Shohojogi: An Automated Bengali Voice Chat System for the Banking Customer Services.

by

Kabir Abdur Rahman Arnab
18201090
Md Ashiqual Hossain
18201094
Istihad Nabi
18201071
Sadia Afrin Shoyti
19201137

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

Department of Computer Science and Engineering
Brac University
March 2023

Declaration

It is hereby declared that

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

Student's Full Name & Signature:

ARNAB

Kabir Abdur Rahman Arnab
ID: 18201090

Md Ashiqual Hossain
ID: 18201094

Inabi

Istihad Nabi
ID: 18201071

S.A.Shoyti

Sadia Afrin Shoyti
ID: 19201137

Approval

The thesis/project titled “ Shohojogi: An Automated Bengali Voice Chat System for the Banking Customer Services.” submitted by

1. Kabir Abdur Rahman Arnab (18201090)
2. Md Ashiqual Hossain (18201094)
3. Istihad Nabi (18201071)
4. Sadia Afrin Shoyti (19201137)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on March 24, 2023.

Examining Committee:

Supervisor:
(Member)

Iqbal

Dr Muhammad Iqbal Hossain

Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Robiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
Brac University

Ethics Statement

We now state that this thesis is based on our research findings. All other materials used have been appropriately documented. This work has never been submitted, in whole or in part, to any other university or institution for the award of a degree

Abstract

As the name refers, this research aims to develop an automated voice chat system in Bengali language for the banking system. The system enables clients to learn about detailed information such as account opening queries, loan requirements, fund transfer limits, etc. through a natural language-based interactive voice response system. The system will use speech recognition technology to understand the customer's voice commands and respond accordingly in Bangla.It uses the sentence summarization technique and also uses text-to-speech technology to provide spoken responses to the customers. Customer call centers have grown in popularity as a result of pandemics and are now widely employed in a variety of industries, including e-commerce, hospitals, banks, credit card assistance, and government agencies, among others. Also, it is more difficult to satisfy all of the call center clients due to humans' constraints on being available 24 hours a day and the variation in waiting times. In order to effectively manage consumers by giving a domain-based answer in the customer's local tongue, customer service must be automated, especially in emerging nations like Bangladesh where the number of contact support centers is growing. While most people speak in Bangla, there hasn't been much progress made in automating customer service in the local tongue. By recognizing user voices, defining users' issues in the standardized Bengali language, and gathering users' replies into the database to provide feedback in accordance with the queries, our established approach, "Shohojogi", can reply to that customer's requirement. The ability to listen and speak with the user is implemented using speech recognition by the wav2vec2 model while for text summarization we used the seq2seq model and the ability to understand and find the related information is implemented by using the doc2vec model. Finally, we use gTTS for text-to-speech conversion.

Keywords: Voice Chat System, Bangla Speech Recognition, Text Similarity, Text to Speech (TTS).

Dedication

We wish to devote all of our educational efforts and sacrifices to our wonderful parents, without whom we are useless. We likewise dedicate our thesis report to Dr. Muhammad Iqbal Hossain, sir, who filled in as our supervisor and who mentored us, taught us in the advancement of our abilities and characters as proficient experts.

Acknowledgement

BRAC University provided support to facilitate the completion of this study.
We appreciate the Almighty Allah for protecting us during the COVID-19 outbreak and allowing us to complete our research.
We would like to take this opportunity to thank our supervisor, Dr. Muhammad Iqbal Hossain, and another special mention, Dr. Farig Yousuf Sadeque, for everything that they have done to help us and for letting us work for them.
Finally, many thanks to our parents for their kind prayers and support. Also, we lost our beloved Hossain Arif, sir, along the way, but he will live on forever in our hearts.

Table of Contents

List of Figures

List of Tables

Nomenclature

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

ϵ	Epsilon
ϵ	Epsilon
υ	Upsilon
υ	Upsilon
API	JavaScript Object Notation
$BIQAS$	Bengali Intelligence Question Answering System
$BNLP$	Bengali Natural Language Processing
CNN	Convolutional Neural Networks
GS	Google Speech
NLP	Natural Language Processing
RNN	Recurrent Neural Networks
SVD	Singular Value Decomposition
$TF-IDF$	Term Frequency-Inverse Document Frequency
AWS	Amazon Web Services
DM	Distributed Memory
HMM	Hidden Markov Model
$LSTM$	Long Short-Term Memory
$LVCSR$	Large Vocabulary Continuous Speech Recognition
MOS	Mean Opinion Score
POS	Parts Of Speech

Chapter 1

Introduction

The term ”customer service” in the banking sector refers to the many channels through which a financial institution communicates with its clients in order to respond to their needs, inquiries, and complaints. Today’s customer service extends much beyond the standard phone support worker. It can be accessed through email, the web, text messages, and social media. Many businesses also offer self-service assistance, allowing customers to obtain answers at any time of day or night. Customer service is more than just answering questions; it’s an essential aspect of the promise your company makes to its customers. The objective of customer service within the banking sector is to establish favorable customer experiences and cultivate enduring customer relationships. The efficacy of customer service is a crucial element in the banking industry, as it facilitates the establishment of confidence and allegiance among customers, thereby potentially resulting in heightened contentment and preservation. For example, BRAC Bank uses the information center phone number as its customer service. This information center will listen to the user’s questions and tell the information that users need. They also have a section where the customers can file their complaints and within 48 hours [10], a service provider will help them with solutions.
An automated chat system is a piece of software that can carry on a discussion with a human (through text or speech) about specific subjects and in a particular language. The advancement of technology drives customer service to improve, and the goal of this study is to further that development. This study hopes to create a voice intelligence conversation bot that can provide services similar to those offered to customer service. According to a report, voice automated systems can save businesses $20 million globally, and this sum is expected to reach $8 billion by end of 2023. A voice system allows users to communicate with a device or service by just speaking to it. It also can understand a said inquiry or request and structure an appropriate audio answer using artificial intelligence and Deep learning [36]. Voice automated system released a list honoring current popular voice brand innovators because of the increased acceptance of voice-first devices like smart speakers in recent years, an increasing number of businesses have realized that voice is a great tool for interacting and engaging with their audience such as Bank of America, Mercedes, BBC, Nike, Sephora, etc. [23]. Also, PayPal pioneered using a voice-activated system in 2016 for their payment and customer care system [34]. Human customer support engagements will be decreased by 30% in the next two years, according to McKinsey. This ultimately indicates that in the future, chatbots and

speech bots might control up to 70% of these encounters [45]. Voice systems reduce waiting times and improve the quality of customer service. Voice bots have a distinct advantage in terms of immediacy, owing to the fact that they save clients from having to put in a question or request. If you only look at the data, humans can type 53.5 words per minute on a normal keyword, but they can speak around 161 words per minute, which is more than triple the number of words they can type [45]. As a result, speech recognition system assistants deliver on their promise of providing real-time customer assistance conversations with little downtime. Many consumers appear to be concerned about the quality of customer service when it comes to automation. However, why should it be considered harmful if a VoiceBot is used to answer inquiries that can be answered automatically and hence swiftly and at any time? When given the option of waiting longer to speak to a person or receiving an answer directly from a Voice assistant, many individuals will choose artificial intelligence. Because, in the end, the criterion for a satisfying customer experience is whether customer problems were quickly and competently recorded and resolved, not whether a human being talked to them. Although they are not identical, it is helpful to think of a voice bot in the same way that a chatbot is supposed to. Both are built on the same underlying technology and are designed to understand a customer's issue and find and offer the best possible solution. Text-based chatbots can perform various tasks ranging from booking a hotel reservation to paying a bill, thus making life easier for consumers [45]. The use of text as an input modality is a constraint because it necessitates using a keyboard at all times when a user (the blind) wants to speak with the bot[15]. Voice bots are more natural and efficient because they do not rely solely on humans' most basic mode of communication (speech/voice) [34]. Nonetheless, it will offer a better-integrated customer experience with quick voice feedback, allowing users to multitask more efficiently. The use of voice services provides a level of trust from the user to the service being used [19], boosting usability and ensuring that the service is available to users at all times [3]. Users can complete tasks considerably faster using voice chats than typing. Younger generations have largely fueled the popularity and adoption of voice bots across a variety of use cases, including customer service. In fact, 51% of consumers aged 14 to 17 had used a speech or voice recognition interface or gadget. Consumers aged 18-34 accounts for 38%, customers aged 35-55 account for 27%, and consumers aged 55 and up an account for only 15% [32]. This speech intelligence conversation bot combines speech recognition with the capacity to listen to human input and convert it to text, as well as turn the output text back into voice [18] [1], and make it intelligent by using a chatbot system that can learn to understand the user's question and provide the appropriate information [5]. If something is too complex or needs a human touch, it's passed to a live agent. The agent receives all context and details to complete the call.

The target of our project is to design a Bangla automated voice chat system named ”Shohojogii” that can provide customers with a convenient and efficient way to perform banking tasks in Bangla, without the need for physical visits to a bank branch or access to the internet banking. It will also help banks to reduce their operational costs and improve customer satisfaction by providing a more personalized and responsive service. It is designed to handle basic banking activities, providing users with relevant info and bank details in a seamless and efficient manner. Customers can get information on various types of deposit accounts, different types of loans and their requirements, online and mobile banking services, credit cards, and other pertinent details. The idea is to mimic what happens in the bank and build a personal connection between the customer and the system. After all, the corporation believes that all of its digital projects should be able to communicate in the same language and offer the same chances.

1.1 Research Problem

There is a growing need for a Bangla-language voice chatbot for banking in Bangladesh. While many banks offer online banking services, most of them only provide support in English, which is not the native language of many Bangladeshis. As a result, customers who prefer to communicate in Bangla may face difficulties accessing banking services and obtaining information about their accounts. Also, many people in Bangladesh may not be able to read or write, which makes it even harder for them to use banking services. Therefore, there is a need for a voice chatbot that can assist customers in their native language, Bangla, and provide them with an easy and efficient way to access banking services, make transactions, and get account information.

The banking industry is one of the most competitive and constantly evolving sectors in the world. With the rise of digital technology, banks have been forced to adapt and innovate in order to stay ahead of the game. One area where this is particularly important is in customer service. In recent years, many banks have turned to voice command bots as a way to improve customer service.

Every day, a corporation can face a variety of customer service issues. Some are simple to deal with, while others are more difficult. The first issue that clients face is the length of time it takes to resolve their issues. Over 80% of consumers say they expect an immediate response to customer service inquiries [6]. Recent studies have shown that the average customer support request is a whopping 12 hours. While the simple solution is to "hire more agents," this is not an option for small firms and entrepreneurs in particular. Customers frequently become frustrated as a result of wasting time repeating information, and when several agents are involved, this means additional time spent waiting for incoming calls or chats to be answered. One of the reasons behind the delay is that they are being switched between departments. Therefore, VoiceBot is the easiest way to provide instant answers to customers.

Another problem worth mentioning is customers find it extremely aggravating to be on the phone with a support staff who they believe isn't properly qualified to assist them or who lacks even basic expertise about a product or service [6]. When customers contact a company, they want to acknowledge everything there is to know about the products and services. A lack of skilled or inexperienced personnel might completely derail a project. Such agents obstruct the delivery of a positive client experience. Therefore, in this kind of situation voice bot is a perfect solution. Bots can provide any kind of solution within a fraction of time with relevant knowledge about the situation. In short, they are customized to act as a professional customer service representative.

Another problem is a company requires extra people to interact with customers to improve customer service efficiency. To handle the increasing number of customer demands, the company will need to hire more personnel [6]. Employees are not always properly trained to provide appropriate customer care. More people need more training and salary which will cost a big amount of money for any company. In addition, the majority of consumers can have the same problem and require the same type of solution. So if they use a speech recognition conversation bot, then all this money for hiring and training people will be saved or companies can invest this for developing other sectors of their business or even can expand it more and bots can also provide solutions to those limited problems which are simple enough or the most common ones.

Sometimes Customer care representatives offer favors that cannot be fulfilled [2]. They make a promise to a customer and then break it, or they can't do anything because of the policy. Some customer service representatives who make promises may not follow through because they are interested in their work. They might make a promise to get rid of a customer [2]. Bots do not make fake promises. We can design a bot who is empathized with the customers however they always provide genuine information and company policies.

Salespeople are humans, too, and they require relaxation. Off-hours and on holidays, the customer care representatives are unavailable. In most circumstances, users cannot obtain services at midnight or on public holidays, which might be inconvenient for some customers. So the voice chat can be handy in this type of circumstance because it is an automation service that provides service 24 hours a day.

Making sure that your employees are capable of providing excellent customer service to physically challenging and ill clients have a significant impact. It is critical that businesses give the best possible service to this segment of the population. Being physically present or contracting with customer service in any other way will be difficult for these individuals. As a result, voice chat will make people's lives easier or simpler. They may receive the information they need by simply utilizing the voice command.Even so, dealing with these kinds of issues will get more difficult in the following days.

Furthermore, corporations are heavily reliant on their customer service, but with the help of modern technology, we are ready to overcome most of these problems.As we all know, speech recognition voice chatbots can help us reduce the interaction cost-the amount of the physical and mental work of these services. As a result, implementing a voice chat system in the customer service sector can result in long-term reform as well as alleviate the dependency on customer service representatives.

1.2 Research Aim and Objective

This research intends to enhance banking customer service by adopting voice command technology, which allows customers to conveniently obtain information. One of the main reasons we use voice commands is because it allows us to multitask more successfully by allowing us to use our gadgets without having to write or glance at the screen.The following is a list of the research's objectives: The aim and objectives of the research for the Bangla voice chat system can be defined as follows:

- To develop a natural language processing-based Bangla voice chat system that can understand and respond to user requests in the Bengali language.
- To evaluate the performance and usability of the Bangla voice assistance using a user-centered approach, with a focus on accuracy, response time, user satisfaction, and task completion rate.

- To contribute to the field of natural language processing and voice technology by developing and evaluating a Bangla voice chat system, and by highlighting the importance of developing technology that caters to the unique needs of diverse language communities.

- To minimize the cost of a bank by replacing most service representatives with a speech recognition voice chat system.

- To assist the customers 24/7 as a voice bot will always be accessible.

- To make physically challenged people's life simpler as they just need to use voice commands to get the information they require.

Chapter 2

Related Work

In recent years, several studies have explored the use of natural language processing techniques in automated customer service systems. One such technique is automatic speech recognition, which involves converting spoken language into text format.

To achieve high accuracy in speech recognition, researchers have explored various models and algorithms. One such model is XLSR-Wav2Vec2, which is a pre-trained speech recognition model that has achieved state-of-the-art performance on various benchmark datasets [30]. XLSR-Wav2Vec2 was trained on a diverse set of audio data from languages around the world, including Babel, Multilingual LibriSpeech (MLS), and Common Voice. By leveraging this pre-trained model, the Shohojogi system is able to accurately transcribe spoken Bangla language.

In addition to speech recognition, researchers have also explored text-to-speech synthesis techniques to improve the overall user experience. One such technique is Google Text-to-Speech (gTTS), which is a free and open-source software library for text-to-speech conversion [17]. By using gTTS, the Shohojogi system is able to convert text responses into natural-sounding Bangla language, enhancing the overall user experience.

Researchers have used several strategies, including Doc2Vec and Word2Vec, to measure sentence similarity. An addition to the Word2Vec model called Doc2Vec may identify the semantic content of a sentence or a paragraph [13]. The Word2Vec neural network-based model can represent words in a high-dimensional space and capture their meaning and context. [11].

In addition to these models, researchers have explored various similarity metrics to measure sentence similarity. One such metric is cosine similarity, which measures the cosine of the angle between two vectors in high-dimensional space. Another metric is Jaccard similarity, which measures the similarity between two sets of words [35].

By leveraging state-of-the-art models such as XLSR-Wav2Vec2 and gTTS, and techniques such as Doc2Vec and Word2Vec with cosine similarity and Jaccard similarity metrics, the Shohojogi system is able to provide accurate and relevant responses to customer queries, enhancing the overall user experience.

Overall, the Shohojogi system contributes to the ongoing research on automated customer service systems by demonstrating the effectiveness of natural language processing and speech recognition techniques. By leveraging state-of-the-art models and techniques, the system is able to provide accurate and relevant responses to customer queries, improving the overall user experience.

Chapter 3

Background Studies

3.1 Literature Review

In the developed architecture named “AIMS TALK”, the recognition of speakers from an audio source is accomplished with the help of the MFCC feature extraction method [9].All of these MFCC features are low-frequency aspects. Additionally, Gaussian Mixture Model is used to build the feature-matching technique that is embedded within the maximization process. (GMM). Customers’ voice inquiries can be translated into text by an ASR model. Using a technique called Bangla Sentence Summarizing, the system is able to condense lengthy sentences down to their essential parts. Using an attention model, the Seq2Seq Bangla news summarization method is applied to the customer feedback, which is then summarized. The training loss can be reduced by a factor of 0.001 using the Seq2seq model, which corresponds to a good outcome in the experimental test. By utilizing Sentence Transformer to encode all of the sentences in the database and saving the encoded score in an array, encoding takes time for every user each time. The sentences are fed into a pooling layer, and then the BERT model is used to generate a score prediction vector. The ”gTTS” Python package is used to convert the text to speech[41].
The system, named “Adheetee”, accepts both text and voice commands as input. But a user can also issue commands by inputting them in. Here, a Speech to Text (STT) system is used to convert spoken commands into written ones. This is done by utilizing Google’s STT API[25]. After that, pull relevant terms from the given text. Because the orders are delivered in natural language, it must extract the keywords from the command. This means that two users can provide the same command in completely different ways, either verbally or in writing. It needs to analyze the keywords to establish the nature of the command, such as whether it is a simple or essential one. In response to a user’s request, the system checks its internal state to see if it has the necessary data, or it makes an API call to retrieve the data from elsewhere. Data is gathered from the system’s knowledge base and/or stored in the user’s profile for further use. The Bangla text is translated by a machine translation API into English text, and then the translated English text is used as a parameter in the external API. Machine translation is handled using the Google Translate API [29]. A JSON and text file collection with various types of data from many domains and functionalities is included in the knowledge base, which also has a SQLite database. The goal of this decentralized database is to facilitate both access and change. An index JSON file stores commands, keywords, and flags for

whether the command is basic or core, whether the system can handle the request or whether an external API call is required, whether the keyword(s) associated with the command need to be translated into English and other relevant data [25].
The original goal of the Bangla automated voice chat system named “Alapi”, was to have it converse in Bangla on a limited set of topics and answer limited sets of queries. Python is the primary development language for the system. The user’s voice is first recorded by the device’s microphone and then processed by the system. It takes the audio input and translates it into text using Google’s speech recognition API. A written response is produced when the text data is analyzed by an AI model. Google’s Text-to-Speech API is then used to transform the text response into an audio file. (gTTS). The system listens to the user’s voice through the device’s microphone. In order to accomplish this, PyAudio is employed. In order to send the voice data for speech recognition and speech-to-text conversion, PyAudio processes it into a data file. The system employs the Google Speech (GS) API to identify the language of the input voice data and to transform the voice data into a text file. The training information is saved in a Json file with a predetermined structure. The queries are tokenized into individual words using Natural Language Processing (NLP) and then saved in a python list. Three fully connected (FC) layers, each with multiple neurons and weighted interconnections, have been added to the model for the purpose of model training. First, the system tokenizes the text form of input data by extracting individual words in order to provide an appropriate answer for the provided input. Then, it’s turned into an array and compared to the provided tags. Once the tag with the highest matching percentage has been determined, output consisting of a single randomly selected response under that tag is returned and then converted to audio. The success or failure of question and answer set prediction is tracked in a NON-SQL database developed with the help of the MongoDB database management system [39].
The Bengali Intelligence Question Answering System (BIQAS), a system that uses Bengali natural language processing to answer questions based on arithmetic and statistics.(BNLP). The process can be broken down into three separate stages: gathering relevant documents, processing raw data, and establishing a foundation for answering user queries. For use in preliminary processing, relevant corpora are included. Cosine similarity, Jaccard similarity, and the Naive Bayes method are all recommended to help find the connection between the queries and their results. Vectors are the focus of the Cosine Similarity metric. In this scenario, the TF-IDF model is used to communicate both the papers and the questions to the vectors. SVD techniques were employed to reduce execution time and space complexity [27].

3.2 Speech Recognition

Hidden Markov Model (HMM) is a statistical model that is commonly used in speech recognition, handwriting recognition, and other pattern recognition tasks. In speech recognition, the HMM is used to model the relationship between an acoustic signal and the sequence of phonemes that make up a word or sentence. Moreover, the HMM is used to determine the most likely sequence of hidden states that correspond to a given speech signal[8]. However, Bangla has a complex script with a large number of characters, which makes it difficult to transcribe speech signals into text accurately.

Gaussian Mixture Model (GMM) is a statistical model that is commonly used in pattern recognition, including speech recognition. In speech recognition, GMMs are used to model the relationship between the acoustic features of speech signals and their corresponding phonemes. The GMM assumes that the acoustic features of speech signals are generated by a mixture of several Gaussian distributions, each representing a different phoneme [33].In the case of Bangla speech recognition, the acoustic features of speech signals can have a very high dimensionality, which makes it difficult to accurately model the relationship between these features and the corresponding phonemes using a GMM.

Kaldi is an open-source speech recognition toolkit widely used in research and industry. Kaldi is designed to handle a variety of speech recognition tasks, including keyword spotting, speaker diarization, and large vocabulary continuous speech recognition (LVCSR). Kaldi uses advanced techniques from machine learning and signal processing to achieve state-of-the-art performance in speech recognition [42].However, Kaldi relies heavily on the availability of large datasets and language resources, such as lexicons and language models, which may not be readily available for Bangla.

DeepSpeech2 is a neural network-based speech recognition system that is designed to handle large vocabulary continuous speech recognition (LVCSR) tasks[24].In terms of accuracy for speech recognition in the Bangla language, DeepSpeech2 may not be the perfect fit for training Bangla language due to several challenges. Furthermore, there is a lack of large, high-quality transcribed Bangla speech datasets, which can make it difficult to train DeepSpeech2 models effectively.

CMUSphinx is an open-source speech recognition toolkit that provides a suite of tools and libraries for developing speech recognition systems. It is designed to be highly configurable and can be used for various types of speech recognition tasks, including isolated word recognition, keyword spotting, and large vocabulary continuous speech recognition (LVCSR) [21]. However, implementing CMUSphinx can be complex due to the need for specialized knowledge of signal processing, language modeling, and speech recognition algorithms.

Wav2Vec2 is modern voice recognition technology uses self-supervised learning approaches to increase the precision and effectiveness of speech recognition. [44]. Since Wav2Vec2 can be configured to work in a variety of languages and acoustic settings and is incredibly accurate and effective, it is regarded as one of the greatest speech recognition systems currently on the market. It also requires little specialist expertise of signal processing or language modeling and is quite simple to implement.

XLS-R wav2vec2 is a state-of-the-art speech recognition model developed by Facebook AI Research. It is an extension of the original wav2vec2 model, which uses a self-supervised learning approach to learn speech representations directly from raw audio data.The XLS-R wav2vec2 model is considered the best to implement and easy to use due to its open-source codebase and the availability of pre-trained models in various languages [43]. Additionally, it has achieved state-of-the-art results on several benchmark datasets for speech recognition in low-resource languages, including Bangla.This demonstrates the effectiveness of the model in handling the challenges posed by Bangla speech recognition, such as a large number of phonemes and the lack of high-quality training data.In terms of accuracy for speech recognition in the Bangla language, Wav2Vec2, and XLSR Wav2Vec2 have shown promising results.

3.3 Text Summarization

Extractive summarization is a popular text summation approach that involves selecting the most important lines or phrases from the source text and combining them to form a summary. Using this method, you will select the crucial phrases or sentences from the original material and combine them to produce a summary. Methods like TF-IDF, TextRank, or Latent Semantic Analysis can be used to achieve this. (LSA) [16].Each sentence is given a score based on its importance or relevance to the text as a whole. A number of variables, such as phrase frequency, sentence length, placement within the sentence, or identified entities, may affect the score. After each sentence has been scored, the sentences that received the highest marks are those that will be featured in the summary. Depending on how long it is, the amount of sentences that should be in the summary can either be predetermined or determined as it goes. A summary is then produced by combining the selected sentences. The important ideas or specifics from the original material should be communicated in a comprehensible, logical summary.

By comprehending the context and meaning of the text and then creating new sentences that encapsulate the key points of the text, the approach of abstractive summarization creates a summary. This method creates new sentences that more effectively and concisely communicate the text's essential concepts, going beyond simply taking the most significant passages from the original text. This technique entails creating a summary that could include fresh words and phrases that aren't found in the original material. Techniques like deep learning-based models like seq2seq and transformer-based models like BERT can be used for this. [37]. Abstractive summarization in Bangla language has several advantages over extractive summarization. First, it can capture the essence of the text more accurately and concisely. Second, it can handle complex sentence structures and idiomatic expressions. Third, it can generate summaries that are more readable and coherent than extractive summaries.

The Natural Language Processing team created XL-Sum, an open-source extractive summarization toolkit for the Bangla language (NLP). XL-main Sum's goal is to create a summary of a given input text by picking the key phrases from the original text. Sentence scoring and sentence selection make up the two steps of the extractive summarization method used by the toolkit [40].Based on the XLNet language model, the xl-sum package is a Python library for text summarization. Although the pre-trained model was developed using English text, the library can also be modified to sum up information in Bangla. In conclusion, even though xl-sum is already trained on English text, it can be customised for summarization on Bangla by fine-tuning the model on Bangla text.

A cross-lingual summarization framework is implemented in the Python text summarization library known as CrossSum. Using data from summarization models developed on texts in other languages, this framework enables the summarization of texts in one language.Pre-trained summarization models for several languages, including English, French, and Spanish, are available in the CrossSum library. These models are built on the transformer-based BERT architecture, which excels at tasks requiring natural language processing [38].A number of tools are also included in the package for pre- and post-processing the text data as well as for assessing the calibre of the summaries produced by the models. Users are given the option to select the summarization method that best suits their needs from the library's capabilities for both extractive and abstractive summarization. Overall, based on cutting-edge summarization models, CrossSum is a strong and adaptable framework for cross-lingual text summarization. The summarization process can be tailored using a variety of tools provided by the library, which can be used with a wide range of languages and data sources.

3.4 Sentence Similarity

Vector-based sentence similarity is a popular technique in text mining and natural language processing. Word embeddings are a popular technique for representing sentences as vectors. They map each word in a sentence to a high-dimensional vector space based on the context in which it appears. To create a vector representation of the entire sentence, these vectors can then be averaged.

Another vector-based method is called latent semantic analysis (LSA), which factors a word frequency matrix of a set of sentences using singular value decomposition (SVD) to produce a lower-dimensional vector space representation. The cosine similarity between two sentences' respective vector representations in this lower-dimensional space can then be calculated to determine how similar they are to one another.

The method of calculating the cosine similarity between two sentences is frequently employed. Using word embeddings like Word2Vec or doc2Vec, the sentences are first transformed into vector representations in this method. These embeddings convert each word in a sentence into a high-dimensional vector, which are then averaged to produce a single vector representation for the entire sentence [7].The cosine of the angle between two sentences' vector representations is computed to determine how similar two sentences are to one another. The two vectors' dot product, divided by the product of their magnitudes, can be used to achieve this. The cosine similarity between the two sentences is the value that results.

Jaccard similarity is yet another approach that is frequently used to assess how similar two sentences are. This method treats the sentences as collections of words, and it determines how similar they are based on the size of the intersection and union of these collections [28].The words in each sentence are first tokenized and added to separate sets before calculating the Jaccard similarity between the two sentences. The size of the intersection of these sets is then divided by the size of their union to determine the Jaccard similarity.The Jaccard similarity metric is a useful tool for determining sentence similarity.

3.5 Text to speech

eSpeak is an open-source, multi-lingual text-to-speech (TTS) synthesis engine that can be used for speech synthesis on various platforms. It supports many languages, including Bangla, and provides users with a set of customizable parameters for speech output. Implementing eSpeak for Bangla language TTS can be challenging due to the lack of standardized pronunciation rules and language resources. However, eSpeak can still be used for Bangla TTS synthesis with moderate accuracy and quality [20].To use eSpeak for Bangla TTS, a Bangla language voice file needs to

be installed, which provides the engine with the phonetic and acoustic information required for speech synthesis. The voice file for Bangla language is available on the eSpeak website, and it can be easily installed.
Festival is a free and open-source text-to-speech (TTS) synthesis system that is designed to support multiple languages, including Bangla [26].It is a highly customizable system that allows users to control various aspects of speech synthesis such as voice, intonation, and speed. To implement Festival for Bangla language TTS synthesis, a Bangla language voice file needs to be installed, which provides the system with the necessary phonetic and acoustic information.The Festival system converts the text into phonetic units, and then uses the Bangla language voice file to synthesize speech output. In a study by [26], the performance of Festival for Bangla language TTS synthesis was evaluated, and the results showed that the system was able to produce highly intelligible speech output with a mean opinion score (MOS) of 3.64 out of 5.
Google Cloud Text-to-Speech is a cloud-based text-to-speech synthesis service that provides high-quality, natural-sounding speech in multiple languages, including Bangla. It is a powerful and flexible system that allows users to customize various aspects of speech synthesis, such as voice, intonation, and speed.The Google Cloud Text-to-Speech service produces high-quality, natural-sounding voices in a variety of languages, including Bangla. It is a cloud-based text-to-speech synthesis service. It's a very effective and adaptable technology that provides a wide range of options for altering the tone, intonation, and rate of speech data. The gTTS (Google Text-to-Speech) Python library and command-line utility leverage the Google Text-to-Speech API to transcribe the written text into spoken language. Bangla is one of the languages it supports. The user must supply the Bangla language text they wish to synthesize into speech in order to use gTTS for Bangla language TTS synthesis. The system then reads the text aloud using the Google Text-to-Speech API. Language, speed, and pitch are only a few of the voice synthesis parameters that can be adjusted by the user. There has not been a lot of research into how well gTTS performs for Bangla language TTS synthesis. However, reasonable accuracy is anticipated given that it is built on top of the same core Google Text-to-Speech API that has been shown to generate very natural-sounding speech output for the Bengali language. Since gTTS is a Python library, it can be readily incorporated into Python-based programmes and scripts, which is one of its advantages. There is zero preparation or cost associated with using it. In summary, gTTS is a straightforward and user-friendly application for Bangla language TTS synthesis, leveraging the power of the Google Text-to-Speech API to generate high-quality synthetic speech.
Amazon Polly is a cloud-based text-to-speech synthesis service offered by Amazon Web Services (AWS) that provides high-quality, natural-sounding speech in multiple languages, including Bangla. It is a highly flexible and customizable system that allows users to control various aspects of speech synthesis such as voice, intonation, and speed. To implement Amazon Polly for Bangla language TTS synthesis, the user needs to provide the Bangla language text that they want to synthesize into speech. The system then uses advanced neural network models to convert the text into speech output that sounds natural and expressive. The user can choose from a range of high-quality Bangla language voices, which can be customized to suit different applications and preferences. In terms of accuracy for Bangla language TTS synthesis, Amazon Polly has been found to produce highly natural-sounding

speech output. In a study by [22], the performance of Amazon Polly for Bangla language TTS synthesis was evaluated, and the results showed that the system was able to produce highly natural-sounding speech output with an overall mean opinion score (MOS) of 4.4 out of 5. One of the strengths of Amazon Polly is its high accuracy and naturalness, which is achieved through advanced neural network models and machine learning algorithms. Additionally, being a cloud-based service, Amazon Polly is easily accessible to users and developers, and can be integrated into various applications and platforms. In summary, Amazon Polly is a powerful and accurate system for Bangla language TTS synthesis, which provides high-quality, natural-sounding speech output and can be easily implemented through its cloud-based API.

Chapter 4

Methodology

4.1 Workflow

Based on our limited research, we believe that the workflow is the most important aspect of a study. Figure 4.1 depicts the steps in our work cycle. This approach was taken in order to achieve the desired results or to maximize performance. The entire project has been divided into four sections: speech recognition, text summarization, sentence similarity, and text-to-speech. We used the Wav2vec2 deep learning model to recognize speech, the seq2seq model to summarize text, the doc2vec model to detect sentence similarity, and finally the gTTS (Google Text-to-Speech) Python library to turn text into speech using Google's text-to-speech engine.

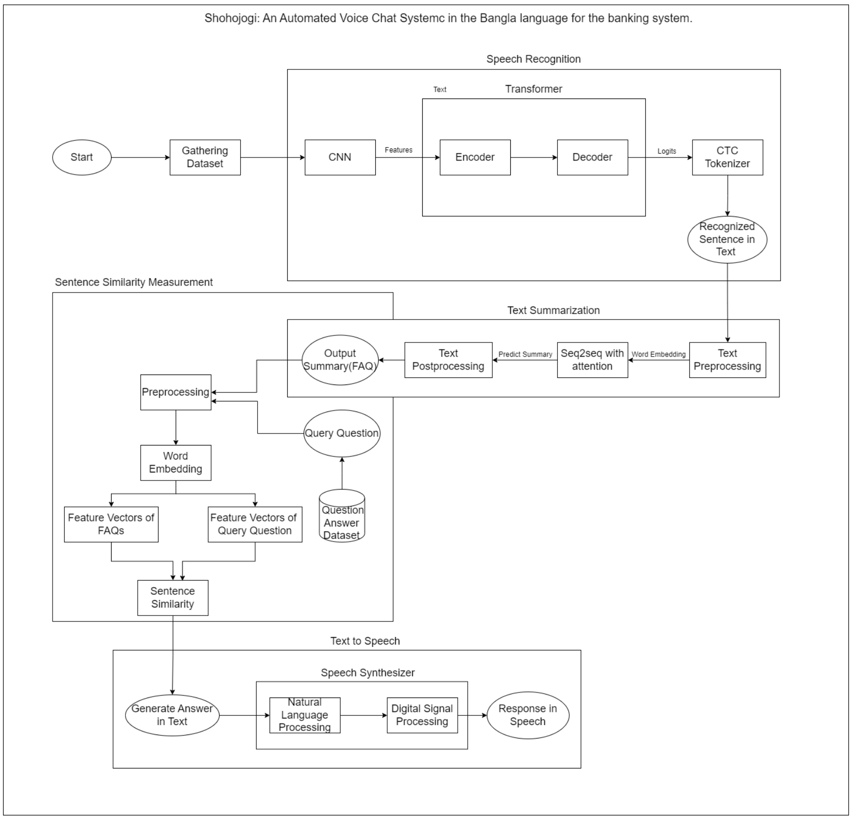


Figure 4.1: WorkFlow

4.2 Speech Recognition

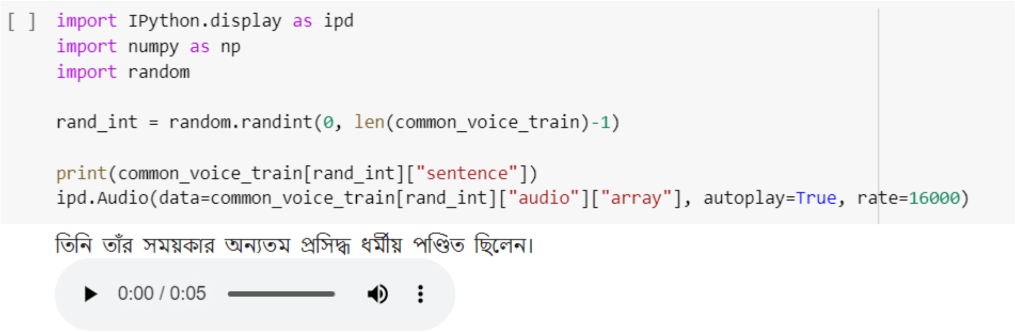
4.2.1 Dataset

The dataset we used for Bengali speech-to-text conversion has been taken from Mozilla's common voice platform, we can address it as Bengali Common Voice Speech Dataset(Corpus v9.0). This dataset contains 400 hours of recorded Bengali speech data in mp3 format from a total number of 19863 speakers. Moreover, The dataset includes recordings from people of different ages, genders, and dialects, as well as people with different levels of education and accents. This diversity is important because it helps to ensure that the speech recognition system can handle a wide range of input. We initially started with the OPENSLR-SLR53-Bengali dataset, however, this dataset has only 196K utterances spoken by a total of 505 speakers only. Therefore, we use Bengali Common Voice Speech Dataset for training in this part of our research.

4.2.2 Preprocessing

The train split of the Bengali Common Voice Speech Dataset includes 206,951 mp3 files with the matching Bengali transcriptions, as well as some meta-data like upvotes, downvotes, gender, etc. We decided to train using only the subset with more upvotes than downvotes because we discovered that 5536 (13%) out of 42941 voted data was untrustworthy. First, a continuous log-mel spectrogram representation of the raw audio data is created for this model by sampling it at a rate of 16 kHz. Frequencies above a specific threshold are rendered logarithmically on a log-mel spectrogram. (the corner frequency). The spectrogram that results is then standardized to have a zero mean and unit variance across all channels. In order to improve the model's resistance to fluctuations in speech patterns and background noise, the audio data is also enhanced with random pitch shifting, temporal stretching, and background noise injection. To utilize in the creation and assessment of the model, the preprocessed data is finally divided into training, validation, and testing sets. We divided the dataset into two groups, with 90% of the dataset put aside for the model's training and 10% remaining for testing. The training dataset was again divided into two groups, with 90% of the dataset being used to train the model and 10% being preserved for validation.

In addition to the above phases, our model also employs a CTC tokenizer for additional data preprocessing. The CTC tokenizer is a form of sequence-to-sequence model that converts audio signals into the textual transcriptions that correspond to them. These transcriptions are then utilized as ground truth labels for the training process. With the help of this tokenizer, the model can learn to recognize speech patterns at the level of phonemes, or individual speech sounds, which can increase the accuracy and robustness of its speech recognition capabilities. Furthermore, feature extraction is yet another crucial stage in this model's data preprocessing. To extract high-level characteristics from the log-mel spectrogram representation of the audio data, the model employs a convolutional neural network. With the help of these features, which record details about the temporal and spectral properties of the audio data, the model may learn to detect speech patterns at various scales, from single phonemes to complete phrases and sentences. In order to create the final textual transcriptions of the input audio signals, the extracted characteristics are then fed into a transformer-based encoder-decoder architecture.



```
[ ] import IPython.display as ipd
    import numpy as np
    import random

    rand_int = random.randint(0, len(common_voice_train)-1)

    print(common_voice_train[rand_int]["sentence"])
    ipd.Audio(data=common_voice_train[rand_int]["audio"]["array"], autoplay=True, rate=16000)
```

তিনি তাঁর সময়কার অন্যতম প্রসিদ্ধ ধর্মীয় পণ্ডিত ছিলেন।

Figure 4.2: Voice Sample Rate

4.2.3 XLS-R Wav2vec2 Model

Wav2Vec2 is a pre-trained model for Automatic Speech Recognition (ASR) that was released in September 2020 by Alexei Baevski, Michael Auli, and Alex Conneau. Soon after the better performance of Wav2Vec2 was shown on LibriSpeech, one of the most popular ASR datasets, Facebook AI showed out XLSR, a multi-language version of Wav2Vec2. XLSR stands for cross-lingual speech representations and means that the model may learn speech representations that are useful in multiple languages. The new version of XLSR, named XLS-R (referring to the ”XLM-R for Speech”), was released by Arun Babu, Changhan Wang, Andros Tjandra, and others in November 2021. For self-supervised pre-training, XLS-R used almost half a million hours of audio data in 128 languages. It comes in sizes from 300 million to two billion parameters.During self-supervised pre-training, XLS-R learns contextualized speech representations by randomly masking feature vectors before sending them to a transformer network. This is similar to how BERT’s masked language modeling goal works. For fine-tuning, a single linear layer is placed on top of the already-trained network to train the model on labeled data of audio downstream tasks like speech recognition.

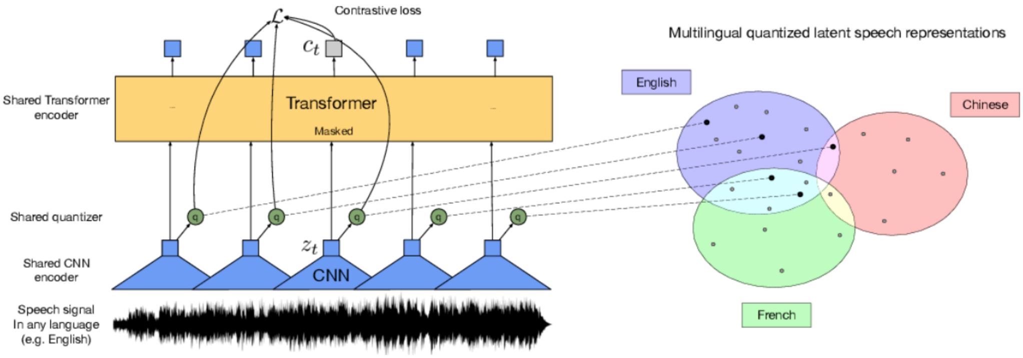

$\mathcal{L}$
Contrastive loss
c_t
Multilingual quantized latent speech representations
English
Chinese
Shared Transformer encoder
Transformer
Masked
Shared quantizer
q
z_t
Shared CNN encoder
CNN
Speech signal In any language (e.g. English)
French

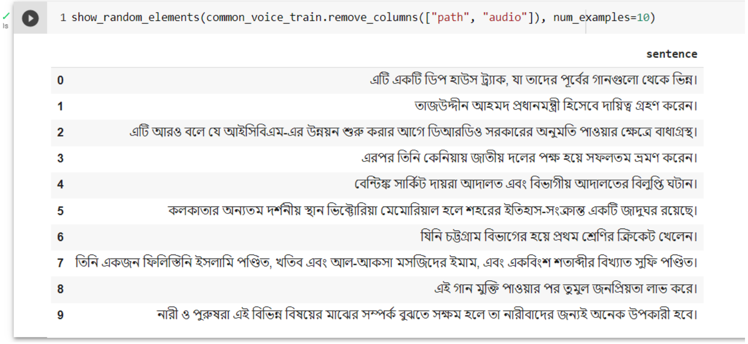
Figure 4.3: Fine-Tune XLSR-Wav2Vec2 for low-resource ASR with Transformers

4.2.4 CTC Algorithm and Wav2Vec2 CTCTokenizer

The Connectionist Temporal Classification (CTC) algorithm is a neural network-based approach used for sequence-to-sequence prediction tasks, like speech recognition. Alex Graves came up with the idea in 2006, and it has since become a popular method in the area. Wav2vec2 is used with CTC to recognize speech by combining a pre-trained Wav2vec2 model with a CTC decoder. The Wav2Vec2 CTCTokenizer is a tokenizer that was made to work with wav2vec2 and a CTC decoder. Wav2Vec2 CTCTokenizer turns the pre-processed audio into a series of vectors. To create a sequence of hidden representations, it feeds the vector sequence into the pre-trained wav2vec2 model. It turns the audio input into a series of vectors that the wav2vec2 model can handle and adds a special ”blank” token that the CTC decoder uses to show spaces between characters. The tokenizer also adds start-of-sequence and end-of-sequence tokens to show where the input sequence begins and ends.

4.2.5 Model Implementation and Training

On the LibriSpeech test/test-other sets, wav2vec2 presently produces state-of-the-art WER of 1.4% / 2.6%. As a result, it was an ideal option for Bengali ASR. As a starting point for training, two methods were considered: a self-supervised pre-trained model (facebook/wav2vec2-largexlsr-53) and an existing fine-tuned model (arijitx/wav2vec2-xls-r-300m-bengali) convergent on another comparable dataset. We observed that fine-tuning an already convergent model marginally reduces performance on the target dataset after early testing. As a result, we decided that the self-supervised pretrained model facebook/wav2vec2-large-xlsr-53 should be used as the foundation for fine-tuning.

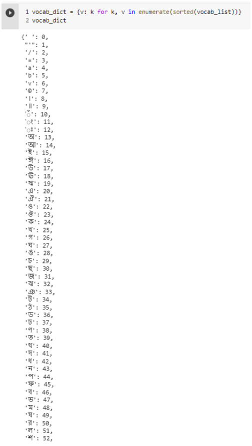


```
show_random_elements(common_voice_train.remove_columns(["path", "audio"]), num_examples=10)
```

	sentence
0	এটি একটি ডিপ হাউস ট্র্যাক, যা তাদের পূর্বের গানগুলো থেকে ভিন্ন।
1	তাজউদ্দীন আহমদ প্রধানমন্ত্রী হিসেবে দায়িত্ব গ্রহণ করেন।
2	এটি আরও বলে যে আইসিবিএম-এর উন্নয়ন শুরু করার আগে ডিআরডিও সরকারের অনুমতি পাওয়ার ক্ষেত্রে বাধাগ্রস্থ।
3	এরপর তিনি কেনিয়ায় জাতীয় দলের পক্ষ হয়ে সফলতম ভ্রমণ করেন।
4	বেন্টিঙ্ক সার্কিট দায়রা আদালত এবং বিভাগীয় আদালতের বিলুপ্তি ঘটান।
5	কলকাতার অন্যতম দর্শনীয় স্থান ভিক্টোরিয়া মেমোরিয়াল হলে শহরের ইতিহাস-সংক্রান্ত একটি জাদুঘর রয়েছে।
6	যিনি চট্টগ্রাম বিভাগের হয়ে প্রথম শ্রেণির ক্রিকেট খেলেন।
7	তিনি একজন ফিলিস্তিনি ইসলামি পণ্ডিত, খতিব এবং আল-আকসা মসজিদের ইমাম, এবং একবিংশ শতাব্দীর বিখ্যাত সুফি পণ্ডিত।
8	এই গান মুক্তি পাওয়ার পর তুমুল জনপ্রিয়তা লাভ করে।
9	নারী ও পুরুষরা এই বিভিন্ন বিষয়ের মাঝের সম্পর্ক বুঝতে সক্ষম হলে তা নারীবাদের জন্যই অনেক উপকারী হবে।

Figure 4.4: Random Sample Data

For training, we used the pytorch-based version of the Transformer model that was made available by huggingface.co and is maintained by them. We improved the pretrained facebook/wav2vec2-large-xlsr-53, which was trained on unlabeled multilingual speech with the intention of receiving more training on labeled data in the future. There were a total of 30 epochs of training, which added up to almost 50 hours of training using a single Nvidia A100 GPU on collab pro+. For 30 epochs of training, the training run-time was about 23 hours. With a learning rate of 3e4, the AdamW Optimizer was employed. After several early tests that either failed to converge with larger hyperparameters or took a long time to show any real convergence with lower rates, these values were chosen.



```
vocab_dict = {v: k for k, v in enumerate(sorted(vocab_list))}
vocab_dict
```

```
{' ': 0,
 "'": 1,
 '/': 2,
 '=': 3,
 'a': 4,
 'b': 5,
 'v': 6,
 '©': 7,
 '।': 8,
 '॥': 9,
 'ঁ': 10,
 'ং': 11,
 'ঃ': 12,
 'অ': 13,
 'আ': 14,
 'ই': 15,
 'ঈ': 16,
 'উ': 17,
 'ঊ': 18,
 'ঋ': 19,
 'এ': 20,
 'ঐ': 21,
 'ও': 22,
 'ঔ': 23,
 'ক': 24,
 'খ': 25,
 'গ': 26,
 'ঘ': 27,
 'ঙ': 28,
 'চ': 29,
 'ছ': 30,
 'জ': 31,
 'ঝ': 32,
 'ঞ': 33,
 'ট': 34,
 'ঠ': 35,
 'ড': 36,
 'ঢ': 37,
 'ণ': 38,
 'ত': 39,
 'থ': 40,
 'দ': 41,
 'ধ': 42,
 'ন': 43,
 'প': 44,
 'ফ': 45,
 'ব': 46,
 'ভ': 47,
 'ম': 48,
 'য': 49,
 'র': 50,
 'ল': 51,
 'শ': 52,
```

Figure 4.5: 81 Vocabulary Found

We employ validation metrics character error rate (CER) and preserve the latest two model checkpoints for monitoring on the training process.

4.3 Text Summarization

This study uses some deep learning approaches to build a model for text summarization. To solve text-related issues, RNN is used. Before using these strategies, we used the BANS dataset from Kaggle. All preparation operations, including lexical analysis, contraction addition, stop word deletion, whitespace creation, character punctuation, lemmatization, and others, are carried out in order to produce clean texts. Then, word embedding and vocabulary are counted using Word2 Vec. The suggested models RNN Encoder-Decoder and Seq2Seq with an attention mechanism are then employed. To highlight the main difference between the system's outputs and the own dataset, they are compared to results from another dataset.

4.3.1 Dataset

With more than 250 million speakers, Bengali is the ninth most widely spoken language in the world. However, Bangla NLP has a serious lack of resources, especially in text summarization. For the purpose of creating NLP models that can efficiently summarize Bangla text, these datasets are very useful resources. Fortunately, the number of datasets for summarizing Bangla text is growing. To help in the advancement of this field of study, researchers and organizations have begun to produce and disseminate Bangla text summary datasets. The BANS bengali dataset from Kaggle [37], which includes 19k short articles and 19k short summaries for the evaluation of the outputs, was used for our research.

4.3.2 Preprocessing

To enable the model to use clean articles, the dataset must first be created and then cleaned and preprocessed.The initial step in data preprocessing is lexical analysis. Before the syntax is broken down into a list of tokens, the changed source code from language preprocessors expressed in sentences is taken first, with all whitespace and comments removed. There are word contractions in every language, including Bengali, that the computer cannot read and comprehend.The entire meanings of the contractions are thus added. Words are eliminated to remove unnecessary details. Punctuation and other characters that could impede text summary processing are also eliminated. The process of lemmatization is then employed to ascertain the word's etymology. The lemmatization method collects all of the word's inflected forms and reduces them to the word's dictionary-based root form. Words are divided into parts of speech (POS) by using the grammar rules. After completing all preprocessing stages, the texts have been cleaned to remove any unnecessary whitespace, characters, stop words, punctuation, or other formatting from the articles and summaries. Both pure texts and summaries are acceptable input sequences for the Bengali automatic news summarizing model.

4.3.3 Model implementation

Count Vocabulary: Vocabulary Before applying word embedding, we must count the vocabulary from the BANS Dataset. In this dataset, there are 16,284 different words that are only found in articles. We use 2,712 words overall. We received a total of 19,326 words and 1726 unique terms for summaries.

Word Embedding: Word embedding requires a Word2Vec file with a machine-readable numeric value for each word. The "bn w2v model," a collection of pre-trained Bengali word vector files, is used in this technique. The desired related terms are produced by the model by using the vectors as inputs.

Models: Compared to most other deep learning techniques, LSTM is better for text summarization. To create accurate and pertinent summaries of news stories, the Seq2Seq learning with attention technique is also applied.

RNN Encoder-Decoder Architecture: RNN has become an effective seq2seq prediction method. The key advantages of this approach are its capacity to train a single end-to-end model particularly on source and target phrases, as well as its control over variable-length input and output text sequences.The Seq2Seq learning framework with an attention mechanism is used in this model.

Sequence to Sequence (Seq2Seq) learning with Attention: Even if the lengths of the input and output may differ, the main objective of a Seq2Seq model is to map a fixed-length input to a fixed-length output. A typical LSTM network cannot map all of the words in the input sequence to the words in the output sequence. The beginning and end of each sequence are marked by tokens that are included in the sequence.The encoder is given the hidden state ht in:

$$h_t = f\left(W_{hh} h_{t-1} + W_{hx} x_t\right)$$

Here, the input vector, xt, and the weights from the prior states are used to calculate the hidden states. The hidden state ht is provided in Eq for the decoder.

$$h_t = f\left(W^{hh} h_t - 1\right)$$

Using Eq., the output at time step t is computed.

$$y_t = \operatorname{softmax}\left(W^s h_t\right)$$

This Seq2Seq Model is graphically depicted in Figure.

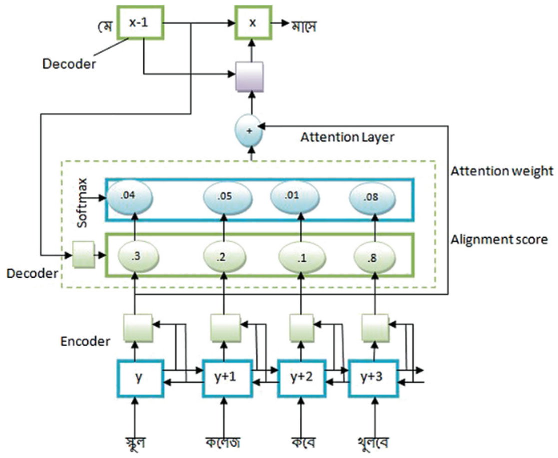


Figure 4.6: Seq2Seq model with attention

4.4 Sentence Similarity Measurement

4.4.1 Dataset

For the section, we make our dataset manually. Questions are gathered from the website and the Facebook page of The City Bank, a well-known bank in Bangladesh. We generated a data collection containing 110 questions which are the most frequently asked questions by customers. Moreover, we have also generated the most relevant answer to the questions.

4.4.2 Data Preprocessing

In the preprocessing part, we first break down the sentences into individual tokens, which are usually words in Bengali. This is done using a tokenizer, which identifies the word boundaries in the text. Stop words are common words in a language that do not carry much meaning. Therefore, we filter out these words from the sentences to reduce noise in the data and improve the accuracy of the similarity calculations. Then we use stemming, Bengali, like many other languages, has inflected forms of words that can vary depending on tense, case, and other factors. Stemming involves reducing each word to its base form, or stem, which can help to group together words with similar meanings. After that, POS (Part-of-speech) tagging is employed. This involves labeling each word in the sentence with its parts of speech, such as noun, verb, adjective, or adverb. POS tagging can help to identify the relationships between words and phrases in the sentence and improve the accuracy of the similarity calculations. Overall, we designed the preprocessing part to clean and normalize the text data, reduce noise and variability, and capture relevant linguistic information that can be used to calculate sentence similarity.

4.4.3 Model Implementation and Training

For creating vector representations of sentences, we implement the Doc2Vec model. The Doc2Vec model, which is based on neural networks, creates vector representations of sentences, paragraphs, and entire documents. It is a development of the well-known Word2Vec model, which creates vector representations of each individual word.

The main idea behind Doc2Vec is to represent each document as a vector in a high-dimensional space, with the goal of having vectors that are close to one another in this space to represent documents with similar semantic meanings. Using the word's context and the document tag, a neural network is trained to predict the context of each word in a document. Each document has a unique identification called a "document tag" that is used to distinguish between the contexts of various documents. The distributed memory (DM) model and the distributed bag of words (DBOW) model are two different variations of the Doc2vec algorithm. In this study, the DBOW model was employed. In this variation, a word in the document is predicted by the model using the document tag as input. The document tag vector alone is used by the model to create a vector representation of the document.

Using the Doc2Vec implementation from the Gensim library, we trained the doc2vec model on the preprocessed sentences. We set the hyperparameters for the model to be 100 for the vector size and 10 for the number of epochs.

The model is trained to predict the context of each word in the dataset given its context and document tag. The model is given the input data and the vocabulary. During the training phase, words from the dataset's documents are randomly selected, and stochastic gradient descent is used to update the model's weights.

After training, the model is run on the preprocessed sentences to produce vector representations. The word vectors of the constituent words are averaged to create the vector representation for each sentence.

4.4.4 Cosine Similarity

For measuring sentence similarity, there are numerous algorithms. To calculate Bengali sentence similarity for our work, we primarily use cosine similarity. The pre-trained Doc2Vec model is loaded and then used to create vector representations of the input sentences. This is accomplished by passing the preprocessed sentences to the Doc2Vec model's infer vector() method, which will then return a vector representation of each sentence. Finally, we used NumPy's cosine similarity() function to determine the cosine similarity between the vector representations of the two sentences. A measure of similarity, the cosine similarity between two vectors, goes from -1 (completely dissimilar) to 1. (identical). If both sentences have a value of 0, they are orthogonal. (i.e., have no similarity).

$$\vec{A}.\vec{B} = ||A||||B|| \cos\theta$$

4.4.5 Jaccard Similarity

We also use the Jaccard similarity algorithm for the sentence similarity calculation, but the output is quite low compared to cosine similarity. Therefore, we implemented the cosine similarity algorithm for this research.

4.5 Text-to-speech (TTS) Synthesis

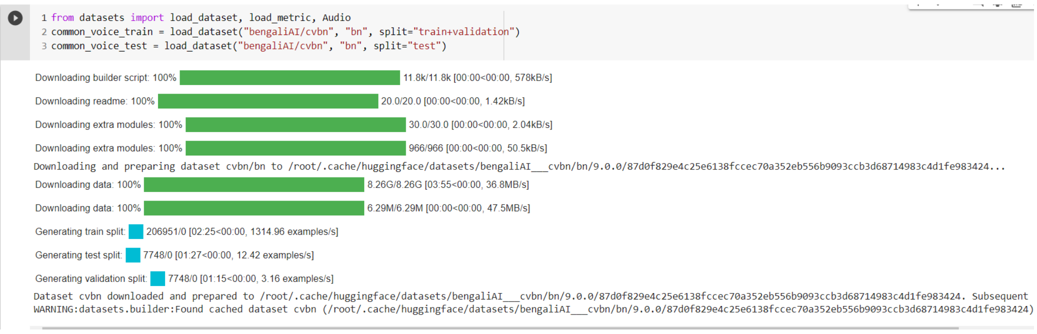
For the final part of the process, the answer needs to be converted from raw text to an audio version for the customer. For this work, we have used the gTTS library (Google Text-to-Speech), which uses Google's Text-to-Speech API to convert text into audio files. Customer's Desired Answer is working as the input text, which has been passed to the gTTS library using the gTTS() function. The library sends a request to Google's Text-to-Speech API, specifying the Bengali language and the input text to be converted to speech. Google's Text-to-Speech API takes the request and turns it into an MP3 file that sounds like the text that was given. The Bengali language support in gTTS is achieved through the use of the Bengali language code (bn) in the request to Google's Text-to-Speech API. The Bengali language text is passed in Unicode format to the gTTS() function for processing.

Chapter 5

Experiments and Results Analysis

5.1 Speech Recognition

Here, we imported the Bengali Common voice dataset and spitted our dataset in two one is for the train set where the test size is 10% of the main dataset and the train set is 90%, then again we splitted the train set into train set and validation set where train set has 81% data of the main dataset and validation set has 9% data.



```
from datasets import load_dataset, load_metric, Audio
common_voice_train = load_dataset("bengaliAI/cvbn", "bn", split="train+validation")
common_voice_test = load_dataset("bengaliAI/cvbn", "bn", split="test")
```

```
Downloading builder script: 100% 11.8k/11.8k [00:00<00:00, 578kB/s]
Downloading readme: 100% 20.0/20.0 [00:00<00:00, 1.42kB/s]
Downloading extra modules: 100% 30.0/30.0 [00:00<00:00, 2.04kB/s]
Downloading extra modules: 100% 966/966 [00:00<00:00, 50.5kB/s]
Downloading and preparing dataset cvbn/bn to /root/.cache/huggingface/datasets/bengaliAI___cvbn/bn/9.0.0/87d0f829e4c25e6138fccec70a352eb556b9093ccb3d68714983c4d1fe983424...
Downloading data: 100% 8.26G/8.26G [03:55<00:00, 36.8MB/s]
Downloading data: 100% 6.29M/6.29M [00:00<00:00, 47.5MB/s]
Generating train split: 206951/0 [02:25<00:00, 1314.96 examples/s]
Generating test split: 7748/0 [01:27<00:00, 12.42 examples/s]
Generating validation split: 7748/0 [01:15<00:00, 3.16 examples/s]
Dataset cvbn downloaded and prepared to /root/.cache/huggingface/datasets/bengaliAI___cvbn/bn/9.0.0/87d0f829e4c25e6138fccec70a352eb556b9093ccb3d68714983c4d1fe983424. Subsequent
WARNING:datasets.builder:Found cached dataset cvbn (/root/.cache/huggingface/datasets/bengaliAI___cvbn/bn/9.0.0/87d0f829e4c25e6138fccec70a352eb556b9093ccb3d68714983c4d1fe983424)
```

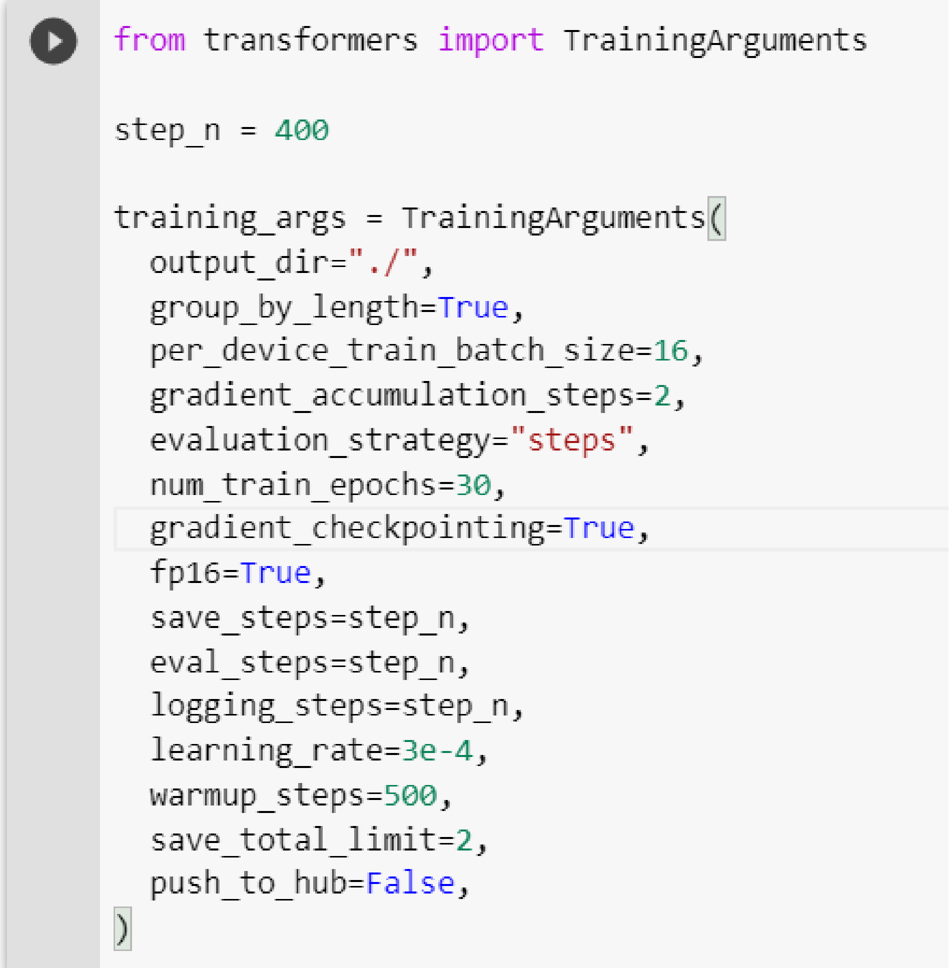
Figure 5.1: Importing Dataset from Common voice

In this research, we used CER(Character Error Rate) metric for evaluation. Character Error Rate (CER) is a metric of the performance of an automatic speech recognition (ASR) system. This value indicates the percentage of characters that were incorrectly predicted. The lower the value, the better the performance of the ASR system with a CharErrorRate of 0 being a perfect score.

CER calculation is based on the concept of Levenshtein distance, where we count the minimum number of character-level operations required to transform the ground truth text (aka reference text) into the OCR output.
It is represented with this formula:

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

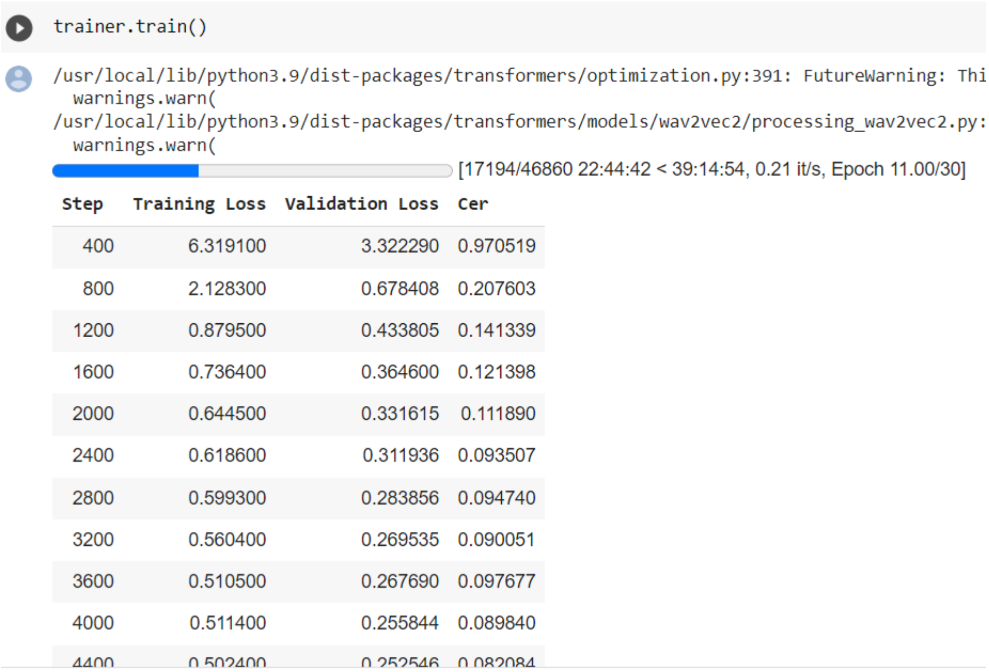
In this instance, the group by length option is set to True, which means that in order to increase training efficiency, the training data will be grouped by similar sequence lengths. Each GPU will process 16 samples at a time during training because the per device train batch size option is set to 16. The gradient accumulation steps option is set to 2, which means that before updating the model parameters, gradients will be accumulated over 2 batches. The evaluation will be carried out every step n steps during training if the evaluation strategy option is set to "steps." The total number of training epochs is specified by the num train epochs option, which is set to 30. If the gradient checkpointing option is set to True, recalculating forward activations during backward pass enables memory-efficient training. When the fp16 option is set to True, mixed precision training is possible and speeds up training. All three of the save steps, eval-steps, and logging steps options are set to step n, which indicates how frequently checkpoints are saved, evaluations are conducted, and training progress is recorded. The initial learning rate for the optimizer is specified by the learning rate option, which is set to 3e-4. The learning rate scheduler's warmup steps are specified by the warmup-steps option, which is set to 500. The maximum number of checkpoints to save during training is specified by the save total limit option, which is set to 2.



```
from transformers import TrainingArguments

step_n = 400

training_args = TrainingArguments(
  output_dir="./",
  group_by_length=True,
  per_device_train_batch_size=16,
  gradient_accumulation_steps=2,
  evaluation_strategy="steps",
  num_train_epochs=30,
  gradient_checkpointing=True,
  fp16=True,
  save_steps=step_n,
  eval_steps=step_n,
  logging_steps=step_n,
  learning_rate=3e-4,
  warmup_steps=500,
  save_total_limit=2,
  push_to_hub=False,
)
```

Figure 5.2: Training Parameters

After training the wav2vec2 model for almost 24h hours in collab pro+ with a single Nvidia A100 GPU the CER was 0.06543, though it was not the lowest. The lowest CER was 0.060310 at step 16400. The training loss was 0.104400 and validation loss was 0.333588



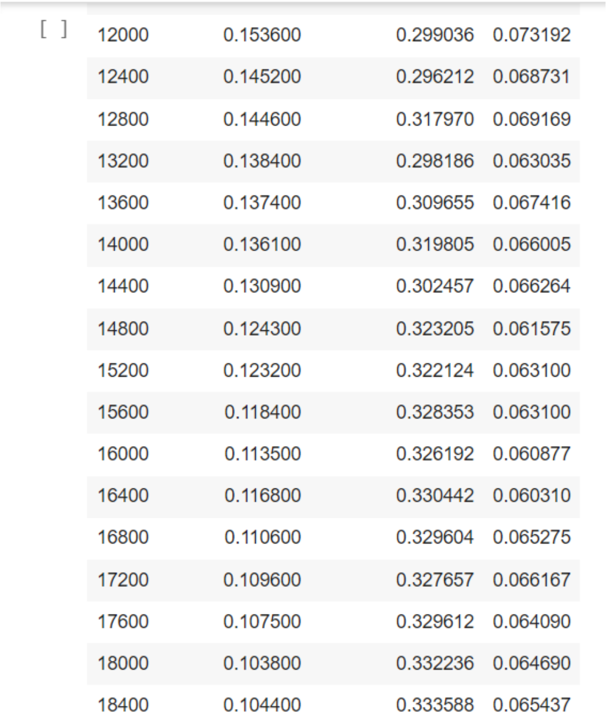
```
trainer.train()
```

```
/usr/local/lib/python3.9/dist-packages/transformers/optimization.py:391: FutureWarning: Thi
  warnings.warn(
/usr/local/lib/python3.9/dist-packages/transformers/models/wav2vec2/processing_wav2vec2.py:
  warnings.warn(
```

[17194/46860 22:44:42 < 39:14:54, 0.21 it/s, Epoch 11.00/30]

Step	Training Loss	Validation Loss	Cer
400	6.319100	3.322290	0.970519
800	2.128300	0.678408	0.207603
1200	0.879500	0.433805	0.141339
1600	0.736400	0.364600	0.121398
2000	0.644500	0.331615	0.111890
2400	0.618600	0.311936	0.093507
2800	0.599300	0.283856	0.094740
3200	0.560400	0.269535	0.090051
3600	0.510500	0.267690	0.097677
4000	0.511400	0.255844	0.089840
4400	0.502400	0.252546	0.082084

Figure 5.3: Wav2Vec Model Training



[ ]

12000	0.153600	0.299036	0.073192
12400	0.145200	0.296212	0.068731
12800	0.144600	0.317970	0.069169
13200	0.138400	0.298186	0.063035
13600	0.137400	0.309655	0.067416
14000	0.136100	0.319805	0.066005
14400	0.130900	0.302457	0.066264
14800	0.124300	0.323205	0.061575
15200	0.123200	0.322124	0.063100
15600	0.118400	0.328353	0.063100
16000	0.113500	0.326192	0.060877
16400	0.116800	0.330442	0.060310
16800	0.110600	0.329604	0.065275
17200	0.109600	0.327657	0.066167
17600	0.107500	0.329612	0.064090
18000	0.103800	0.332236	0.064690
18400	0.104400	0.333588	0.065437

Figure 5.4: Wav2Vec Model Training

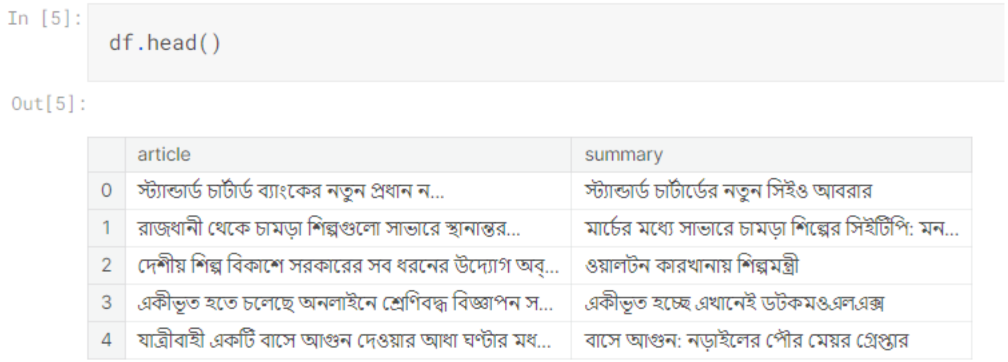
Our aim was to train the data for 100 hours, but due to our limitations to high performance GPU, we only train for upto 50hours with collab pro+. To make up for it, we trained our data with 10,000, 15,000, and 25,000 sample inputs to show how training loss, validation loss and CER improves, each time we increase the number of sample inputs so that we can say, the more hour we train, the less training loss, validation loss and CER we will get.

Sample Inputs	Training Hour	Training Loss	Validation Loss	CER
10000	20	1.750400	1.221247	0.821042
I5000	30	0.872906	0.62794	0.297436
25000	50	0.104400	0.333588	0.06543

Table 5.1: Comparison Between Different Sample Input

5.2 Text Summarization

Text summarization was generated at the very end of our research, therefore its implementation and perfection are still in the training phase. We wanted to summarize the generated text we got from speech to text conversion part so that it would be easier and more efficient to calculate the sentence similarity. However, we barely manage to generate some summaries for Bengali text articles, however, they are not very accurate. So, we can say that this part of our research is still ongoing.



```
In [5]:  df.head()

Out[5]:
```

	article	summary
0	স্ট্যান্ডার্ড চার্টার্ড ব্যাংকের নতুন প্রধান ন...	স্ট্যান্ডার্ড চার্টার্ডের নতুন সিইও আবরার
1	রাজধানী থেকে চামড়া শিল্পগুলো সাভারে স্থানান্তর...	মার্চের মধ্যে সাভারে চামড়া শিল্পের সিইটিপি: মন...
2	দেশীয় শিল্প বিকাশে সরকারের সব ধরনের উদ্যোগ অব্...	ওয়ালটন কারখানায় শিল্পমন্ত্রী
3	একীভূত হতে চলেছে অনলাইনে শ্রেণিবদ্ধ বিজ্ঞাপন স...	একীভূত হচ্ছে এখানেই ডটকমওএলএক্স
4	যাত্রীবাহী একটি বাসে আগুন দেওয়ার আধা ঘণ্টার মধ...	বাসে আগুন: নড়াইলের পৌর মেয়র গ্রেপ্তার

Figure 5.5: Samples From BANS dataset

This sample data is generated from BANS dataset that we used to implement Bangla Text Summarization.
This graph shows how many words are there in each of the articles and summaries. For articles, the highest number of words is 60, and for summaries, it’s between 5 to 10 words.

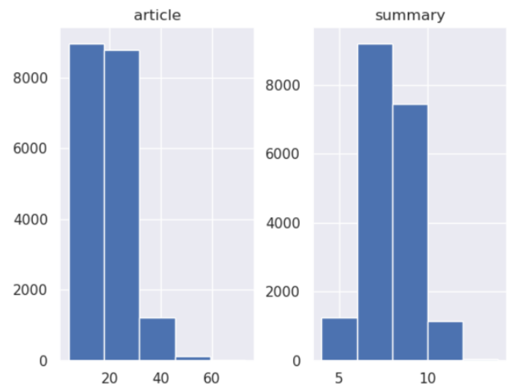
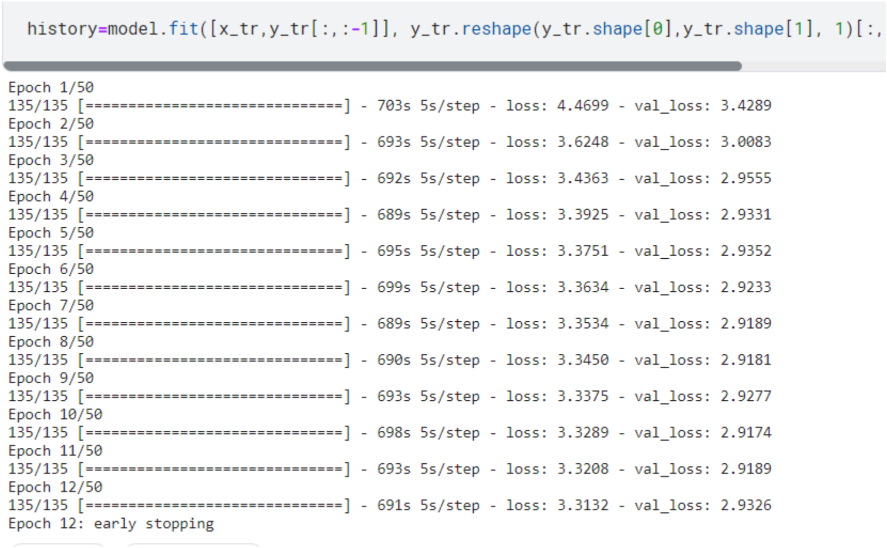


Figure 5.6: Dataset Word Count

Our main goal was to train the whole seq2seq with 50 epochs but because validation loss was not decreasing for the last two epochs, therefore, the training was showing an early stop. We are in the figuring-out phase of this part of our research. After 12 epochs step loss was 3.3132 and the value loss was 2.9326.

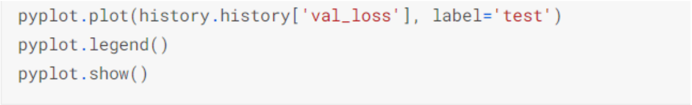


```
history=model.fit([x_tr,y_tr[:,:-1]], y_tr.reshape(y_tr.shape[0],y_tr.shape[1], 1)[:,
```

```
Epoch 1/50
135/135 [==============================] - 703s 5s/step - loss: 4.4699 - val_loss: 3.4289
Epoch 2/50
135/135 [==============================] - 693s 5s/step - loss: 3.6248 - val_loss: 3.0083
Epoch 3/50
135/135 [==============================] - 692s 5s/step - loss: 3.4363 - val_loss: 2.9555
Epoch 4/50
135/135 [==============================] - 689s 5s/step - loss: 3.3925 - val_loss: 2.9331
Epoch 5/50
135/135 [==============================] - 695s 5s/step - loss: 3.3751 - val_loss: 2.9352
Epoch 6/50
135/135 [==============================] - 699s 5s/step - loss: 3.3634 - val_loss: 2.9233
Epoch 7/50
135/135 [==============================] - 689s 5s/step - loss: 3.3534 - val_loss: 2.9189
Epoch 8/50
135/135 [==============================] - 690s 5s/step - loss: 3.3450 - val_loss: 2.9181
Epoch 9/50
135/135 [==============================] - 693s 5s/step - loss: 3.3375 - val_loss: 2.9277
Epoch 10/50
135/135 [==============================] - 698s 5s/step - loss: 3.3289 - val_loss: 2.9174
Epoch 11/50
135/135 [==============================] - 693s 5s/step - loss: 3.3208 - val_loss: 2.9189
Epoch 12/50
135/135 [==============================] - 691s 5s/step - loss: 3.3132 - val_loss: 2.9326
Epoch 12: early stopping
```

Figure 5.7: Seq2seq model training

This graph is generated from the training, showing the training and testing value loss.



```
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()
```

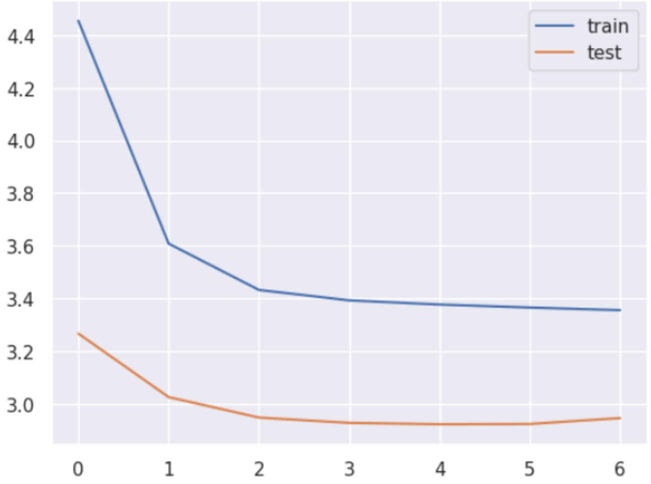
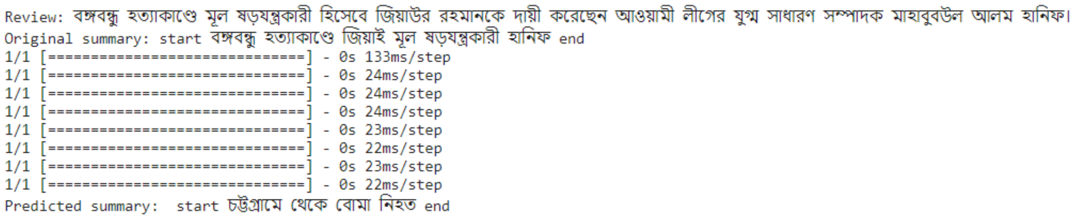


Figure 5.8: Showing the training and testing value loss

This is one of the predictions, which is nowhere near the original summary. We are still working on this, hopefully, we can get our desired result soon.



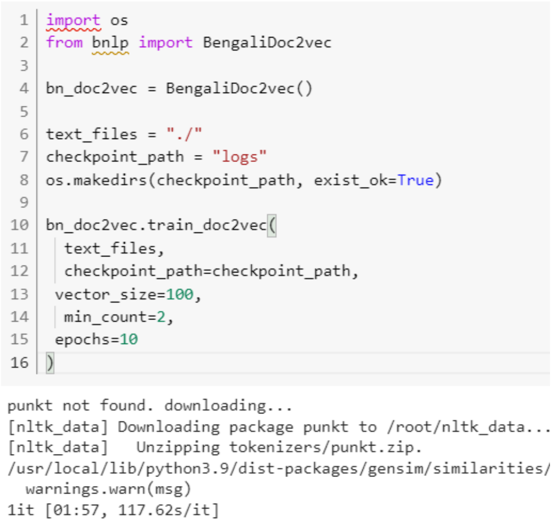
```
Review: বঙ্গবন্ধু হত্যাকাণ্ডে মূল ষড়যন্ত্রকারী হিসেবে জিয়াউর রহমানকে দায়ী করেছেন আওয়ামী লীগের যুগ্ম সাধারণ সম্পাদক মাহবুবউল আলম হানিফ।
Original summary: start বঙ্গবন্ধু হত্যাকাণ্ডে জিয়াই মূল ষড়যন্ত্রকারী হানিফ end
1/1 [==============================] - 0s 133ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
Predicted summary:  start চট্টগ্রামে থেকে বোমা নিহত end
```

Figure 5.9: Predicted summary generation

5.3 Sentence Similarity Measurement

For this section, we primarily employ the BNLP, a natural language processing toolkit for the Bengali language. Implementing this tool helps us tokenize Bengali text through three steps: embedding Bengali documents, Bengali POS tagging, and Bangla text cleaning.We were able to use the doc2vec model for Bangla sentence similarity measurement with the help of BNLP. The requisite modules are imported in this code, and a BengaliDoc2vec class instance is initialized using the bnlp module. Next, we specify the path to the text files we'll use to train the Doc2Vec model and the location where we'll store the finished product.
The Doc2Vec model is then trained on the given text files using the train doc2vec() method of the BengaliDoc2vec class. Along with some hyperparameters, such as the vector size being set at 100, the minimum count being 2, and the number of epochs being 10, we pass the path to the text files.After preprocessing the input text files, the train doc2vec() method trains the Doc2Vec model and saves the trained

model to the designated checkpoint path. After training is finished, we can use the Doc2Vec model to determine how similar sentences in Bengali are.

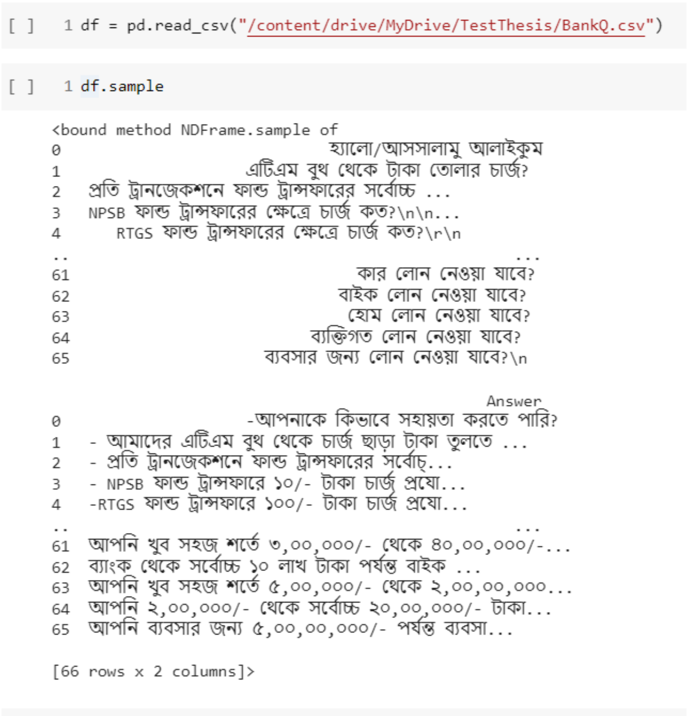


```python
import os
from bnlp import BengaliDoc2vec

bn_doc2vec = BengaliDoc2vec()

text_files = "./"
checkpoint_path = "logs"
os.makedirs(checkpoint_path, exist_ok=True)

bn_doc2vec.train_doc2vec(
    text_files,
    checkpoint_path=checkpoint_path,
  vector_size=100,
    min_count=2,
  epochs=10
)
```

```
punkt not found. downloading...
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
/usr/local/lib/python3.9/dist-packages/gensim/similarities/
  warnings.warn(msg)
1it [01:57, 117.62s/it]
```

Figure 5.10: Doc2vec model training

After training the doc2vec model on a pre-trained model called news article doc2vec/bangla news article doc2vec.model from hugging face we imported our own dataset file. The below image is the prove of that.



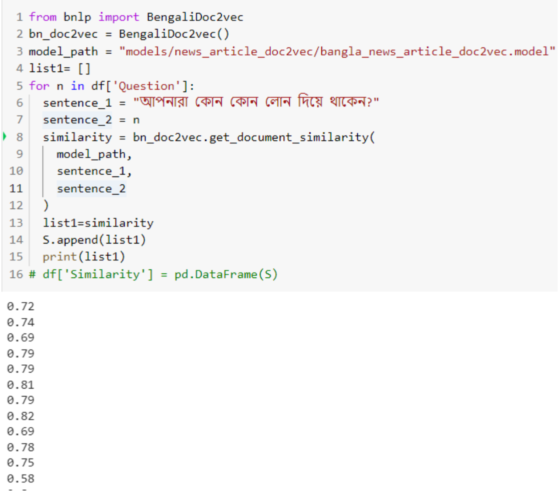
```
[ ]   1 df = pd.read_csv("/content/drive/MyDrive/TestThesis/BankQ.csv")
```

```
[ ]   1 df.sample
```

```
<bound method NDFrame.sample of
0                                    হ্যালো/আসসালামু আলাইকুম
1                          এটিএম বুথ থেকে টাকা তোলার চার্জ?
2   প্রতি ট্রানজেকশনে ফান্ড ট্রান্সফারের সর্বোচ্চ ...
3   NPSB ফান্ড ট্রান্সফারের ক্ষেত্রে চার্জ কত?\n\n...
4       RTGS ফান্ড ট্রান্সফারের ক্ষেত্রে চার্জ কত?\r\n
..                                                   ...
61                                  কার লোন নেওয়া যাবে?
62                                 বাইক লোন নেওয়া যাবে?
63                                  হোম লোন নেওয়া যাবে?
64                              ব্যক্তিগত লোন নেওয়া যাবে?
65                         ব্যবসার জন্য লোন নেওয়া যাবে?\n


                                                 Answer
0                        -আপনাকে কিভাবে সহায়তা করতে পারি?
1   - আমাদের এটিএম বুথ থেকে চার্জ ছাড়া টাকা তুলতে ...
2   - প্রতি ট্রানজেকশনে ফান্ড ট্রান্সফারের সর্বোচ্...
3   - NPSB ফান্ড ট্রান্সফারে ১০/- টাকা চার্জ প্রযো...
4   -RTGS ফান্ড ট্রান্সফারে ১০০/- টাকা চার্জ প্রযো...
..                                                  ...
61  আপনি খুব সহজ শর্তে ৩,০০,০০০/- থেকে ৪০,০০,০০০/-...
62  ব্যাংক থেকে সর্বোচ্চ ১০ লাখ টাকা পর্যন্ত বাইক ...
63  আপনি খুব সহজ শর্তে ৫,০০,০০০/- থেকে ২,০০,০০,০০০...
64  আপনি ২,০০,০০০/- থেকে সর্বোচ্চ ২০,০০,০০০/- টাকা...
65  আপনি ব্যবসার জন্য ৫,০০,০০,০০০/- পর্যন্ত ব্যবসা...

[66 rows x 2 columns]>
```

Figure 5.11: Sample Question and Answer from Dataset

With the aid of the cosine similarity algorithm, this is how we determined the sentence similarity between the condensed customer query and our own dataset. The question similarity between each question in the dataset and the query is what is output. Finding the question and answer that is most closely related to the query takes about two to three minutes once all the outputs have been generated.

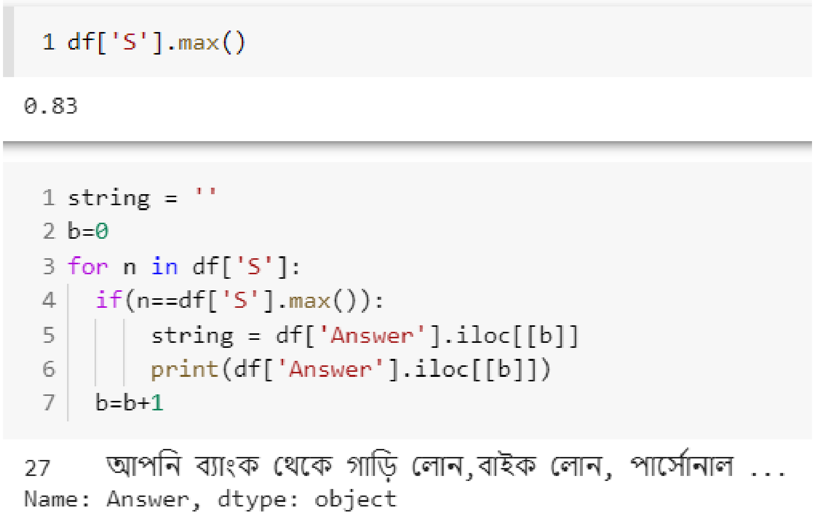


```
from bnlp import BengaliDoc2vec
bn_doc2vec = BengaliDoc2vec()
model_path = "models/news_article_doc2vec/bangla_news_article_doc2vec.model"
list1= []
for n in df['Question']:
  sentence_1 = "আপনারা কোন কোন লোন দিয়ে থাকেন?"
  sentence_2 = n
  similarity = bn_doc2vec.get_document_similarity(
    model_path,
    sentence_1,
    sentence_2
  )
  list1=similarity
  S.append(list1)
  print(list1)
# df['Similarity'] = pd.DataFrame(S)
```

```
0.72
0.74
0.69
0.79
0.79
0.81
0.79
0.82
0.69
0.78
0.75
0.58
```

Figure 5.12: Generating Sentence Similarity

This is the generated response, which is the right response, to the query. It obtains the maximum similarity output and generates the response in accordance with the dataset's question mapping.



```
df['S'].max()
```

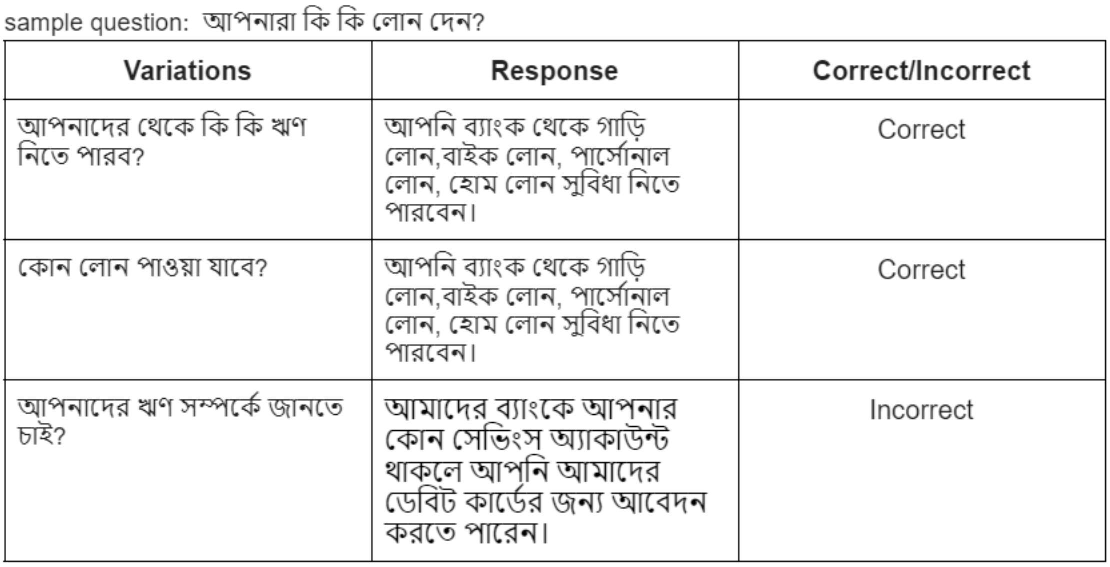
```
0.83
```

```
string = ''
b=0
for n in df['S']:
  if(n==df['S'].max()):
    string = df['Answer'].iloc[[b]]
    print(df['Answer'].iloc[[b]])
  b=b+1
```

```
27    আপনি ব্যাংক থেকে গাড়ি লোন,বাইক লোন, পার্সোনাল ...
Name: Answer, dtype: object
```

Figure 5.13: Generating Response

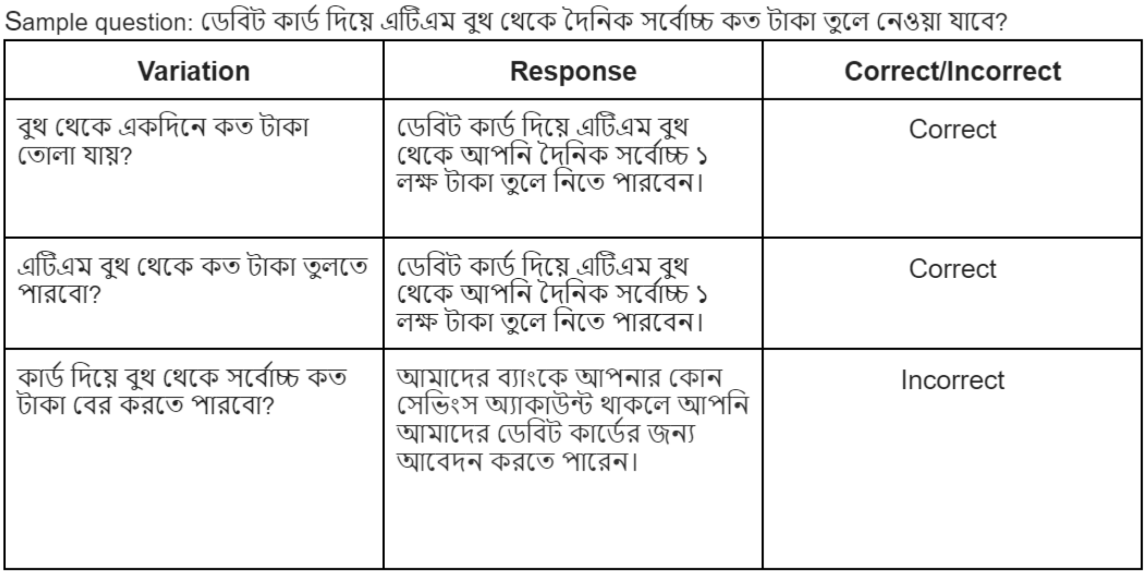
Because the way questions are asked can differ from person to person, we tried asking various questions and observing the outcomes. We wanted to ensure that the system could provide the appropriate response because the question could be posed in various ways in real life. There is always room for development, and the system will get better over time as resources become available.The system typically generates accurate output, which means it generates answers that are relevant to the questions. Several examples of that are as follows:



sample question: আপনারা কি কি লোন দেন?

Variations	Response	Correct/Incorrect
আপনাদের থেকে কি কি ঋণ নিতে পারব?	আপনি ব্যাংক থেকে গাড়ি লোন,বাইক লোন, পার্সোনাল লোন, হোম লোন সুবিধা নিতে পারবেন।	Correct
কোন লোন পাওয়া যাবে?	আপনি ব্যাংক থেকে গাড়ি লোন,বাইক লোন, পার্সোনাল লোন, হোম লোন সুবিধা নিতে পারবেন।	Correct
আপনাদের ঋণ সম্পর্কে জানতে চাই?	আমাদের ব্যাংকে আপনার কোন সেভিংস অ্যাকাউন্ট থাকলে আপনি আমাদের ডেবিট কার্ডের জন্য আবেদন করতে পারেন।	Incorrect

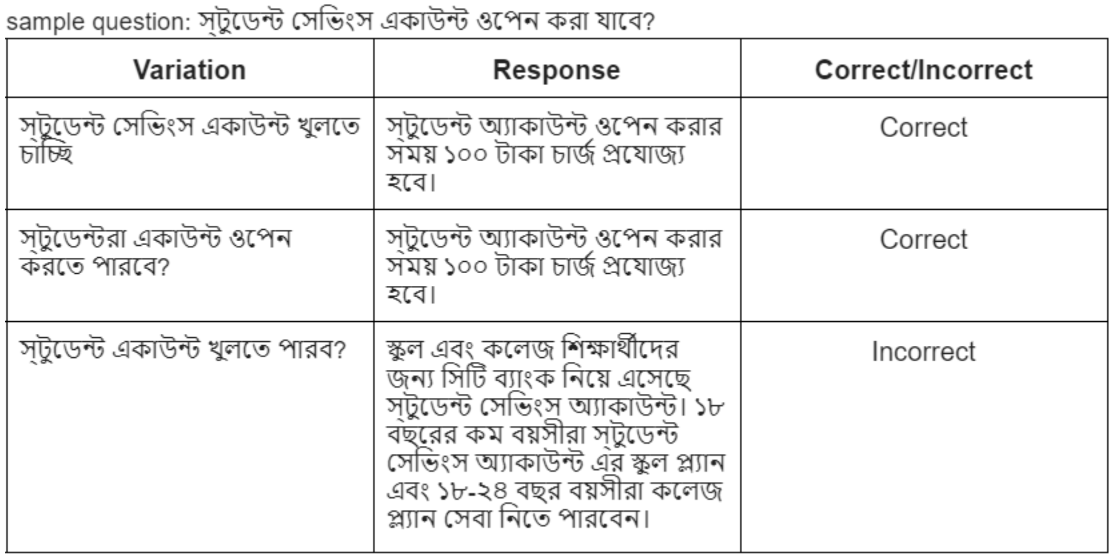
Figure 5.14: Question Variations and Responses



Sample question: ডেবিট কার্ড দিয়ে এটিএম বুথ থেকে দৈনিক সর্বোচ্চ কত টাকা তুলে নেওয়া যাবে?

Variation	Response	Correct/Incorrect
বুথ থেকে একদিনে কত টাকা তোলা যায়?	ডেবিট কার্ড দিয়ে এটিএম বুথ থেকে আপনি দৈনিক সর্বোচ্চ ১ লক্ষ টাকা তুলে নিতে পারবেন।	Correct
এটিএম বুথ থেকে কত টাকা তুলতে পারবো?	ডেবিট কার্ড দিয়ে এটিএম বুথ থেকে আপনি দৈনিক সর্বোচ্চ ১ লক্ষ টাকা তুলে নিতে পারবেন।	Correct
কার্ড দিয়ে বুথ থেকে সর্বোচ্চ কত টাকা বের করতে পারবো?	আমাদের ব্যাংকে আপনার কোন সেভিংস অ্যাকাউন্ট থাকলে আপনি আমাদের ডেবিট কার্ডের জন্য আবেদন করতে পারেন।	Incorrect

Figure 5.15: Question Variations and Responses



sample question: স্টুডেন্ট সেভিংস একাউন্ট ওপেন করা যাবে?

Variation	Response	Correct/Incorrect
স্টুডেন্ট সেভিংস একাউন্ট খুলতে চাচ্ছি	স্টুডেন্ট অ্যাকাউন্ট ওপেন করার সময় ১০০ টাকা চার্জ প্রযোজ্য হবে।	Correct
স্টুডেন্টরা একাউন্ট ওপেন করতে পারবে?	স্টুডেন্ট অ্যাকাউন্ট ওপেন করার সময় ১০০ টাকা চার্জ প্রযোজ্য হবে।	Correct
স্টুডেন্ট একাউন্ট খুলতে পারব?	স্কুল এবং কলেজ শিক্ষার্থীদের জন্য সিটি ব্যাংক নিয়ে এসেছে স্টুডেন্ট সেভিংস অ্যাকাউন্ট। ১৮ বছরের কম বয়সীরা স্টুডেন্ট সেভিংস অ্যাকাউন্ট এর স্কুল প্ল্যান এবং ১৮-২৪ বছর বয়সীরা কলেজ প্ল্যান সেবা নিতে পারবেন।	Incorrect

Figure 5.16: Question Variations and Responses

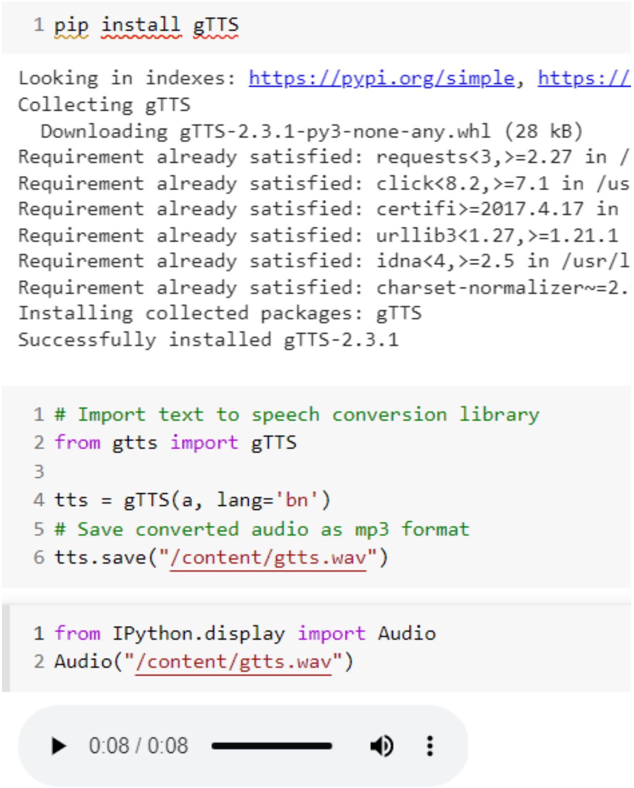
To calculate the accuracy of our system, we randomly choose 10 questions and make at least 3 kinds of variations of that question and observe the responses. We found out - for some questions, the system works really well and generates the proper response. However, for some, it could not generate the correct response. We take notes on the responses and put them into the table below. As per our test, the calculated accuracy of our system is 70.001%. We believe the accuracy may go up if the number of variation questions is more.

Correct Response/Total Variation	Accuracy Percentage
2/3	66.67
3/3	100
3/3	100
2/3	66.67
2/3	66.67
1/3	33.33
0/3	0.00
2/3	66.67
3/3	100
3/3	100

Table 5.2: Predicting the model accuracy

5.4 Text-to-Speech

As for the speech-to-text conversion part, we use the gTTS python library. This library works quite decent with the Bengali language. It saves our time to implement other TTS's like espeak or Festival.



```
pip install gTTS
```

```
Looking in indexes: https://pypi.org/simple, https://
Collecting gTTS
  Downloading gTTS-2.3.1-py3-none-any.whl (28 kB)
Requirement already satisfied: requests<3,>=2.27 in /
Requirement already satisfied: click<8.2,>=7.1 in /us
Requirement already satisfied: certifi>=2017.4.17 in
Requirement already satisfied: urllib3<1.27,>=1.21.1
Requirement already satisfied: idna<4,>=2.5 in /usr/l
Requirement already satisfied: charset-normalizer~=2.
Installing collected packages: gTTS
Successfully installed gTTS-2.3.1
```

```python
# Import text to speech conversion library
from gtts import gTTS

tts = gTTS(a, lang='bn')
# Save converted audio as mp3 format
tts.save("/content/gtts.wav")
```

```python
from IPython.display import Audio
Audio("/content/gtts.wav")
```

Figure 5.17: Implementing Text to Speech

Chapter 6

Conclusion and Future work

The beginning of an automated voice chat system for banking in Bangla, known as Shohojogi, has the potential to fundamentally alter the way banking services are delivered in Bangladesh. Shohojogi can improve the customer experience, expand financial inclusion, and save operational costs for banks by giving customers a user-friendly and accessible system. Shohojogi has the capacity to comprehend client requests and answer with prompt and precise responses. Furthermore, the technology may be simply incorporated into the current financial infrastructure, allowing institutions to provide 24/7 client service without the need for additional personnel. Shohojogi's success depends heavily on ongoing development and improvement, particularly in terms of increasing the precision and effectiveness of its algorithms. Furthermore, banks and developers should both place a high focus on protecting the security and privacy of consumer data. Overall, Shohojogi is an attractive approach to the problems facing the Bangladeshi banking industry. It has the potential to revolutionize the way users access and utilize banking services by making them more effective, convenient, and inclusive with future development and deployment.

Automated voice chat systems in Bangla language have become increasingly popular in recent years, particularly in the banking sector. These systems enable customers to interact with the bank in their native language, which can improve their overall experience. However, despite their benefits, there are several limitations to automated voice chat systems in Bangla language for the banking system. The voice chat system may have a restricted vocabulary, so limiting the user experience. For instance, clients may not be able to communicate with the system using particular financial terminology or technical jargon. Voice recognition accuracy due to the intricacy and diversity of Bangla pronunciation, speech recognition accuracy might be problematic. It may be necessary to educate the system on a wider range of accents and dialects to improve its accuracy. Automatic voice chat systems may have trouble understanding the context of a customer's question, which can lead to misunderstandings and dissatisfaction. To solve this, it may be necessary to train the system in a broader variety of conversational settings and employ more complex natural language processing algorithms. The technology may have limited multi-modal interaction capabilities, limiting the customer experience. To explain their questions, customers may wish to submit screenshots or documents with the system. Automated voice chat services may have an impersonal or robotic tone, resulting in a less engaging customer experience. To solve this, the system can be created with a more engaging personality or more emotional intelligence approaches can be used. When comprehending and interpreting the Bangla language, automated voice chat systems may have accuracy concerns. This might lead to misinterpretation and misconceptions among clients. The automated system may encounter technical difficulties, such as inadequate connectivity or malfunctioning, which may negatively impact the user experience. But so far, there is still considerable room for development, especially in the Bengali language. Several possible areas of future research exist for enhancing the functionality and capacities of automated voice chat systems in the Bangla language for the banking system. Adding more multimodal interaction features, such as image and document sharing, can enhance the consumer experience. The addition of more sophisticated machine learning techniques and larger training datasets can improve the accuracy of speech recognition. To improve context understanding, incorporating more complex natural language processing algorithms can be advantageous. Adding more tailored experiences based on user behavior and past interactions might result in a more engaging customer experience. Creating algorithms and models that can enhance the accuracy of automated voice chat systems' understanding and interpretation of Bangla language.Additionally, there are a number of restrictions to text summarization in Bangla, such as a lack of resources, complex grammar, a restricted vocabulary, and cultural context.Now the system is only partially automated. In the future, we will aim to build a system that is fully automated so that customers have a smooth experience. It took some time for a response to be generated. We will try to generate real time responses in the future. Text-to-speech technology advancements allow for the creation of increasingly natural-sounding automated Bangla voices. Adding extra features and capabilities, such as the ability to handle complex transactions and requests, or providing personalized suggestions based on the customer's banking history. Include feedback mechanisms within the system so that clients can submit feedback on the system's accuracy and efficacy. This feedback can be utilized to enhance the system further. By overcoming these limitations and pursuing these areas of future research, automated Bangla voice chat systems for the banking system can deliver a more engaging and successful customer experience, which will eventually benefit both customers and banks.

Bibliography

[1] C. J. Weinstein, D. B. Paul, and R. P. Lippmann, "Robust speech recognition using hidden markov models: Overview of a research program," 1990.

[2] M. H. Morris and D. L. Davis, "Measuring and managing customer service in industrial firms," *Industrial Marketing Management*, vol. 21, no. 4, pp. 343–353, 1992.

[3] L. J. Hoffman, K. Lawson-Jenkins, and J. Blum, "Trust beyond security: An expanded trust model," *Communications of the ACM*, vol. 49, no. 7, pp. 94–101, 2006.

[4] D. R. Choudhury, P. Bhargava, Reena, and S. Kain, "Use of artificial neural networks for predicting the outcome of cricket tournaments," *International Journal of Sports Science and Engineering*, vol. 1, no. 2, pp. 87–96, 2007, ISSN: 1750-9823.

[5] S. Quarteroni and S. Manandhar, "A chatbot-based interactive question answering system," *Decalog 2007*, vol. 83, 2007.

[6] M. M. Helms and D. T. Mayo, "Assessing poor quality service: Perceptions of customer service representatives," *Managing Service Quality: An International Journal*, vol. 18, no. 6, pp. 610–622, 2008.

[7] C. D. Manning, P. Raghavan, and H. Schütze, "Xml retrieval," *Introduction to Information Retrieval*, 2008.

[8] H. Hassanpour and P. M. Farahabadi, "Using hidden markov models for paper currency recognition," *Expert Systems with Applications*, vol. 36, no. 6, pp. 10 105–10 111, 2009.

[9] M. A. Hossan, S. Memon, and M. A. Gregory, "A novel approach for mfcc feature extraction," in *2010 4th International Conference on Signal Processing and Communication Systems*, IEEE, 2010, pp. 1–5.

[10] Chaldal, "Contactus - chaldal online grocery shopping and delivery in bangladesh: Buy fresh food items, personal care, baby products and more. (n.d.)," Aug. 2013, pp. 973–979.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[12] M. Zhang and Q. Meng, "Automatic citrus canker detection from leaf images captured in field," Feb. 2013. [Online]. Available: shorturl.at/fptP5.

[13] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, PMLR, 2014, pp. 1188–1196.

[14] T. Tulabandhula and C. Rudin, “Tire changes, fresh air, and yellow flags: Challenges in predictive analytics for professional racing,” *Big Data*, vol. 2, no. 2, pp. 97–112, 2014, PMID: 27442303. DOI: 10.1089/big.2014.0018. eprint: https://doi.org/10.1089/big.2014.0018. [Online]. Available: https://doi.org/10.1089/big.2014.0018.

[15] P. A. Angga, W. E. Fachri, A. Elevanita, R. D. Agushinta, *et al.*, “Design of chatbot with 3d avatar, voice interface, and facial expression,” in *2015 international conference on science in information technology (ICSITech)*, IEEE, 2015, pp. 326–330.

[16] S. Abujar, M. Hasan, M. Shahin, and S. A. Hossain, “A heuristic approach of text summarization for bengali documentation,” in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, 2017, pp. 1–8.

[17] P. Prasad, *Crafting qualitative research: Beyond positivist traditions.* Taylor & Francis, 2017.

[18] J. Yushendri, A. R. Hanif, A. A. P. Siswadi, P. Musa, T. M. Kusuma, and E. P. Wibowo, “A speech intelligence conversation bot for interactive media information,” in *2017 second international conference on informatics and computing (ICIC)*, IEEE, 2017, pp. 1–6.

[19] R. Burri, *Improving user trust towards conversational chatbot interfaces with voice output*, 2018.

[20] A. R. Pal, D. Saha, N. S. Dash, and A. Pal, “Word sense disambiguation in bangla language using supervised methodology with necessary modifications,” *Journal of The Institution of Engineers (India): Series B*, vol. 99, pp. 519–526, 2018.

[21] I. J. Bristy, N. I. Shakil, T. Musavee, and A. R. Choton, “Bangla speech to text conversion using cmu sphinx,” Ph.D. dissertation, Brac University, 2019.

[22] H. A. Chowdhury, M. A. H. Imon, A. Rahman, A. Khatun, and M. S. Islam, “A continuous space neural language model for bengali language,” in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, IEEE, 2019, pp. 1–6.

[23] M. Ewing, L. R. Men, and J. O’Neil, “Using social media to engage employees: Insights from internal communication managers,” *International Journal of Strategic Communication*, vol. 13, no. 2, pp. 110–132, 2019.

[24] M. M. Hasan, M. A. Islam, S. Kibria, and M. S. Rahman, “Towards lexicon-free bangla automatic speech recognition system,” in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, IEEE, 2019, pp. 1–6.

[25] S. M. Islam, M. F. A. Houya, S. M. Islam, S. Islam, and N. Hossain, “Adheetee: A comprehensive bangla virtual assistant,” in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, IEEE, 2019, pp. 1–6.

[26] M. M. Joshi, S. Agarwal, S. Shaikh, and P. Pitale, “Text to speech synthesis for hindi language using festival framework,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 04, pp. 630–632, 2019.

[27] M. Kowsher, M. M. Rahman, S. S. Ahmed, and N. J. Prottasha, “Bangla intelligence question answering system based on mathematics and statistics,” in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, IEEE, 2019, pp. 1–6.

[28] C. Lin, T. Miller, D. Dligach, S. Bethard, and G. Savova, “A bert-based universal model for both within-and cross-sentence clinical temporal relation extraction,” in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, 2019, pp. 65–71.

[29] M. M. Rahman, R. Amin, M. N. K. Liton, and N. Hossain, “Disha: An implementation of machine learning based bangla healthcare chatbot,” in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, IEEE, 2019, pp. 1–6.

[30] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, “Supervised contrastive learning for pre-trained language model fine-tuning,” *arXiv preprint arXiv:2011.01403*, 2020.

[31] “How to measure the success of your voice ai bot?,” vol. 28, Feb. 2020.

[32] P. Klaus and J. Zaichkowsky, “Ai voice bots: A services marketing research agenda,” *Journal of Services Marketing*, vol. 34, no. 3, pp. 389–398, 2020.

[33] K. Saakshara, K. Pranathi, R. Gomathi, A. Sivasangari, P. Ajitha, and T. Anandhi, “Speaker recognition system using gaussian mixture model,” in *2020 International Conference on Communication and Signal Processing (ICCSP)*, IEEE, 2020, pp. 1041–1044.

[34] I. Samuel, F. A. Ogunkeye, A. Olajube, and A. Awelewa, “Development of a voice chatbot for payment using amazon lex service with eyowo as the payment platform,” in *2020 International Conference on Decision Aid Sciences and Application (DASA)*, IEEE, 2020, pp. 104–108.

[35] V. Verma and R. K. Aggarwal, “A comparative analysis of similarity measures akin to the jaccard index in collaborative recommendations: Empirical and theoretical perspective,” *Social Network Analysis and Mining*, vol. 10, pp. 1–16, 2020.

[36] “What is a voicebot and why your company might want one?,” vol. 28, Feb. 2020.

[37] P. Bhattacharjee, A. Mallick, and M. Saiful Islam, “Bengali abstractive news summarization (bans): A neural attention approach,” in *Proceedings of International Conference on Trends in Computational and Cognitive Engineering: Proceedings of TCCE 2020*, Springer, 2021, pp. 41–51.

[38] N. R. Bhowmik, M. Arifuzzaman, M. R. H. Mondal, and M. Islam, “Bangla text sentiment analysis using supervised machine learning with extended lexicon dictionary,” *Natural Language Processing Research*, vol. 1, no. 3-4, pp. 34–45, 2021.

[39] M. M. Hasan, A. Roy, and M. T. Hasan, “Alapi: An automated voice chat system in bangla language,” in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, IEEE, 2021, pp. 1–4.

[40] T. Hasan, A. Bhattacharjee, M. S. Islam, *et al.*, “Xl-sum: Large-scale multilingual abstractive summarization for 44 languages,” *arXiv preprint arXiv:2106.13822*, 2021.

[41] S. I. Pranto, R. A. Nabid, A. M. Samin, *et al.*, “Aims talk: Intelligent call center support in bangla language with speaker authentication,” in *2021 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, IEEE, 2021, pp. 1–6.

[42] S. Guchhait, A. S. A. Hans, and J. Augustine, “Automatic speech recognition of bengali using kaldi,” in *Proceedings of Second International Conference on Sustainable Expert Systems: ICSES 2021*, Springer, 2022, pp. 153–166.

[43] M. Rakib, M. Hossain, N. Mohammed, F. Rahman, *et al.*, “Bangla-wave: Improving bangla automatic speech recognition utilizing n-gram language models,” *arXiv preprint arXiv:2209.12650*, 2022.

[44] T. T. Showrav, “An automatic speech recognition system for bengali language based on wav2vec2 and transfer learning,” *arXiv preprint arXiv:2209.08119*, 2022.

[45] J. (Gallemard, *Voice bots are the future of customer service*, Year Published. [Online]. Available: https://blog.smart-tribune.com/en/.