for comparison, achieving exact alignment between human and automated timecodes may be difficult. Despite the fact that post-processing methods have improved subtitle quality, they have not yet caught up to the standard of professional subtitles. Additional factors that contribute to the difficulty of this endeavor include the resource-intensive nature of fine-tuning models such as Wav2Vec2, the importance of usability concerns, and the need for user input.

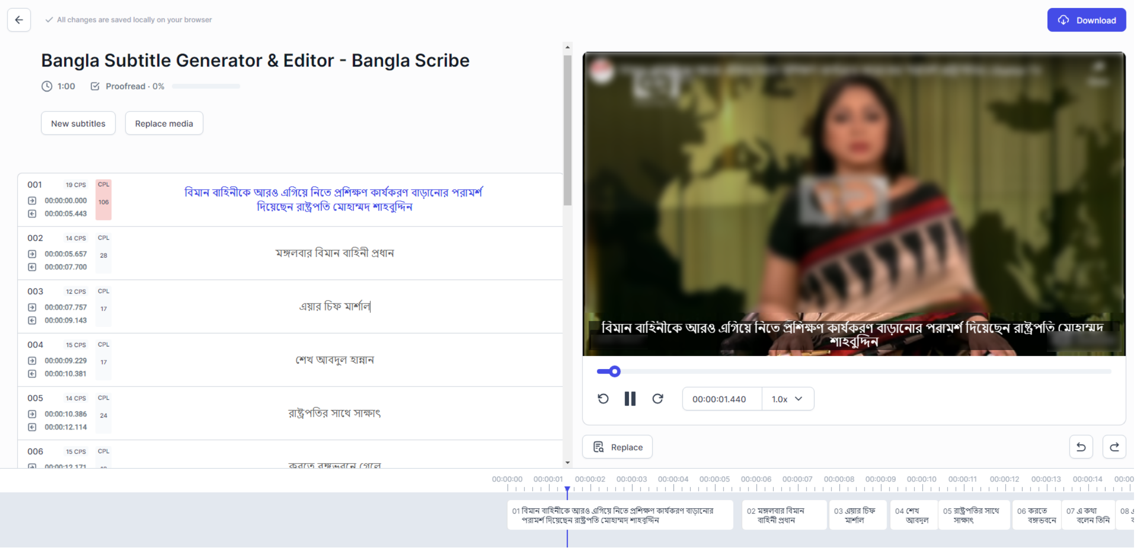
However, it is essential to keep in mind that these obstacles are only steps on the never-ending path towards the goal of generating multimedia material that is more accessible to Bengali speakers. Addressing these problems via data gathering, language adaption, and model optimization efforts will eventually lead to the improvement of subtitle production systems as future work progresses. The overall objective is to overcome these restrictions so that correct and synchronized subtitles may be provided to a wider audience, and so that this technology can be applied to other languages that need improved accessibility.

7.2 Future Work

The conclusion of this study is a significant step towards the construction of an automated subtitle-generating system for Bengali audio material, and it symbolizes the achievement of an essential milestone. However, there are a number of potential areas for future study and improvements that, if pursued, might accelerate the development of this technology and expand its scope of influence.

1. **Production-Level Deployment:** The immediate objective is to transform the prototype of the study into a system that is ready for production. In order to do this, the user interface has to be refined, the computational infrastructure needs to be optimized, and scalability needs to be ensured so that the system can support a rising user base. The creation of a user interface (UI) that is based on the web will facilitate easy access for customers, and the platform will be made available on a subscription basis in order to cater to the varied requirements of content developers, media organizations, and educational institutions. Figure 7.1 depicts the proposed UI of BanglaScribe which will be our SaaS (Software as a service) platform to auto-generate and edit Bengali subtitles for audio and videos.

2. **Language Expansion:** While Bengali has been the primary emphasis so far, future plans call for the automatic subtitle-generating system to be expanded so that it may serve new languages. We will be focusing on languages like indic-ocean languages as they are low-resourced and there is a big literature



Bangla Subtitle Generator & Editor - Bangla Scribe

1:00 Proofread · 0%

001	19 CPS 00:00:00.000 00:00:05.443	CPL 106	বিমান বাহিনীকে আরও এগিয়ে নিতে প্রশিক্ষণ কার্যকরণ বাড়ানোর পরামর্শ দিয়েছেন রাষ্ট্রপতি মোহাম্মদ শাহবুদ্দিন
002	14 CPS 00:00:05.657 00:00:07.700	CPL 28	মঙ্গলবার বিমান বাহিনী প্রধান
003	12 CPS 00:00:07.757 00:00:09.143	CPL 17	এয়ার চিফ মার্শাল
004	15 CPS 00:00:09.229 00:00:10.381	CPL 17	শেখ আবদুল হান্নান
005	14 CPS 00:00:10.386 00:00:12.114	CPL 24	রাষ্ট্রপতির সাথে সাক্ষাৎ
006	15 CPS 00:00:12.171	CPL	করতে বঙ্গভবনে গেলে

বিমান বাহিনীকে আরও এগিয়ে নিতে প্রশিক্ষণ কার্যকরণ বাড়ানোর পরামর্শ দিয়েছেন রাষ্ট্রপতি মোহাম্মদ শাহবুদ্দিন

00:00:01.440

01 বিমান বাহিনীকে আরও এগিয়ে নিতে প্রশিক্ষণ কার্যকরণ বাড়ানোর পরামর্শ দিয়েছেন রাষ্ট্রপতি মোহাম্মদ শাহবুদ্দিন
02 মঙ্গলবার বিমান বাহিনী প্রধান
03 এয়ার চিফ মার্শাল
04 শেখ আবদুল
05 রাষ্ট্রপতির সাথে সাক্ষাৎ
06 করতে বঙ্গভবনে
07 এ কথা বলেন তিনি

Figure 7.1: WebUI of Proposed BanglaScribe. A SaaS Platform to Auto-Generate & Edit Subtitles from Bengali Multimedia.

gap found. As part of this growth, we will be collecting data, refining our model, and adapting our language use to suit each of our target languages. The goal for the long run is to make the system a bilingual and multicultural instrument, which will improve accessibility and inclusion for a worldwide audience.

3. **Improved Post-Processing Techniques:** In spite of the fact that the existing system makes use of post-processing techniques to improve subtitle quality, continuing research might investigate even more sophisticated approaches. It is possible to use algorithms that are based on natural language processing (NLP) in order to improve the grammar, punctuation, and overall readability of subtitles, bringing them closer to the quality of subtitles that are created by professionals.

4. **Real-Time Subtitle Generation:** As a result of the progression of technology, the creation of capabilities for the production of automated subtitles in real-time is becoming an increasingly viable possibility. Accessibility and the overall quality of the user experience may be significantly improved by including a real-time mode in live events, streaming, or video conferencing.

5. **Integration with Existing Platforms:** Through collaborative efforts with pre-existing multimedia platforms and video-sharing services, it may be possible to realize more streamlined processes for the incorporation of the subtitle production system into the process of content creation. Plugins, application programming interfaces (APIs), and agreements with major streaming and video-sharing platforms are examples of this.

6. **Enhanced Evaluation Metrics:** A more nuanced comprehension of system performance may be attained via the ongoing development and improvement of assessment measures. The investigation of measures that take into account cultural and linguistic context, audience involvement, and cognitive load may lead to more complete evaluations of the quality of subtitles.

7. **User Feedback and Iterative Development:** It is vital, in order to make continuous improvements, to solicit input from users, and carry out usability tests. A method of development that is iterative and takes into consideration the preferences of users, the requirements of accessibility, and the many kinds of material that are always developing will guarantee that the system continues to be effective and relevant.

8. **AI and ASR Advancements:** It is essential to stay at a level of developments in both artificial intelligence (AI) and automated speech recognition (ASR) technology. Adopting models, methods, and pre-training datasets that are considered to be state-of-the-art may lead to considerable increases in the accuracy of ASR as well as the quality of subtitles.

In a nutshell, the work that has to be done in the future for this automated subtitle generation system entails both improvements in terms of technology and expansions in terms of strategy. The intention is to transition it from a research project into a tool that is easily accessible to a large audience and can be adapted so that it caters to a variety of audiences, encourages inclusion, and makes it easier to produce content in a variety of fields and languages. The long-term goal is to make interesting and approachable multimedia experiences available to users of all types, including those who create and consume information.

Bibliography

[1] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. DOI: 10.1109/5.18626.

[2] B. Guenebaut, *Automatic subtitle generation for sound in videos*, 2009.

[3] B. Das, S. Mandal, and P. Mitra, “Bengali speech corpus for continuous automatic speech recognition system,” in *2011 International Conference on Speech Database and Assessments (Oriental COCOSDA)*, 2011, pp. 51–55. DOI: 10.1109/ICSDA.2011.6085979.

[4] A. Graves, *Sequence transduction with recurrent neural networks*, 2012. arXiv: 1211.3711 `[cs.NE]`.

[5] A. Hannun, C. Case, J. Casper, *et al.*, *Deep speech: Scaling up end-to-end speech recognition*, 2014. arXiv: 1412.5567 `[cs.CL]`.

[6] K. Tarafder, N. Akhtar, M. Zaman, M. Rasel, M. Bhuiyan, and P. Datta, “Disabling hearing impairment in the bangladeshi population,” *The Journal of Laryngology & Otology*, vol. 129, no. 2, pp. 126–135, 2015.

[7] D. Amodei, S. Ananthanarayanan, R. Anubhai, *et al.*, “Deep speech 2: End-to-end speech recognition in english and mandarin,” in *International conference on machine learning*, PMLR, 2016, pp. 173–182.

[8] X. Che, S. Luo, H. Yang, and C. Meinel, “Automatic lecture subtitle generation and how it helps,” *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*, pp. 34–38, 2017.

[9] D. Sabane, P. A. Pawar, and R. S. Kadam, “Cuda compatible gpu as an efficient hardware accelerator for automatic subtitle generation,” 2017.

[10] Y. Zhang, W. Chan, and N. Jaitly, “Very deep convolutional networks for end-to-end speech recognition,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4845–4849. DOI: 10.1109/ICASSP.2017.7953077.

[11] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, “Advances in all-neural speech recognition,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Mar. 2017. DOI: 10.1109/icassp.2017.7953069.

[12] J. Cho, M. K. Baskar, R. Li, *et al.*, “Multilingual sequence-to-sequence speech recognition: Architecture, transfer learning, and language modeling,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 521–527. DOI: 10.1109/SLT.2018.8639655.

[13] L. Dong, S. Xu, and B. Xu, “Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition,” in *2018 IEEE International Con-*

ference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5884–5888. DOI: 10.1109/ICASSP.2018.8462506.

[14] Y. Zhou, C. Xiong, and R. Socher, "Improving end-to-end speech recognition with policy learning," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2018, pp. 5819–5823.

[15] V. B. Aswin, M. Javed, P. Parihar, *et al.*, "Nlp driven ensemble based automatic subtitle generation and semantic video summarization technique," *ArXiv*, vol. abs/1904.09740, 2019.

[16] S. Kriman, S. Beliaev, B. Ginsburg, *et al.*, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6124–6128, 2019.

[17] D. S. Park, W. Chan, Y. Zhang, *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *ArXiv*, vol. abs/1904.08779, 2019.

[18] M. Ravanelli, T. Parcollet, and Y. Bengio, "The pytorch-kaldi speech recognition toolkit," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6465–6469. DOI: 10.1109/ICASSP.2019.8683713.

[19] S. Schneider, A. Baevski, R. Collobert, and M. Auli, *Wav2vec: Unsupervised pre-training for speech recognition*, 2019. arXiv: 1904.05862 [cs.CL].

[20] X. Yue, G. Lee, E. Yılmaz, F. Deng, and H. Li, "End-to-end code-switching asr for low-resourced language pairs," in *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, IEEE, 2019, pp. 972–979.

[21] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and lstm encoder decoder models for asr," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2019, pp. 8–15.

[22] R. Ardila, M. Branson, K. Davis, *et al.*, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2020, pp. 4211–4215.

[23] A. Gulati, J. Qin, C.-C. Chiu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *ArXiv*, vol. abs/2005.08100, 2020.

[24] S. Mandal, S. Yadav, and A. Rai, *End-to-end bengali speech recognition*, 2020. arXiv: 2009.09615 [eess.AS].

[25] N. Moritz, T. Hori, and J. L. Roux, "Streaming automatic speech recognition with the transformer model," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6074–6078, 2020.

[26] N. P. Patel, H. Pansuria, J. Patel, B. M. Darji, and P. Sahatiya, "Automatic caption generator," *Journal of emerging technologies and innovative research*, 2020. [Online]. Available: https://www.jetir.org/view?paper=JETIR2004103.

[27] A. Ramani, A. P. Rao, V. Vidya, and V. R. B. Prasad, "Automatic subtitle generation for videos," *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 132–135, 2020.

[28] X. Sun, Q. Yang, S. Liu, and X. Yuan, "Improving low-resource speech recognition based on improved nn-hmm structures," *IEEE Access*, vol. 8, pp. 73 005–73 014, 2020. DOI: 10.1109/ACCESS.2020.2988365.

[29] Y. Zhao, C. Ni, C.-C. Leung, S. R. Joty, E. S. Chng, and B. Ma, “Speech transformer with speaker aware persistent memory.,” in *INTERSPEECH*, 2020, pp. 1261–1265.

[30] Y. Zhu, P. Haghani, A. Tripathi, *et al.*, “Multilingual speech recognition with self-attention structured parameterization.,” in *INTERSPEECH*, 2020, pp. 4741–4745.

[31] T. Javed, S. Doddapaneni, A. V. Raman, *et al.*, “Towards building asr systems for the next billion users,” in *AAAI Conference on Artificial Intelligence*, 2021.

[32] S. Kiran, U. Patil, P. S. Shankar, and P. Ghuli, “Subtitle generation and video scene indexing using recurrent neural networks,” *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 847–854, 2021.

[33] A. Kumar and R. K. Aggarwal, “An exploration of semi-supervised and language-adversarial transfer learning using hybrid acoustic model for hindi speech recognition,” *Journal of Reliable Intelligent Environments*, vol. 8, no. 2, pp. 117–132, Apr. 2021. DOI: 10.1007/s40860-021-00140-7. [Online]. Available: https://doi.org/10.1007/s40860-021-00140-7.

[34] A. Singh, S. Dharmesh, R. Jethwa, and M. Khasde, “Generation of transcript in multiple languages,” *SSRN Electronic Journal*, 2021. DOI: 10.2139/ssrn.3866510. [Online]. Available: https://doi.org/10.2139/ssrn.3866510.

[35] W. Zhou, M. Zeineldeen, Z. Zheng, R. Schlüter, and H. Ney, *Acoustic data-driven subword modeling for end-to-end speech recognition*, 2021. arXiv: 2104.09106 `[cs.CL]`.

[36] S. Alam, A. Sushmit, Z. R. Abdullah, *et al.*, “Bengali common voice speech dataset for automatic speech recognition,” *ArXiv*, vol. abs/2206.14053, 2022.

[37] S. Dutta, S. Jain, A. Maheshwari, S. Pal, G. Ramakrishnan, and P. Jyothi, *Error correction in asr using sequence-to-sequence models*, 2022. arXiv: 2202.01157 `[cs.CL]`.

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

[39] M. Rakib, M. I. Hossain, N. Mohammed, and F. Rahman, “Bangla-wave: Improving bangla automatic speech recognition utilizing n-gram language models,” *ArXiv*, vol. abs/2209.12650, 2022.

[40] H. S. Shahgir, K. S. Sayeed, and T. A. Zaman, “Applying wav2vec2 for speech recognition on bengali common voices dataset,” *ArXiv*, vol. abs/2209.06581, 2022.

[41] T. T. Showrav, “An automatic speech recognition system for bengali language based on wav2vec2 and transfer learning,” *ArXiv*, vol. abs/2209.08119, 2022.

[42] J. C. Vásquez-Correa and A. Álvarez Muniain, “Novel speech recognition systems applied to forensics within child exploitation: Wav2vec2.0 vs. whisper,” *Sensors*, vol. 23, no. 4, 2023, ISSN: 1424-8220. DOI: 10.3390/s23041843. [Online]. Available: https://www.mdpi.com/1424-8220/23/4/1843.

[43] *Animaker*, https://www.animaker.com/subtitle-generator, Accessed: 2023-04-30.

[44] *Nova a.i.* https://wearenova.ai/nova-tools/automatic-subtitles/, Accessed: 2023-04-30.