# A Silver-tongued Writer: A Deep Learning and NLP based Bangla Sentence Composer with Proper Linguistic Affection

by

Arko Mazhar
23141091

Muhammad Rafid Zaman
18201115

Md. Nazrul Islam
19301124

Sumaiya Mehjabeen
19101116

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

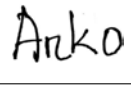
Department of Computer Science and Engineering
School of Data and Sciences
Brac University
September 25, 2023

# 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.

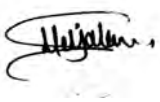
**Student's Full Name & Signature:**



Arko Mazhar
23141091

Muhammad Rafid Zaman
18201115

Md. Nazrul Islam
19301124



Sumaiya Mehjabeen
19101116

# Approval

The thesis/project titled "A Silver-tongued Writer: A Deep Learning and NLP based Bangla Sentence Composer with Proper Linguistic Affection" submitted by

1. Arko Mazhar (23141091)
2. Muhammad Rafid Zaman (18201115)
3. Md. Nazrul Islam (19301124)
4. Sumaiya Mehjabeen (19101116)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 25, 2023.

**Examining Committee:**

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam, PhD
Professor
Department Of Computer Science and Engineering
Brac University

Co-supervisor:
(Member)

Md Tanzim Reza
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam, PhD
Professor
Department Of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

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

# Abstract

Communication is one of the most essential parts of human life. With the advent of the 21st century, the usage of online interaction has expanded significantly. Text-based communication is prevalent in many aspects of our lives. For example, social media, email, and business invitations. It has also allowed us to maintain relationships with others, irrespective of time or geographic location. While communicating, people typically overlook the text-style representation. This establishes a barrier between the sender and receiver, preventing the message from being correctly interpreted by the receiver. Due to a lack of proper selection of words, one might face challenges in getting the expected reply from the receiver. In addition, erroneous interpretations of these writings might have detrimental psychological impacts on individuals. The fundamental objective of this paper is to develop a Bangla sentence composer model capable of detecting impolite statements in written text and transferring the style to politeness with proper affection. We begin by discussing several classification approaches. For this purpose, we used a combination of deep learning and transformer-based techniques to identify linguistic politeness from Bangla sentences. Simple RNN, CNN BiLSTM, LSTM, GRU, and various pre-trained versions of BERT were the algorithms we employed. Among all models, BERT produced much superior outcomes than the others. BanglaBERT Showed the best result with an accuracy of 84%. Also, we constructed our own dictionary to replace the impolite and harsh words. After correctly recognizing the expression, we intend to replace harsh  impolite words from input sentences using our custom-made dictionary. The dictionary consists of 410 impolite, harsh words and their corresponding polite version. After that, the result is modified using the BanglaT5 BanglaParaphrase, a Bangla text-to-text generation model. With the aid of this combined deep learning approach, we attempt to identify the optimum sentence variant that does not negatively affect human emotion. Using these techniques, we can achieve preferable results that will earn the recipient's appreciation through our preferred sentence composer approach.

**Keywords:** Politeness; Impolite; Style transfer; Writing; Bangla sentence composer; Paraphrase; Linguistic politeness detection; Simple RNN; NLP; LSTM; CNN BiLSTM; BERT; BanglaT5; Bangla Paraphrase; Dictionary; Text-To-Text generation.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Communication is the process of exchanging information and thoughts with one another. While communicating with someone, we can express our thoughts and feelings. Our expressions can be presented in many different ways such as sentimental, emotional, polite, impolite, etc. When we compose and verbalize a sentence, it carries significance in the receiver's mind. The structure of sentences can determine how our message will be perceived. Depending on the context, a sentence might have multiple meanings. In this digital age, everyone is dependent on online text-based communication. However, everyone does not pay heed while composing their messages. A study from NAEP shows that only 27% of the 12th-grade students are proficient in writing[1]. For this reason, when someone expresses their opinion through social media, email, or text messages, it might sound rude, harsh, or impolite to the receiver. Our goal is to classify and make the sentence appropriate for the receiver with politeness.

According to a study conducted by Avast Foundation[2], about 39% of the participants have the potential to comment on something with rude, harsh, and abusive sentences. Due to such practice, the human mind faces adverse effects. This can hamper a person both mentally and physically. As people do not have to face direct consequences, they tend to use these comments. A report of Scientific American[3] shows that, In a 2002 study of 2,013 persons, 88% said they encounter impolite and disrespectful people every day. These encounters can leave stress on a person's mind which might lead to fear, rage, embarrassment, bewilderment, anxiety, loneliness, self-doubt, despair, sleeplessness, weariness, vomiting, etc

The above-stated issue might not be addressed properly in our society. Our goal is to look at this issue from a problem-solving perspective. Some study reveals that paraphrased sentences should be utilized to preserve the same emotions with a change in sentence form; other research shows that we can classify phrases by altering the words with appropriate affections. Changing the sentiments of a sentence alters the words of the sentence. In this paper, we change sentences to new sentences with appropriate affection keeping the meaning preserved. For this, one must classify sentences based on some categories and change the sentences to a new sentence which will be more impactful to win over readers' minds.

With the advancement of technology, communication through online mediums such as e-mail, social media, discussion forums, and blogs has been increasing. The advent of the internet has made communication not only simpler and more expedient, but it has also made it possible for us to maintain relationships with others regardless of the passage of time or physical distance. We aim to make this communication effective by ensuring politeness through the sentences.

## 1.1 Problem Statement

Integrating a new language into the digital realm poses a formidable challenge. In order to assist individuals in composing written content with accuracy, software developers have incorporated several features such as auto-fill, spell-check, and grammar-check functionalities specifically designed for the English language. Moderation technologies have been implemented to assess and identify undesirable comments made by users. Algorithms employed in social media platforms possess the capability to effectively moderate their extensive user base, hence obviating the necessity for personal intervention. However, it should be noted that this assertion may not hold true for every language. Despite being one of the most frequently spoken languages in the world, Bangla is one of them [4]. One of the primary challenges that social media platforms have is the management of significant issues. In order to demonstrate this, we conducted an analysis on a dataset [5] comprising a substantial volume of offensive language that remained visible on the Facebook platform due to inadequate moderation measures.

In addition to the task of content moderation, there exists a requirement for individuals to engage in formal writing in various aspects of their daily lives, including but not limited to composing letters, applications, and emails [6]. The task of composing these messages while upholding linguistic decency can present certain difficulties. Unintentionally compiling unpleasant content can not only create a negative image but also diminish the likelihood of receiving respectful treatment in the future[7]. The prevalence of this issue is particularly pronounced among male students [8]. To minimize such a consequence and cultivate a positive initial impression, we must participate in comprehensive research and develop a deep understanding of various writing approaches through the examination of existing literature. However, it is crucial to recognize that not all persons have the advantage of being able to allocate time for preparation and continuously store formulas in their memory. According to a study conducted by [9], the findings indicate that a mere 1% of 12th-grade students possess proficient elaborative persuasive writing skills.

Both of these issues can be resolved through the implementation of an artificial detection system for unpleasant words and phrases, which can then provide users with polite alternatives. This technique has the potential to enhance the efficacy of auto moderation technologies for Bangla, which is ranked as the 7th most widely spoken language in the world.

In the course of this study, our objective was to develop a system capable of enhancing the linguistic politeness of Bangla sentences.

## 1.2 Research Objective

Politeness plays a vital role in social interaction and is believed to facilitate communication in human contact by reducing the probability of conflict and confrontation[10]. It is closely related to power dynamics and social distance between conversational participants. Additionally, it is essential to utilize the appropriate level of gentility to get your message delivered and get the response you need. When you speak with courtesy and a pleasant attitude, your message will likely be received without resistance or rejection. However, while positive words can improve one's self-esteem and self-image, unpleasant words and impolite expressions can have long-lasting negative impacts. "Negative words might have a greater influence than we may realize"[11]. For instance, a manager's negative attitude may lead to poor workplace communication or a stranger's rude words may reinforce a person's low self-perception. Therefore, eliminating negative words from one's lexicon might create a chain reaction of positivity. With these harsh words to avoid in writing, we must exclude them from our academic and professional writing as well. In this regard, we have to avoid the impact of negative and impolite words by using positive alternatives. An affirmative phrase can help you sound more neutral, allowing you to convey any argument without offending anyone.[12] We aim to detect the expression of a text and if it contains any impolite or harsh expression, we rephrase it into a polite sentence without changing its meaning. This allows people to express themselves politely and more acceptably, regardless of the expression contained in the sentence. We haven't seen any direct research on the problem concept and its solution, which has made us zealous to go ahead with the research. Thus, our objectives of the research are:

- To effectively detect the linguistic impolite expressions from a large-scale Bangla dataset by deep learning as well as transformer-based techniques such as Bidirectional Encoder Representations from Transformers (BERT) and neural networks.
- To get the highest outcome, we train the models with training data along with different preprocessing techniques in Natural Language Processing (NLP).
- To replace impolite words with their polite counterparts, a dictionary is created from scratch.
- To modify sentences for optimal politeness conversion while preserving meaning, a BanglaT5 BanglaParaphrase text-to-text generation model is implemented.
- To provide a sentence composer approach that generates a more suitable sentence with proper linguistic affection without changing the original meaning.

The remaining paper is organized as follows: In Chapter 2, the literature review gives an insight into the previous research related to our topic. Chapter 3 describes the methodology or procedure used to achieve the purpose of this research, including all the implemented models. In Chapter 4, the result is presented and discussed. Lastly, Chapter 5 contains both the conclusion of the paper and a demonstration of future endeavors.

# Chapter 2

# Literature Review

To gather knowledge on this unfamiliar field we went through a few research papers to see how others tackled this issue. However, there was not much direct research done that deals with politeness style transfer of sentences. So, we studied papers that deal with classifiers that can separate appropriate vs. inappropriate sentences and papers that deal with other types of style transfer or paraphrasing. Here, we briefly explained our findings.

## 2.1 Detection of Impoliteness

As politeness detection is a similar concept to sentiment analysis, we found the first paper [13] where the authors had used a dataset consisting of 22 fairy tales data to predict emotions from text and text-to-speech synthesis as the primary agenda. Here, 90% train, 10% test data was utilized for 10-fold cross-validation with SNoW architecture. The dataset consisted of 1580 sentences whereas Neutral sentences occurred at about 59.94% from 7 types of categories. Two tuning methods which are i).sep-tune-eval and ii)same-tune-eval were applied due to the availability of limited data. However, the issues were the Dataset being too small, data can't be distinguishable with ease and the original annotated label was not combining with the class of Emotion.

So, we turned to paper[14] where dataset size was not an issue. The authors used blog data, which transfers emotions more directly than other types of data like personal writing, diaries, or emails, to classify emotions. As the annotation agreement differs from judge to judge Cohen's Kappa method was used on the categories to get the mutual exclusiveness consensus. According to the method, agreement was highest found on happiness and fear with an average of 0.77 and 0.79. Later on, they used the MASI technique to measure agreement. For the experiment, they implemented Support Vector Machines(SVM) and Naïve Bayes and got an accuracy of 73.89% and 72.08% respectively. Finally, they illustrated the limitation, and how context and semantics are required to classify emotion.

However, the accuracy was lower than another group of researchers [15]. Throughout their work, they tried to find out if being an admin of Wikipedia or Stack Exchange makes you more rude to others. Since politeness is a subjective topic, they used Z-score normalization to deal with subjectivity. Later, they used two classifiers on the

labeled data, Bag of Words(BOW) and a linguistically informed classifier (Ling.). They tried a cross-domain train test to see which works better but found out the most accuracy comes from when both train test data were taken from wiki and classified by Ling. which was 83.79%. Here, the accuracy is high for a machine to predict emotion but we found a paper [16] that improved it even more. The authors propose a model of the connections between the phrases and word sequences in each sentence, using a composite CNN-RNN framework. They applied the "Gated Representation Alignment" (GRA) model for sentence categorization using CNN and RNN. In this model, they tested their model on datasets containing both clean and noisy texts, they obtained the clean data set from Stanford Sentiment Treebank (SST5), and the noisy dataset from Yelp. The accuracy of GRA over SST5 is 51%, over SST2 87.9%, and Yelp 58.1%. So, in our search for the best work done on sentiment analysis, we found that SST2 has the best accuracy in terms of sentiment analysis. However, they had a robust dataset to work with.

In the first paper [17], the authors tried to create a polite transformer that takes a non-polite sentence and transfers it to a polite one. They used Nui and Bansal Classifiers to give sentences a politeness score. Sentences above 90% politeness were labeled as p9 and sentences below politeness score were labeled P0. Then they extracted the top 10 most used words of both P0 and P9 and found out the words that are responsible for making sentences polite or nonpolite. Their function added a tag to include extra phrases or replaced the nonpolite words with a tag. Later the tag gets replaced with a polite word. They applied various methods to do this but Style transfer from non-parallel text by cross alignment (CAE) claimed to yield the highest accuracy of 99.62 percent which will make any researcher suspicious of their way of defining accuracy.

In the spirit of simple changes, next research [18], authors found that many people face problems with reading long sentences and the author tried to solve the problem with controllable sentence simplification. By this, they have made sentences easy to read by expanding and excluding words. This research has established that according to the WikiLarge test, the state of the art is 41.87 SARI. Here, they used sequence to sequence model to multiple NLP, by conditioning on relevant attributes to the parametrization mechanism that is tailored to the particular task. They train transformer models using the Fairseq toolkit. For training and evaluation datasets, they used the WikiLarge dataset. They used FKGL and SARI methods for simplicity and evaluated the overall score. Moreover, for sentence simplification, they used the EASSE Python package by computing FKGL and SARI. Their Wiki-Large testing dataset includes SARI and FKGL. There is a 95% statistic around each score, which is the average of 10 runs. On the other hand, since BLEU was inappropriate for evaluation, they did not apply it. Also, the writer employs control tokens that are measurable given the source and shortened sentences. The main effect of tokens, DepTreeDepth, LevSim, and Wor-dRank in this study was to shorten the text. They also added the NbChars control token during training and set its ratio to 1.00 to clearly study the influence of those control tokens. Finally, this paper used tokens to generate simplified sentences and make people read any sentence easily.

However, language is always changing, evolving, and adapting to newer generations.

Having a solid refined dataset is easier to work with but can lose relevance over time. So, we looked into a paper [19], where they used an unsupervised model. The author here tried to transfer text style which is a subtask of paraphrasing. They want to change a sentence but without changing its meaning after conversion. As the existing method was not able to transfer the sentence completely, that's why they tried to shorten the problem by this process. The author used NLP. Also, they used the seq2seq model and RNN encoder system, and they adopted the auto-encoder deep neural network model with long short-term memory(LSTM). Moreover, for training and testing data they used Shakespeare's plays. Here, they tried to translate Shakespeare's text into modern English, keeping the overall semantic content. They divided 33,726 sentences in both styles for training, 4,216 for testing, and 4,216 for validation. To auto-paraphrase the sentence they used BLEU score. On the other hand, they did not use parallel texts for training as this remained unchanged and the traditional evaluation metrics like BLEU scores are meaningless therefore this may need to be improved further. In the end, their contribution was to suggest that style transfer can be conducted at a higher level, such as the author's individualized writing style not only at the lexical, phrasal, or syntactic levels.

## 2.2 Addressing the Issue of Impoliteness

In the next paper [20], the authors tried to paraphrase any sentence to reduce plagiarism by changing the sentence but without changing its semantics after conversion. The author used NLP and it is performed in five steps and they are lexical analysis, syntax analysis, semantic analysis, discourse integration, and pragmatic analysis. They have done this in various systems such as text segmentation, tokenization, part-of-speech tagging, classification of sentences, database generation, and refarming rules. For tokenization, Stanford POS tagger is used, also, to tag words english-left3words-distsim tagger model is used. Even when applying refrain rules, a regex pattern matcher is used to match the pattern of the sentence. Moreover, it can be used in making robots understand difficult sentences, or even develop an intelligent system to make decisions like humans. To conclude, this paper helps to paraphrase English sentence structure to remove plagiarism without changing the meaning of the actual sentence. However, very little of it helps change the tone of sentences but helpful nonetheless.

In the next paper [21], the purpose was primarily to acknowledge the general-purpose, domain-independent embedding of sentences based on supervision from a database of paraphrases to create universal sentence embeddings. Additionally, to comprehend the comparability and outcomes of three supervised NLP architectural tasks: sentence similarity, sentiment classification, and entailment. The authors go through compositional models of many functional architectures based on different categories including simple addition operations, and richly-structured functions like RNN, CNN, and LSTM. Six neural network-based compositional models capable of encoding arbitrary word sequences with a high cosine similarity into a vector were presented in the paper. It also can efficiently transfer into domains including in-domain and out-of-domain tasks. The authors propose experimenting with 24 textual similarity datasets across numerous domains as a data collection. They are as follows: All datasets from every SemEval semantic textual similarity (STS) task

(2012- 2015), SemEval 2015 Twitter task, SemEval 2014 Semantic Relatedness task, and two tasks that used PPDB data. As training data, they used the XL section of the PPDB, which had 3,033,753 unique phrase pairs, and for hyperparameter tuning, they used 100k PPDB XXL sample instances. They have observed seven table findings. Their approach can strengthen entailment models when used as a precondition as well as generic text similarity, and even when employed as a fixed representation in a classifier, it can achieve strong performance. The authors also provide a future checklist for upgrading embeddings. They proposed pursuing this by developing new models and efficiently handling undertrained words.

In this Paper[22], The authors have implemented a new benchmark for the Natural Language Generation Task for the Bangla language. Although Bangla is a language that is spoken by a large number of people, there are only a small number of resources available for it, which presents a challenge for individuals who work in the field of NLP. In order to accomplish this, the authors aim to enrich the NLP field for the Bangla language by pretraining BanglaT5. The researchers utilized a pure Bangla text corpus of 27.5 GB that is effective for tasks such as machine translation, text2text generation, and many others. The dataset contains a total of 5.25 million documents which is perfect according to the authors for this task. Encoder-decoder-only models were chosen for pretraining the models for generative tasks. For the model architecture, the T5 base model was used with "12 layers, 12 attention heads, 768 hidden sizes, 2048 feed-forward size using GeGLU activation (Shazeer, 2020), and a batch size of 65536 tokens for 3 million steps on a v3-8 TPU instance on GCP. This was accomplished with a total of 65536 tokens." The authors have compared 4 state of the art multilingual models for their comparison purpose which are mT5, mBART50, XLM-ProphetNet, and IndicBART. Their custom model showed 4-9.54% better results than the above models. The authors also mention some limitations of their model. The dataset might have some toxic-based statements which can lead the model to show unintended results which might be problematic for real-world deployment. Lastly, they conclude by expressing their interest in training the introducing new tasks for conversational inquiry responding and the creation of individualized communication.

In this paper [23], discussed metaphoric sentence generation. They wanted to generate a symmetrically similar sentence from a Metaphoric sentence as Metaphoric sentences are often difficult to understand for human beings. This paper contains mainly two methods and those are free generation and controlled generation. The authors collected two datasets, one dataset was from another paper and another one is from the MNS corpus. Here, the first dataset was taken from another paper that works in replacing metaphoric words with literal paraphrases, and the second one, the MNS corpus was used by mapping and using FrameNet annotation. The dataset was structured as a collection of input sentences (I), vocabulary terms (S), and target vocabulary (T), which were then mapped accordingly. After that, they applied different kinds of methods and those are the seq2seq model, SOW-REAP Paraphrase Generation, Free Generation, and Controlled Generation. The researchers used the T5 system for both the free and controlled generation models in the seq2seq model. After that, the free generation method generates without knowing what could be the interesting and valuable output for the given input. So, the context was altered

with the use of a control code, namely "Activate metaphors". Then, the controlled method's primary objective is to construct paraphrases that include explicit encoding of the semantics of the intended metaphor. Moreover, a test set consisting of 250 pairs of composite metaphors was generated by the researchers. In the SOW-REAP model, they got a human evaluation score of 3.176 which is worse than other models. However, the free models performed better in fluency and the All model got 3.511 in fluency. Also, it's better in sentence fluency and the score is 3.680. For metaphoricity, the control model got the best score. In the end, this study demonstrates that the incorporation of control in the process of metaphor production via the use of conceptual domains enhances the metaphoricity of paraphrases produced. However, it is observed that this approach may lead to a reduction in fluency and semantic similarity. Also, they explored automated evaluation metrics.

This article [24], presented the Chatbot interaction with Artificial Intelligence framework trains a transformer-based chatbot-like architecture for task classification using natural human interaction rather than code, interfaces, or formal instructions. The authors worked on 13,090 labeled data for their transformer-based paraphrase models. They used 70% of the data for training purposes and 30% for testing purposes. For text-to-text commands, the researchers used state-of-the-art transformer-based categorization methodologies. In this research, they used different types of models and those are BERT, DistilBERT, DistilRoBERTa, RoBERTa, XLM, XLM-RoBERTa, XLNet, and T5. The T5 paraphrase model underwent training using the Quora question dataset. In these models, they got the accuracy for BERT, DistilBERT, DistilRoBERTa, RoBERTa, XLM, XLM-RoBERTa, and XLNet respectively 98.55%, 98.34%, 98.55%, 98.96%, 14.81%, 98.76%, 35.68%. As XLM and XLNet models gave very poor accuracy, they omitted these two models. After training, they got the best result for the RoBERTa model, and the classification accuracy is 98.96%. Finally, an accuracy of 99.59% was achieved in the output prediction process by using either a Logistic Regression model or a Random Forest model. In the future, metrics like accuracy, precision, recall, and F1 Score might potentially provide more accurate results by using a K-fold cross-validation model, as shown by the findings of this research study.

Since natural language processing(NLP) gets all the attention, there is very little out there to benchmark natural language generation(NLG) for Bangla, especially low-resource NLGs. So creators of BanglaNLG[25] prepared a benchmark tool to score various NLG tasks such as machine translation, text summarization, question answering, dialogue generation, news headline generation based on this criterion: diversity of data size, difficulty of accomplishing the model, accessibility of the model and dataset, presence of self-evaluation matrices. But they did not stop at a benchmark tool. To prove the capability of their tool, they created BanglaT5, a sequence-to-sequence transformer model. To train the model, they used 4.8 million data points with an average sequence length of 402.32 tokens. BanglaT5 performed better than various multilingual models in text summarization(TS), question answering(QA), and news headline generation(NHG) in Bangla. In TS, BanglaT5 scored 13.7 whereas second best scorer was mBART-50 with 10.4. On average, the tested models scored 10.3. This means the specialized model for Bangla outperformed the multilingual models. Similarly for QA, BanglaT5 scored 68.5/74.8

which is 8.9/9.2 points better than the second best model. They also scored 2.9 points higher than the second-best model in terms of NHG tasks. That is why we use BanglaT5 for tokenization and seq-to-seq transformation. However, the authors were not satisfied with the results as it can be improved by filtering the dataset further so the model does not generate biased or toxic Bangla texts. They also want to add more tasks to score in the future to make it a benchmark tool for everything Bangla-related.

In the paper titled "Fine-Grained Emotional Paraphrasing along Emotion Gradients" [26] the author introduced a task consisting of modifying the emotional potencies of the paraphrases in fine grain while preserving the meanings of the originals. They proposed a framework for fine-grained emotional rephrasing through emotion gradients, that is, by fine-tuning text-to-text Transformers using multi-task training that includes Emotion-Labeling of Training Datasets, Fine-Tuning Text-to-text Transformer Models, and Paraphrasing with Emotional Transition Graph acknowledged by GoEmotions (Demszky et al., 2020). Microsoft Research Paraphrase Corpus, Quora Questions Pairs, Google PAWS-Wiki, and Twitter Language-Net paraphrasing corpus were used to fine-tune the dataset. They labeled the dataset and fine-tuned the T5 model using numerous transformer models, including BART. Various matrices, including ROUGE-1, ROUGE-2, ROUGE-L, BLEU, Google BLEU, and METEOR, are used to evaluate the models' ability to perform paraphrasing. Through the resulting chart of fine-tuning, it is perceived that the models trained using emotion-labeled paraphrase datasets outperformed the control t5-base model and improved the emotional paraphrasing ability for each dataset and matrix. Another significant observation is that fine-tuning models perform better with limited data than T5 models. The performance of emotion transitions can be enhanced more than in the absence of constraints. In future research, the author will place a greater emphasis on emotion classification, which plays a crucial role in emotion paraphrasing, as well as on datasets.

In this paper [27], the author transforms a contrived dataset into the finest Bangla dataset by ensuring diversity and semantics through an automated filtering pipeline, with the goal of easing the low resource crisis for the Bangla language in the NLP domain and enhancing other Bangla datasets. Using the state-of-the-art transition model, Bangla data collected from the RoarBangla website is translated into English for the dataset. Initially, a fabricated Bangla Paraphrase model is introduced, ensuring diversity through the PINC Score Filter and semantics through a multilingual BERT model, BERTScore, as well as a monolingual BERT model, BanglaBERT. The author then introduced the N-gram Repetition Filter in order to modify the irrelevantly long-growing paraphrases that had duplicate portions due to training the first two filters. A Punctuation Filter is utilized to guarantee the dataset is noise-free. The author used a variety of metrics, including BLEU, sacreBLEU, METEOR, and multilingual ROUGE-L, to evaluate a variety of criteria; also the dataset is further augmented. After fine-tuning BanglaBERT with token classification and mT5-small, the tokens are masked with XLM-RoBERTa. The author divided the entire dataset into 80:10:10 proportions for training, validation, and testing purposes, accordingly. The dataset is fine-tuned (Hyperparameter Tuning) for comparative analysis with IndicBARTSS and IndicBART. By training

BanglaBERT for 20 iterations on a Dataset containing 30 POS Taggers, 92% accuracy was achieved. Then, seven parts of speech are carefully identified, and tagged. As a result, the trained model consisting of BanglaT5, BanglaT5-aug, mT5-small, mT5-small-aug11, IndicBART, and IndicBARTSS on BanglaParaphase performs exceptionally well for semantic preservation and ensures a high level of diversity. In conclusion, they asserted that their decision to use a large mono-lingual corpus is sufficient for peripheral paraphrasing tasks.

Lastly, a research paper took the help of reinforcement learning where the authors describe a deep reinforcement learning strategy to paraphrase generation in their work [28]. They propose a new data-driven framework, called RbM (Reinforced by Matching) that consists of two modules, a generator and an evaluator. The generator may generate a paraphrase of a given text, while the evaluator can determine the semantic similarity between two sentences. The generator is a sequence-to-sequence learning model with an attention and copy mechanism, where,s the evaluator is a deep matching model or a decomposable attention model. As data sets, they employ the Quora question pair dataset and the Twitter URL paraphrasing corpus. In addition, they perform both automated and human assessments. They initially trained the generator with cross-entropy loss and then fine-tuned it with policy gradient and evaluator supervision as rewards. In the implementation of a recurrent neural network (RRN) on the generator, they consider the pointer generator and construct the model using an encoder-decoder architecture. In addition, the evaluator was trained using both positive and negative data instances. When both instances are provided as training data, Supervised Learning (SL) is utilized, but Inverse Reinforcement Learning (IRL) is used when only positive training data is available. Moreover, the author presented several training approaches that they said were essential and effective for achieving the highest outcome. Furthermore, in order to compare the RbM-SL and RbM-IRL approaches with the existing neural network-based methods, the authors chose five baselines for both manual and automatic evaluation. In the results section for Quora datasets, the RbM-SL consistently outperforms automatic measures, however, with additional evaluator assessment, the RbM-SL achieves an accuracy of 87% in distinguishing positive and negative paraphrases. The authors claim that RbM-IRL performs better on the Twitter corpus than RbM-SL, which performs best with curriculum learning. The authors further demonstrate that the models have higher relevance and fluency scores. In manual evaluation, RbM-IRL achieves the best fluency score whereas RbM-SL achieves the best relevance score.

To sum up, all these recent and previous researches helped us to find the pros and cons between different algorithms and concepts that can be used to solve this problem. From all these, we finally selected traditional deep learning models and transformer-based BERT models for our classification task. Also, we formed the overall approaches that can be used to change the sentence tone. This will allow us to effectively transform sentences and reform them in a better-structured way. This concludes our quest for finding various techniques on text style transfer which will be helpful to build a tool that can effectively classify the linguistic impoliteness & convert impolite sentences to polite ones.

# Chapter 3

# Methodology

The entire procedure (the overview diagram, description, and preprocessing of the dataset and the models of deep learning that are implemented) is elaborately discussed in this section. An overview diagram of the entire research is given below:

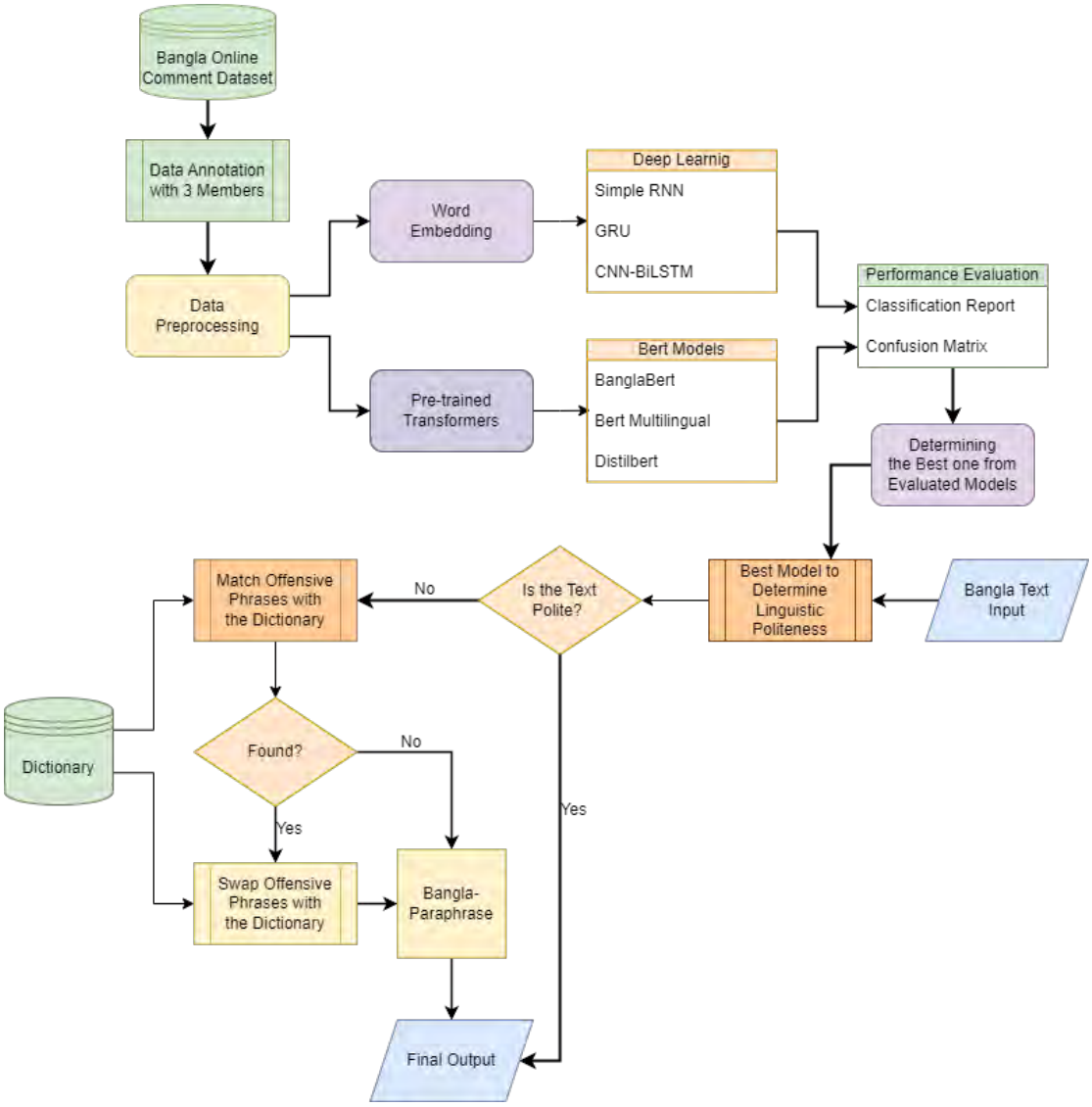



Figure 3.1: The Top-Level Overview of the Proposed Method

## 3.1 Dataset Description

The data used in this study was collected from a recently published research paper containing 44001 Bangla language comments, including Facebook comments. It was determined that the dataset was pertinent to our investigation. In this "Dataset description" section, the collection, annotation, final labeling, and preprocessing of datasets are elaborated upon.

### 3.1.1 Data Collection

A Bangla language dataset was required in order to complete this challenge because the focus of this article is on conducting classification work on phrases based on the level of politeness they include. Due to the small amount of study that has been done on the Bangla language compared to English, we were unable to collect a dataset that was based on politeness categorization tasks. For this reason, we collected the dataset from a recently published research paper. After reading the study [5], we discovered that the dataset was determined to be comparable to our investigation, which includes around 44001 comments and their corresponding attributes such as category, gender, and comment reaction number. The dataset also included comments from Facebook that were categorized according to whether or not they were considered to be acts of cyberbullying. These categories were Non-bully, Sexual, Threat, Troll, and Religious. As the label categories may be regarded as useful for our work, we decided to conduct our study using this particular dataset. Since our research is focused on the politeness and impoliteness of various words and sentences, the three aspects that were discussed earlier weren't relevant to our work and, as a result, we did not require their consideration. Therefore, we were only interested in the comments data because of this reason, and we did not take into account any of the other aspects. We need labels that are based on whether a statement is polite or not so that we may complete the politeness categorization portion of our research. So, we labeled the comments into two classes and those are i) Polite & ii) Impolite.

For our research, we took 12300 Comments from the Dataset. In Figure Table 3.1 below, there are lots of Bengali comments in the dataset from which the sample is shown in the figure, some of them are polite and some of them are impolite sentences.

Table 3.1: Data Sample

| |
|---|
| তাকে জিজ্ঞাসা করেন ' আসিফ মহিউদ্দিন' এর অর্থ কি? সে বলল তার কো ন সৃষ্টিকর্তা নাই। |
| তাহলে এই নাম ধারন করে আছে কেন? এটা তার ভন্ডামি নয় ? |
| করোনা যুদ্ধের প্রথম শহীদ ! |
| বিশ্বকাপ জয়ের মাধ্যমেই এই বৃত্ত পূর্ণ হবে ইনশাআল্লাহ ! |
| ওর চোখের মনিতে গোল গোল ঐটা কি? কন্ট্রাক লেন্স? |
| মাশআল্লাহ |
| সে ভোট ডাকাত সরকারের এজেন্ট |
| তুই বুঝস, পরকাল কি? আগে পড়াশুনা কর, তারপর পান্ডিত্য ফলা! শালা, অশিক্ষিত! |
| শুভ কামনা নিরন্তর। |

### 3.1.2 Data Annotation

In order to construct a model, we will need to provide the model with data that is appropriately labeled. We have adhered to specific standards in order to determine whether or not a comment falls into the polite or impolite class due to the fact that this phase is highly crucial. Below we are providing the guidelines that we followed.

**Polite:**

1. Sentence has positive words and meanings Example: অন্যরকম, ভালো লাগলো □
2. Sentences that express neutral meaning. Example: প্রথমে তো চিনতে পারিনি

**Impolite:**

1. Sentence with direct Slangs or Abusive Words Example: বাইন চুদ, বিজ্ঞাপন দেয়ার লোক পাচ্ছিস না। সালা জুকার।
2. Sarcastically making fun of someone Example : ভাগিনা তুমি কই? করোনা কবে শেষ হবে সেই ব্যাপারে যদি একটা ভবিষ্যতবানী দিতা জাতি কৃতজ্ঞ হতো!
3. Expressing negative thoughts for something with impoliteness. Example: এই ক্ষমা চাওয়াটাও আরেক রকম বাটপারি!!

Data labelings can have multiple discrepancies. So, there can be discrepancies, if a single person labels the dataset. On the other hand, multiple annotators can bring a sense of agreement to the annotation. For this part, 3 annotators from our research have labeled the comments, and the label with majority agreement is selected for the final Label. Through this, we can also reduce the bias in our research.

Table 3.2: Snippet of Labeled Dataset

| Comments | Annotator 1 | Annotator 2 | Annotator 3 | Final Label |
|---|---|---|---|---|
| বোকাচোদা একটা। | i | i | i | i |
| বাহ! প্রথমে তো চিনতেই পারিনি | i | p | p | p |
| বিবাহ করিতে চাইনা এবং angels দেখেছি। বাকিগুলিও দেখব। loved your works Tahsan Khan bhai | p | p | p | p |
| তোর দেশে ফেরার অপেক্ষায় রইলাম। জেলে একটা কামরা বুক করে রাখা থাকল । | i | i | i | i |
| হক কথা। সাবাস । | p | p | p | p |
| বাংলাদেশ ফেসবুক ইজ বিনোদন | p | p | i | p |

### 3.1.3 Exploratory Data Analysis

For text data in Natural Language Processing, it is essential to conduct a deep analysis of our data so that we may discover any insights that will aid in making data-driven decisions. Exploratory data analysis (EDA) is therefore more valued when working with text data. We proceeded by analyzing the frequency of words in the comments. We plotted these words using Word Clouds, which are graphical representations of the frequency of different words in the text. It gives greater prominence to the more common words, which are larger than the less frequent terms. Below, we are providing the visualization for the most occurring words from our Polite(p) and Impolite(i) labeled comments.

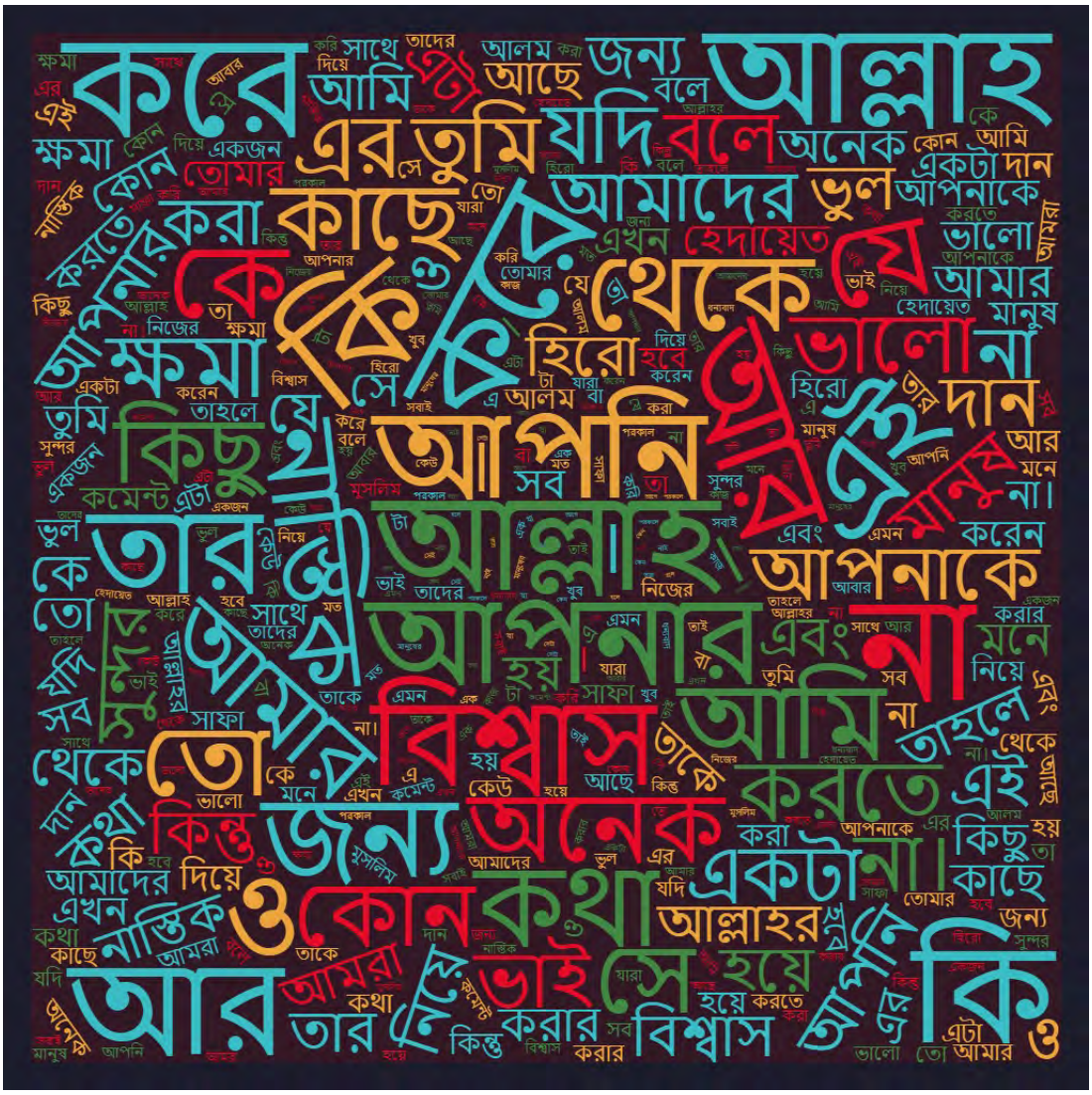



Figure 3.2: Polite Wordcloud


Figure 3.3: Impolite Wordcloud

Here, we can see the most frequently occurring words. As there were many irrelevant single words, we can handle those through the preprocessing. This will aid future detection of the most frequently used impolite or polite words when we need to paraphrase them.

Then we checked the sufficiency of polarity (polite denoted as "p" and impolite denoted as "i") for the label of the comment sentences. A pie chart of the balanced dataset is given below:

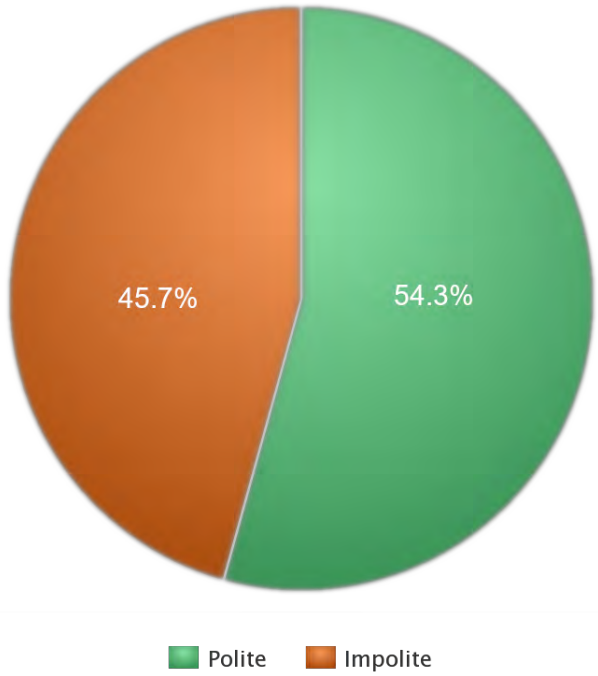



Figure 3.4: Data Label Distribution

### Conclusion from the EDA

From the above analysis, we can come upon some decisions. Those are:

1. People have used some similar types of negative words and slang that can be replaced by the same category of positive phrases to make it a polite sentence.
2. We have an adequate amount of polarity in the case of polite and impolite sentences which are respectively 54.3% and 45.7%. Therefore, the dataset is somewhat equally distributed and won't have the problem of showing bias towards a particular polarity.

### 3.1.4 Preprocessing of the Data

Before we utilize the data to train our model, one of the most critical things that needs to happen is called "data preprocessing." Data that has not been handled can result in an ineffective model. Through data preprocessing, we deal with noisy unstructured data. We make the data usable and then use it to accomplish the tasks we have set out for ourselves by preprocessing the data. Below we have used a few data preprocessing techniques from NLP which are given below,

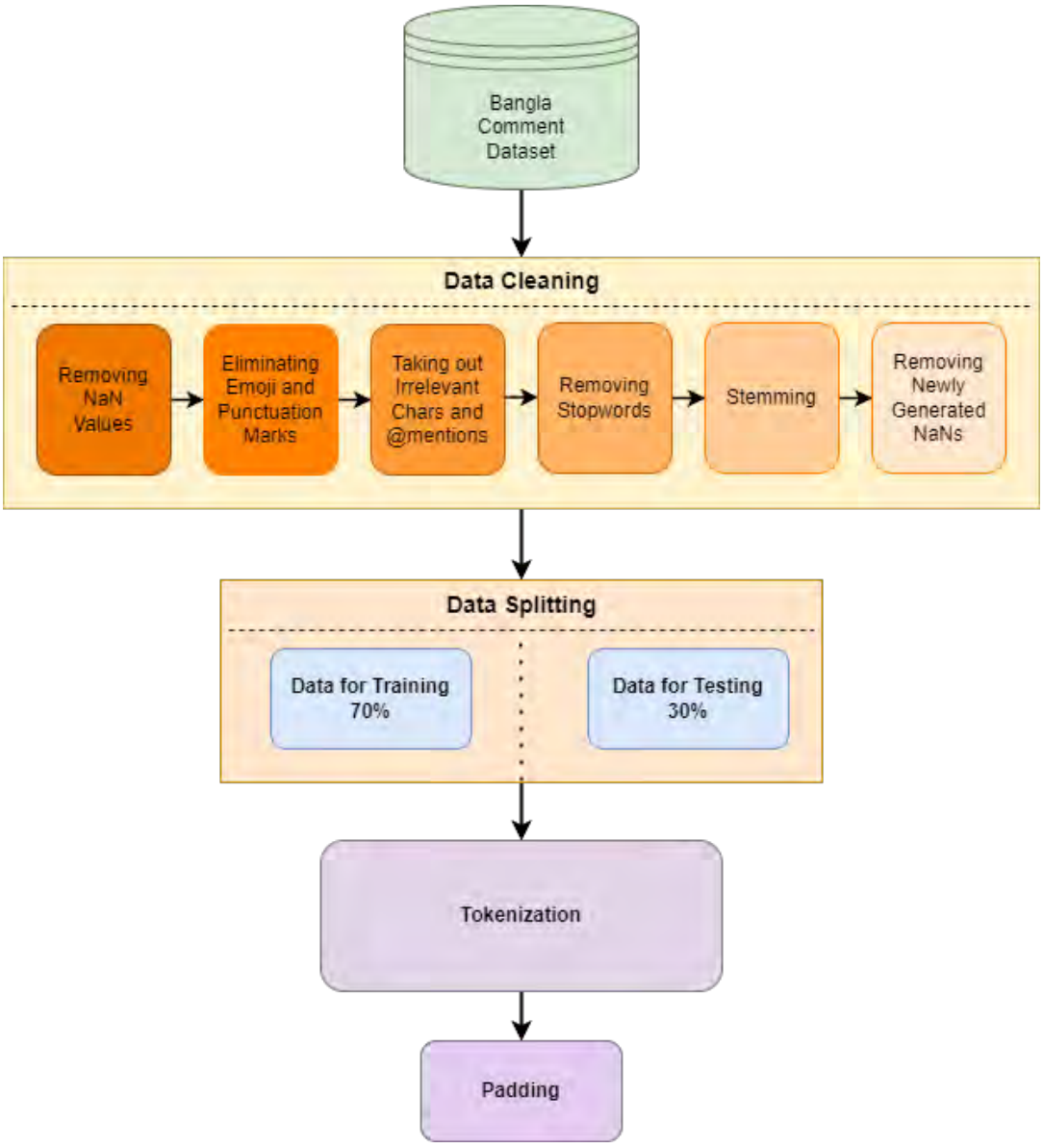



Figure 3.5: Flowchart of Data Preprocessing

### Data Cleaning

We removed irrelevant emoji, punctuation, or characters which may not be efficient for the coding and take much time so that our code runs smoothly. Moreover, there are lots of Stopwords in the comment. As an example, in English, the Stopwords are 'the', 'is', 'and' etc. Similarly, in the Bengali language, there are lots of Stopwords such as 'অনেকে', 'অন্তত', 'অথবা', 'অথচ' etc. We preprocessed Stopwords because Stopwords do not show any emotions or politeness and impoliteness for that reason to make code more effective and faster we removed stopwords. Eventually, there are some mentions in the comments that do not show any emotions for the sentence but unnecessary thoughts so we have removed those as well. As a result of data cleaning, some comments might turn null again, for example, a comment only containing emojis. So, we have to get rid of those null comments once more.

### Punctuation Removal

Punctuations do not hold any significance in text classification, so we need to remove them so that all the words are treated in an equal manner. Punctuations include noise in our data. Also, in NLP tasks it reduces the complexity and the model can be better.

### Stop Words Removal

This is also a common NLP preprocessing step. In this phase, we remove the frequently occurring terms from our dataset and retain only the words that are predominantly unique or convey meaning for sentence expression. The most common stop words such as এ, একি, এমন, অথচ, এখানে do not have an impact on the sentence classification task. We remove those using the BNLP toolkit.

### Tokenization

Tokenization is the step where the text data is converted into chunks of data. Through tokenization, we convert raw text data into data strings. It breaks the sentence into smaller units and generates meaningful parts. Also, according to different models, numerical data can be assigned to text data.

### Stemming

Stemming reduces any word to its base/root form, eg. sleeping to sleep. It is simpler than lemmatization in the sense that it only removes 'er', 'ly', 'est', 's', etc. from the end of the word. It cannot find the root of many words such as 'better' where the root word is 'good'. The stemmed word of 'better' will remain the same. Few example of Stemming in Bangla is given below:

Table 3.3: Stemming in Bangla

| Before Stemming | After Stemming |
|---|---|
| খেলা, খেলো, খেলে | খেলা |
| বাংলাদেশের, বাংলাদেশি | বাংলাদেশ |
| করছিলাম, করছি, করবো | কর |

## 3.2 Model Selection For Linguistic Politeness Classification

In the beginning, Simple RNN, GRU, CNN+bidirectional LSTM, and BERT are implemented to determine the optimal deep-learning approach for this research, with BERT demonstrating superior performance. Five Bert models are then implemented in order to improve performance and accuracy. In this section, the models and reasoning for their selection are demonstrated.

### 3.2.1 Simple RNN

In the deep learning approach for sequential or series data modeling, a Recurrent Neural Network (RNN) is a special type of Feedforward Neural Network model with a significant feature known as a 'Hidden State' or 'Memory State' that can remember the outcome or specific parameters of the previous state of a sequence and feed them to the current state. All input and output data elements in conventional feedforward neural networks are independent of each other. If the data are arranged in a sequence, there must be dependencies between the data elements that recall previous details and predict subsequent details. The Hidden Layer in RNN solves this dependency issue. The concept behind the Hidden State in RNN is to store the 'memory' of details of the previous state in order to generate the subsequent state by performing the same task at every step with the same parameters, thereby reducing the parameter complexity. In addition, RNN incorporates a feedback loop that primarily assists in predicting the next state output using the previously hidden state input. However, the weights that circulate throughout the network remain unchanged. If we considered the Hidden state as Hi and the input state at each step by Xi, the formula for generating the next state would be h= (UX + Wh-1 + B). Hence, RNN is considered the standard feedforward NN model. The structure of an RNN model would look like this:

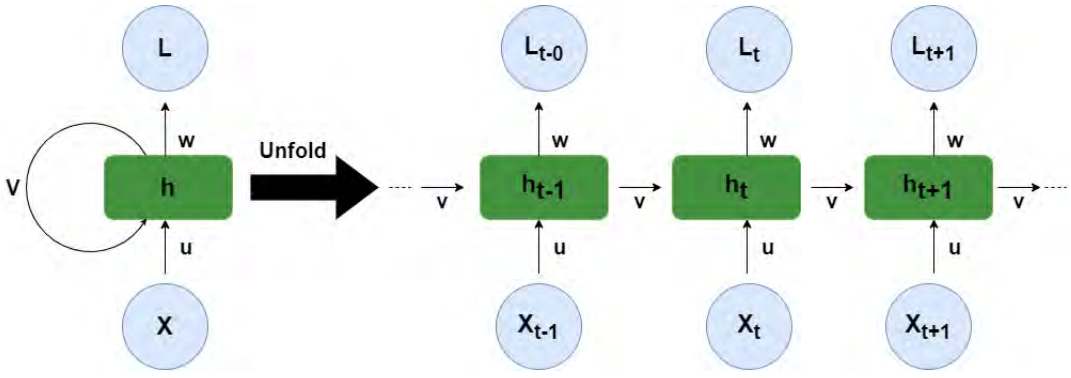



Figure 3.6: Simple Recurrent Neural Network

### 3.2.2 LSTM

Long Short Term Memory is a specialized kind of recurrent neural network (RNN) that is capable of dealing with the Vanishing Gradient Problem, which is the issue of long-term order dependencies in sequential data in RNN. In the case of long-timescale dependencies, even though RNN can make accurate predictions, it cannot reserve long-term memory due to network blockage, which creates a gap in the network and reduces performance efficiency. As the LSTM network enables the long-term retention of information, the LSTM memory cell is used as a building component for RNN layers. Consequently, it is appropriate for tasks such as machine transition, speech recognition, language modeling, and time series forecasting. As a result, we decided to use LSTM as one of our analysis models. LSTM typically has three gates for determining which information to discard and which to retain, which information is significant, and where the next hidden cell should be. The advantage of using LSTM for our classification model is that it will retain essential information and efficiently determine which class the information belongs to. On the other hand, LSTM is a complex model which deals with a lot of computations and there are a large number of parameters, making it very time-consuming and challenging to train and analyze the model. In addition, LSTM confronts the problem of exploding gradient, which causes divergence and unpredictability in long-term dependencies, prompting us to examine additional models. For a better understanding, the basic LSTM architectural diagram and LSTM network are given below:

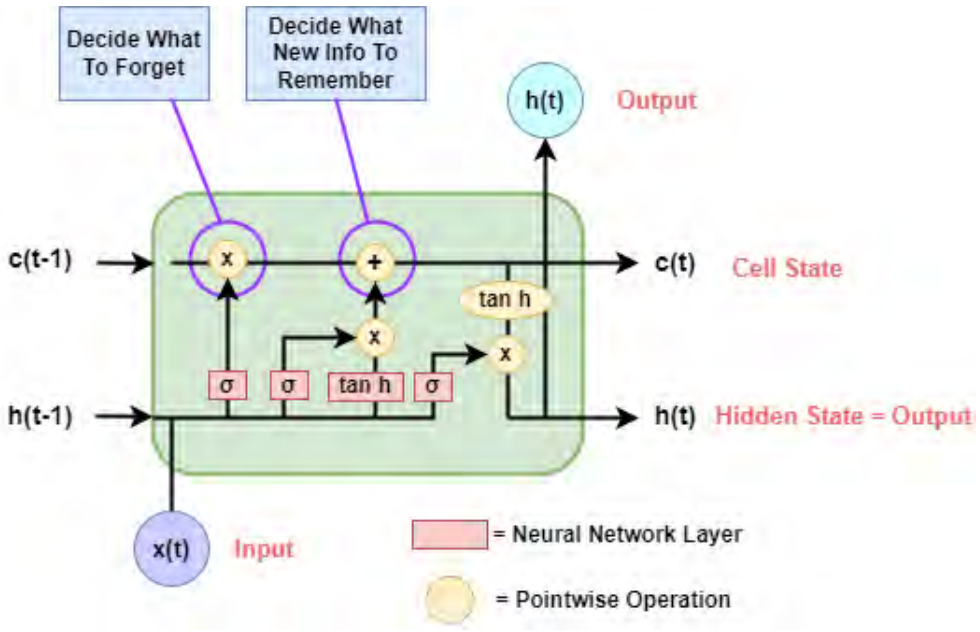



Figure 3.7: Long Short Term Memory

### 3.2.3 CNN + Bidirectional-LSTM

The CNN Bidirectional LSTM is a hybrid model suitable for modeling sequence data or NLP-related tasks, with two LSTM layers for processing input in both the forward and reverse directions. CNN is most commonly applied for the categorization of images, but it is also capable of classifying text. In this implementation, we used CNN layers in the outer portion of the model and LSTM layers alongside a dense layer in the output portion. CNN performs well in this instance in terms of extracting features, while LSTM is responsible for the interpretation aspect. Here, CNN excels at extracting features, while LSTM excels at interpreting them. We have used a 1D layer on top of the texts to classify the text input we have. The vocabulary size was set to 25000, the embedding dimension was 300, and the maximum length was 100. The embedding algorithm was then executed, with the embedding dimension set to 300 and the maximum length set to 100. Two layers of Bidirectional LSTM were implemented. In addition, we introduced dropout layers and L2 regularization to the model to prevent it from overfitting the data. On average, the accuracy of the test set was satisfactory. For a better understanding, the structure of this hybrid model is given below:

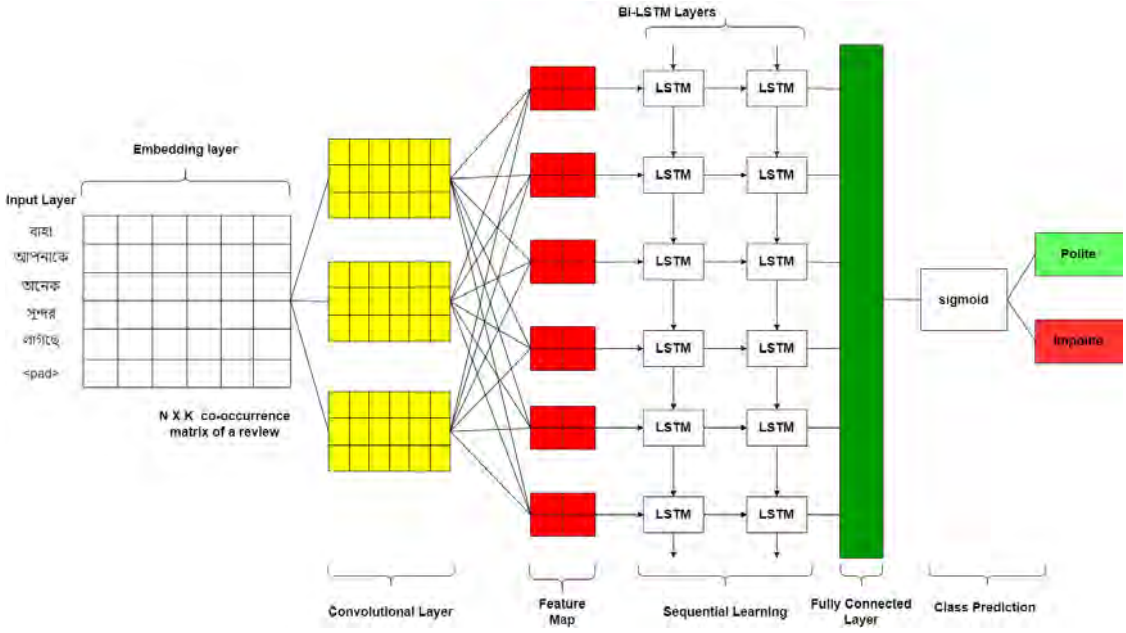



Figure 3.8: CNN + Bidirectional-LSTM

### 3.2.4 GRU

The Gated Recurrent Unit (GRU) is an alternative to the Long Short Term Memory (LSTM) kind of Recurrent Neural Network. At each time interval, the gating mechanism in GRU updates the hidden state of LSTM, which controls the movement of data in and out of the network. The reset gate and the update gate constitute the GRU's gating mechanism. GRU operates with fewer parameters (3 weight matrices) and utilizes less memory, so it is faster and less susceptible to overfitting than LSTM. LSTM and GRU both operate with sequential data, but LSTM performs more accurately with extended sequences. It is also capable of solving the exploding vanishing problem and is equally functional as LSTM. Consequently, GRU is preferable to LSTM in certain circumstances.

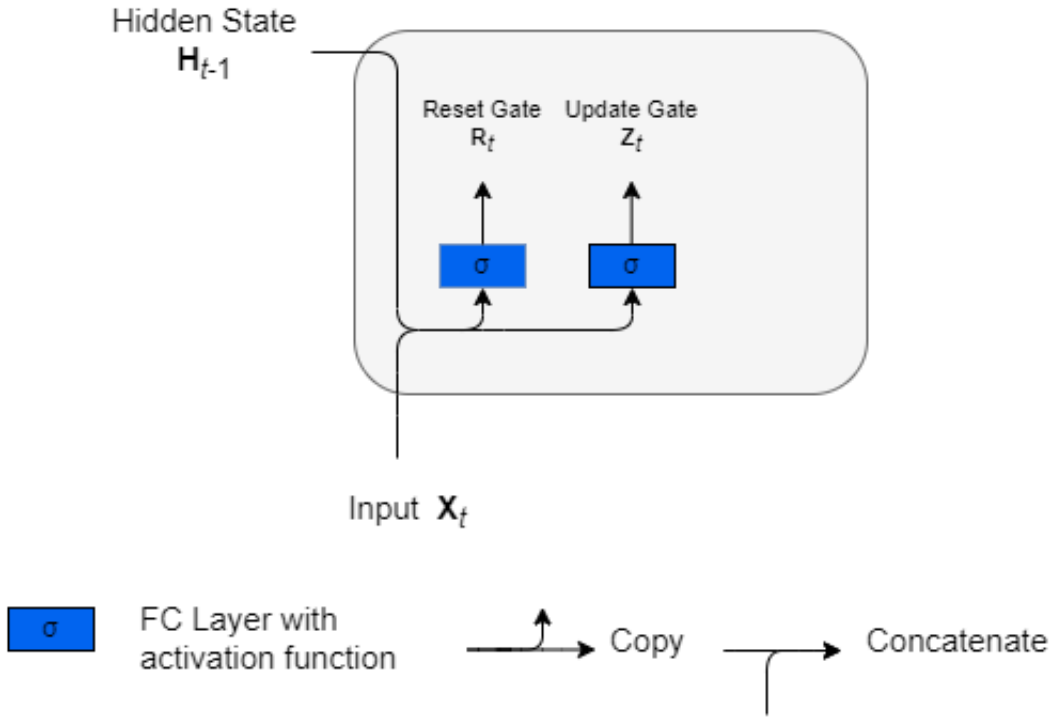



Figure 3.9: The Gated Recurrent Unit

### 3.2.5 BERT

Bidirectional Encoder Representations from Transformers is what BERT stands for. It allows us to establish a contextual connection among texts. BERT functions as an encoder-decoder, receiving input data and producing output. It can read the entire text input as a single text, which is indicative of the Bidirectional Workflow. Pretraining and precise tuning are the two components of BERT. BERT acquires a language by simultaneously training on two unsupervised tasks. The BERT models have been implemented using the ktrain container class. The model was implemented with the learner function as the final step. The observed highest accuracy was 84%, which is the highest of all methods utilized.

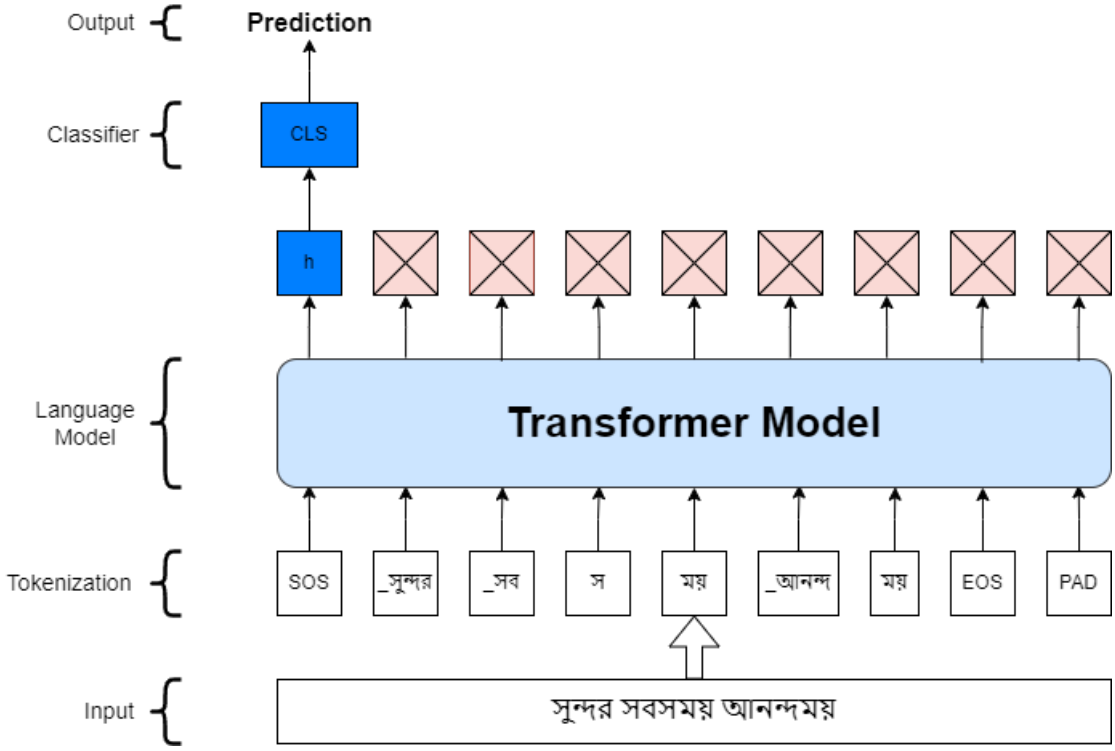



Figure 3.10: BERT

We tried out different variants of BERT to find out what works best with our research objective. A brief description of each variant is as follows,

#### BERT multilingual base (uncased)

The uncased version of Bert based multilingual transformer model fundamentally focuses on fine-tuning masking tasks for an entire text and classifying the sequence or executing tokenization. Unlike the case version, before the tokenization stage, the text or sentence is lowercase. In other words, the uncased version is not case-sensitive.

#### BERT multilingual base (cased)

The BERT multilingual base (cased) model is particularly case-sensitive. Before the tokenization step the sentence is the same as the input sentence here. BERT cased

models usually work better where details of sentences are significant such as parts of speech tagging.

**Bangla BERT Base**

It is a pre-trained Bengali language transformer model that uses language masking. The model is proposed by Sagor Sarkar as per the name capable of training and masking a large corpus. We trained our dataset that is pre-labeled manually. The model has 768 hidden layers and about 110M parameters. This model previously excellently accomplished the best outcome compared to Bengali electra and mBERT models in the Bengali text classification task.

**DistilBERT base (uncased)**

The faster and more efficient version of the BERT model can likely pre-train a large dataset in a self-supervised manner known as Distilbert. The important feature of the model is the three pre-trained categories namely Distillation loss, Masked language modeling, and Cosine embedding loss. We have taken the uncased version of the model for classification purposes.

**BanglaBERT**

It performs similarly to other BERT models. According to its name, it was developed by the CSE researchers at Buet. This model is an ELECTRA classifier model that has been pre-trained with an individual normalization pipeline and fine-tuned with checkpoints to attain outstanding outcomes. Additionally, this model incorporates pre-training checkpoints for the BanglaBert classifier. The model is trained with a significant number of Bengali datasets in order to achieve the best performance in natural language processing (NLP) for the Bengali language. This model is not case-sensitive, as it is an uncased model.

### 3.2.6 Text-to-Text Transfer Transformer (T5)

As the name suggests, the T5 model is a transformer that can solve various NLP tasks such as machine translation, classification, regression tasks, document summarization, etc. It uses an encoder-decoder architecture. T5 is trained on common crawl web extracted text named C4 dataset. In the T5 model, a few words get removed from the original text sequence with unique tokens at random. Then tokens get replaced with other sentinel tokens. They do it by using a BERT-sized transformer. Then fine-tune it using GLUE, ASQuAD, WTM14, etc., and evaluate validation by choosing the best result.

## 3.3 Implementing Dictionary

At first, we tried to transfer politeness right after being classified as impolite. However, the model we are using to paraphrase is BanglaT5. The transformer model was trained on the majority of inoffensive data from [29]. It was intentionally filtered in such a way so that generated sentences are always unbiased and presentable. However, we explicitly needed to deal with the impoliteness of sentences. However, during the study, we came across a method of substituting impolite words with a tag and substituting the tag with polite words [17]. Authors had various words labeled as p0 or P9 where P0 is least polite and P9 is most polite. Inspired by their approach, we prepared a dictionary of swear words to combat the issue of intentionally filtering impolite sentences in the dataset.

### 3.3.1 Dictionary Formulation

A handful of profane words (e.g. অসভ্য, কুত্তার বাচ্ছা, কুলাঙ্গার, দালাল, বাল etc.) were persistent throughout the dataset which were causing confusion to BanglaT5. So, we decided to make a custom dictionary that contains the most frequently used swear words as keys and subsequent polite words as values in a key: value pair format. A total of 410 curse words/phrases were addressed in our dictionary.

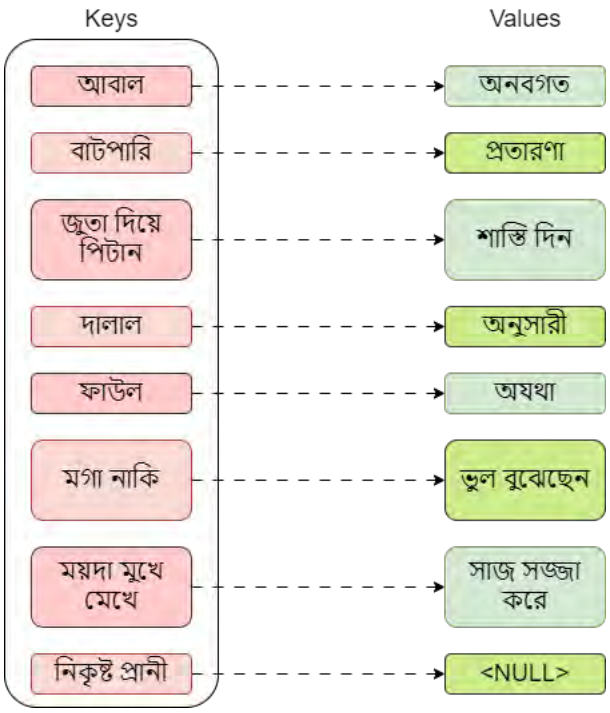



Figure 3.11: Snippet of the Dictionary

### 3.3.2 Mechanism of the Dictionary

After classifying a sentence as impolite, we took the sentence through our filter i.e. the dictionary to combat the weakness of the BanglaT5 model. If a curse word/phrase were found, it would be replaced by its polite counterpart. But, not all impolite sentences had derogatory terms in them. If that was the case, the sentence would directly go to the paraphraser. In some cases, there were hateful words that could not be replaced with anything polite. So we left the value NULL (The last word from Figure 3.11).

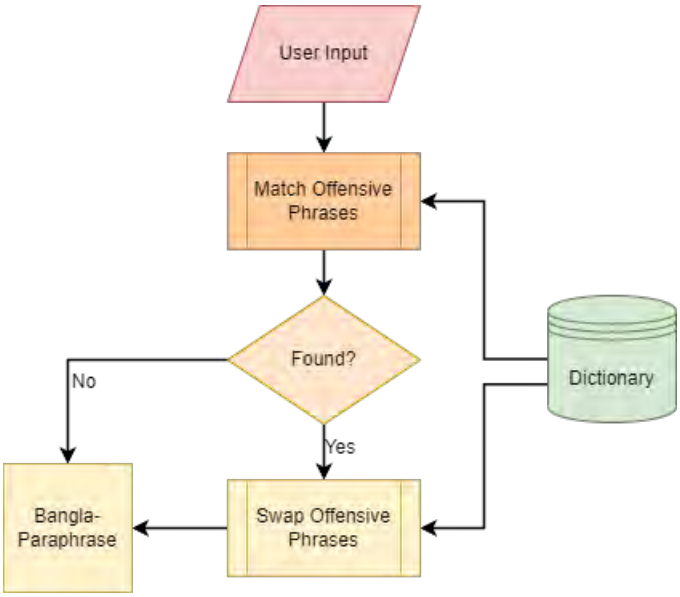



Figure 3.12: Dictionary Procedure

### 3.3.3 BanglaParaphrase

To generate the final output we used BanglaParaphrase which is a model fine-tuned over the BanglaT5 model. To generate paraphrases the authors trained BanglaBERT using a token classification head, utilizing a dataset of 30 parts-of-speech (POS) tags. The corpus used for training consisted of 7393 phrases, which in turn contained 102937 tokens. The training process consisted of 20 epochs, where each epoch had a batch size of 32. The learning rate used was 0.00002, and a linear learning rate scheduler was employed. The dataset was partitioned into three subsets, namely the training set, test set, and validation set, following an 80:10:10 ratio. Following the completion of the training process, the POS tagger successfully assigned tags to seven specifically selected components of speech, namely VM (Main verb), VA (Auxiliary Verb), JJ (Adjective), NV (Verbal Noun), AMN (Adverb of Manner), ALC (Adverb of location), and NST (Spatio Temporal Noun).

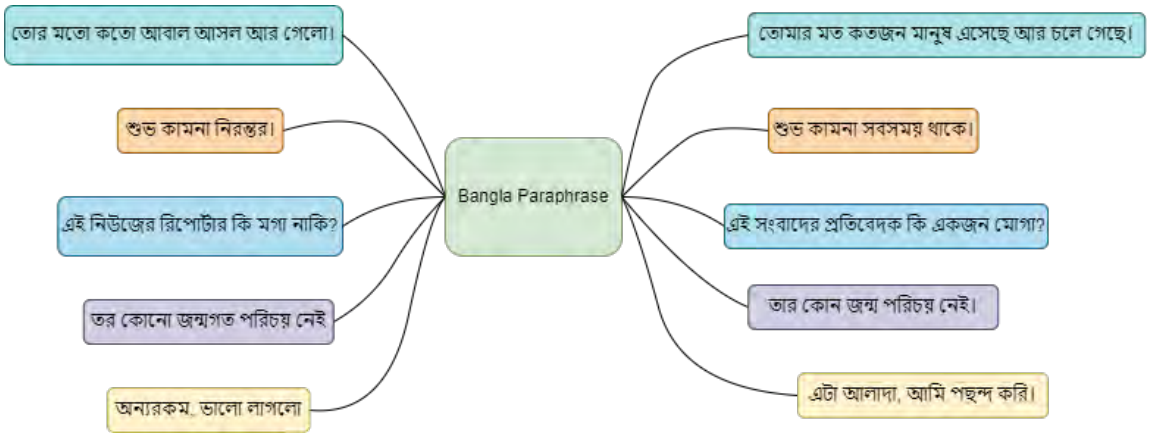



Figure 3.13: BanglaParaphrase Outputs

However, the authors were not satisfied with the results as it can be improved by filtering the dataset further so the model does not generate biased or toxic Bangla texts. They also want to add more tasks to score in the future to make it a benchmark tool for everything Bangla-related.

### 3.3.4 Final output

Sentences that went through all the filters are turned into tensors to make them easy to understand for the machine. We are using PyTorch tensors. Sentences get normalized so the model does not form any bias toward large integer values. Next, we use the tokenizer from BanglaT5 to turn our normalized tokens into PyTorch tensors. Then nonparaphrased tensors are put into the T5 model to generate sequential paraphrased tokens. Lastly, we use a batch decoder to turn those numerical tokens into words that form the paraphrased sentence.

Table 3.4: Effectiveness of the Dictionary

| User Input | Output using only BanglaParaphrase | Output with Dictionary & BanglaParaphrase |
|---|---|---|
| তোর মতো কতো আবাল আসল আর গেলো। | তোমার মত কতজন মানুষ এসেছে আর চলে গেছে। | তোমার মত কতজন অপরিচিত লোক আসল আর গেল। |
| আপনার মতো লোক দেখানো শয়তান ইমানধারকে জুতা দিয়ে পিটানো দরকার। | তোমার মত মানুষকে দেখানোর জন্য, শয়তান ইমানদারকে জুতা দিয়ে মারা উচিত। | তোমার মত মানুষকে দেখানো ইমানদারকে শাস্তি দাও। |
| আপনারা কি তার দালাল নাকি সে যা বলবে তা প্রচার করবেন। | আপনি কি তার দালাল হবেন নাকি তিনি যা বলবেন, তা প্রচার করবেন? | আপনি কি তাঁর অনুসারী হবেন অথবা তিনি যা বলেন, তা প্রচার করবেন। |
| এই নিউজের রিপোর্টার কি মগা নাকি? | এই সংবাদের প্রতিবেদক কি একজন মোগা? | এই সংবাদের প্রতিবেদক কি ভুল করেছেন? |
| তুই রাস্তার জারজ সন্তান,তর কোনো জন্মগত পরিচয় নেই, কারণ তুই তো না দেখা জিনিস বিশ্বাস করিস না | তুমি রাস্তার আবর্জনার ছেলে, তার কোন জন্ম পরিচয় নেই, কারণ তুমি যা দেখনি তাতে বিশ্বাস | তুমি রাস্তার ছেলে, তার কোন জন্ম পরিচয় নেই, কারণ তুমি যা দেখনি তাতে বিশ্বাস করো না |

# Chapter 4

# Results and Discussion

This study was conducted based on a dataset by self-labeling for sentence classification model training and for making sentence composers. Also, a dictionary is made from the previous data set to change the abusive impoliteness from sentences before paraphrasing. In this research, the pre-trained BERT models were suggested as they are trained on vast amounts of data, and some deep learning models like SimpleRNN, GRU, and CNN-BiLSTM were also applied. BERT architecture models were utilized for comparison with Deep learning models. A confusion matrix was built for each method to demonstrate the accuracy of each class. Eventually, for greater comprehension and research we provide precision, recall, F1-score, and support score.

## 4.1 Performance Metrics:

The evaluation of the model's performance was conducted using metrics like accuracy, precision, recall, and F1-score. The performance metrics are given below.

Accuracy means the overall correctness of a model. It provides the idea of a model's performance and how much it has done well. However, in some cases, it does not provide actual results where the dataset is not well organized or correctly labeled. For positive and negative classes, the accuracy is measured by this given formula.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Here,
TP = True Positive; (TP refers to the count of accurately anticipated positive cases.)
TN= True Negative : ( TN refers to that instincts were correctly predicted as negative because they belong to actually negative class)
FP = False Positive; (FP refers to the situations that have been forecasted as positive, but in reality, they are negative.)
FN =False Negatives (FN indicates the count of what is really positive, but they were incorrectly classified as negative by the model)

Precision refers to the quantification of the level of accuracy in the positive predictions generated by a model. Also, it means the model could predict how many

correct predictions are made from all the positive predictions. Precision measures the ability of a model to avoid making false positive predictions. For example, in text classification, high precision means that by the system if a text is predicted as an impolite sentence, there is a high chance that the sentence is actually impolite. The formula for calculating precision is given below.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall refers to the quantification of a model's capacity to accurately identify all pertinent positive instances within a given dataset. Moreover, a greater recall value signifies improved performance in correctly recognizing positive cases inside the model. In order to calculate the percentage of correct positive predictions, it is necessary to know how many positive instances are overall present in the collection. For example, in text classification, a high recall rate indicates that the system possesses the ability to accurately identify all instances of texts including a polite term, hence minimizing the occurrence of false negatives, whereas texts containing the polite phrase are erroneously categorized as impolite. The formula for calculating recall is given below.

$$\text{Recall} = \frac{TP}{TP + FN}$$

The F1 score is a metric that integrates precision and recall into a unified measure, hence providing a fair assessment of a model's performance. Also, it is the mean of precision and recall and it falls within the range of 0 to 1. The F1 score is very useful for scenarios where in the dataset there is an imbalance between the number of positive and negative instances. To calculate the F1 Score, the formula is given below.

$$\text{F1 Score} = \frac{2 \times Precision \times Recall}{\text{Precision+Recall}}$$

## 4.2 Linguistic Politeness Classification Model:

To correctly address linguistic politeness We tried classical deep-learning models and advanced transformer-based models. The results from the models are as follows,

### 4.2.1 Deep Learning Approach

The deep learning approaches applied for this research are SimpleRNN, GRU, and CNN-BiLSTM. The percentage of accuracy of SimpleRNN is 73%, GRU is 77%, and CNN-BiLSTM is 77% in deep learning approaches. Below we are providing the predicted classes for 3915 comment data using the deep learning models.

### Simple RNN

This model is an elementary form of RNN (Recurrent Neural Network). It is a simple data processing deep learning approach for sequential data as it processes data sequentially. The accuracy for this model is 73%. This model can accurately predict 1644 polite labels and 1227 impolite labels. The confusion matrix for SimpleRNN is given below:

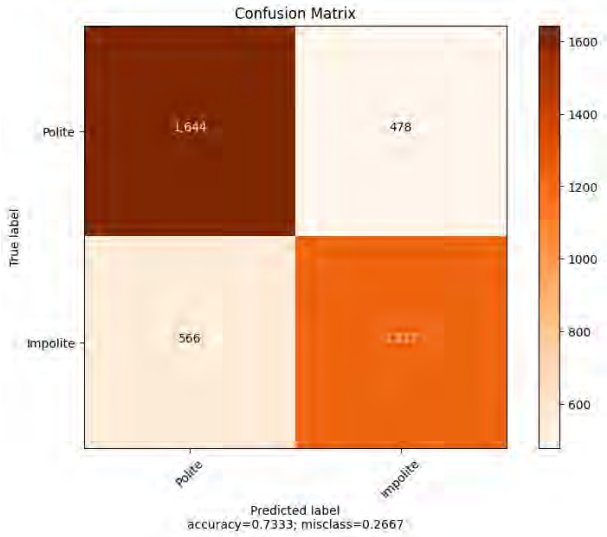



Figure 4.1: Confusion Matrix of SimpleRNN

In the table below, the Accuracy, Precision, Recall, and F1-score of Simple RNN are given,

Table 4.1: Classification Result of SimpleRNN

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.74 | 0.77 | 0.76 | 2122 |
| Impolite | 0.72 | 0.68 | 0.70 | 1793 |
| Accuracy | | | 0.73 | 3915 |
| Macro Avg | 0.73 | 0.73 | 0.73 | 3915 |
| Weighted Avg | 0.73 | 0.73 | 0.73 | 3915 |

### GRU (Gated Recurrent Unit)

GRU (Gated Recurrent Unit) is a variant of RNN (Recurrent Neural Network) that tackles some of the drawbacks of RNNs, as an example of the vanishing gradient issue. As it is a deep learning strategy that uses neural networks with multiple neural networks. The accuracy for this model is 77%. This model can accurately predict 1718 polite labels and 1304 impolite labels. The confusion matrix for GRU (Gated Recurrent Unit) is given below.

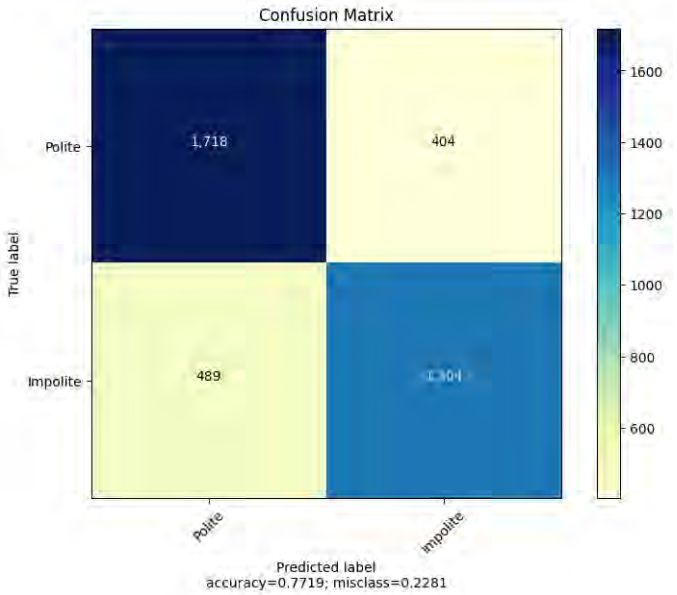



Figure 4.2: Confusion Matrix of GRU

Now, for both sentiments, the precision value, recall value, and f1-score vary which is given in the below table.

Table 4.2: Classification Result of GRU

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.78 | 0.81 | 0.79 | 2122 |
| Impolite | 0.76 | 0.73 | 0.74 | 1793 |
| Accuracy | | | 0.77 | 3915 |
| Macro Avg | 0.77 | 0.77 | 0.77 | 3915 |
| Weighted Avg | 0.77 | 0.77 | 0.77 | 3915 |

### CNN-BiLSTM

This model is a hybrid of two models CNN (Convolutional Neural Network) and BiLSTM (Bidirectional Long Short-Term Memory) and these two models are deep learning architectures. The accuracy for this model is 77%. This model can accurately predict 1806 polite labels and 1213 impolite labels. The confusion matrix for CNN-BiLSTM is given below.

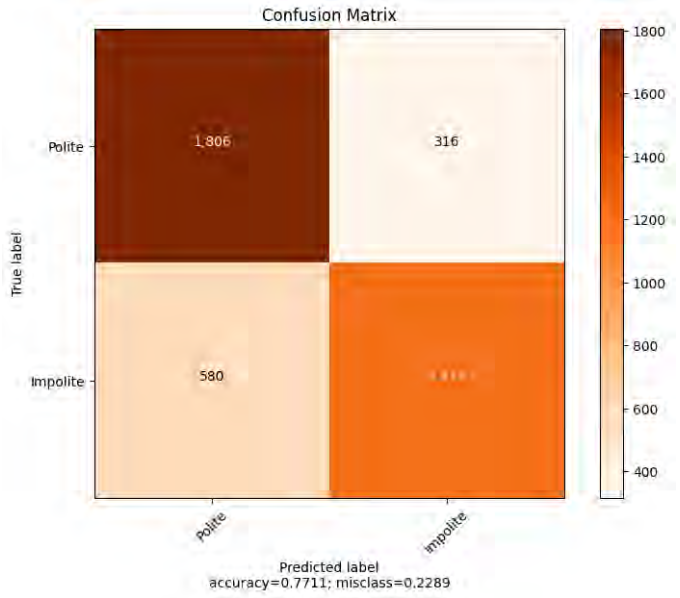



Figure 4.3: Confusion Matrix of CNN-BiLSTM

Now, for both sentiments, the precision value, recall value, and f1-score vary which is given in the below table,

Table 4.3: Classification Result of CNN-BiLSTM

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.76 | 0.85 | 0.80 | 2122 |
| Impolite | 0.79 | 0.68 | 0.73 | 1793 |
| Accuracy | | | 0.77 | 3915 |
| Macro Avg | 0.78 | 0.76 | 0.77 | 3915 |
| Weighted Avg | 0.77 | 0.77 | 0.77 | 3915 |

### 4.2.2 Transformer Based Deep Learning Approach

The transformer architecture based- deep learning approaches for this research are BanglaBERT, BERT multilingual base (uncased), BERT multilingual base (cased), Bangla BERT Base, DistilBERT base (uncased), BERT base (uncased). The percentage of accuracy of BanglaBERT is 84%, BERT multilingual base (uncased) is 83%, BERT multilingual base (cased) is 82%, Bangla BERT Base is 82%, DistilBERT base (uncased) is 81%, and BERT base (uncased) is 81% in transformer-based approaches.

#### BanglaBERT

This model is designed especially for the Bengali language and it is also a part of the BERT (Bidirectional Encoder Representations from Transformers) family. The accuracy for this model is 84%. This model can accurately predict 1807 polite labels and 1481 impolite labels. The confusion matrix for BanglaBERT is given below.

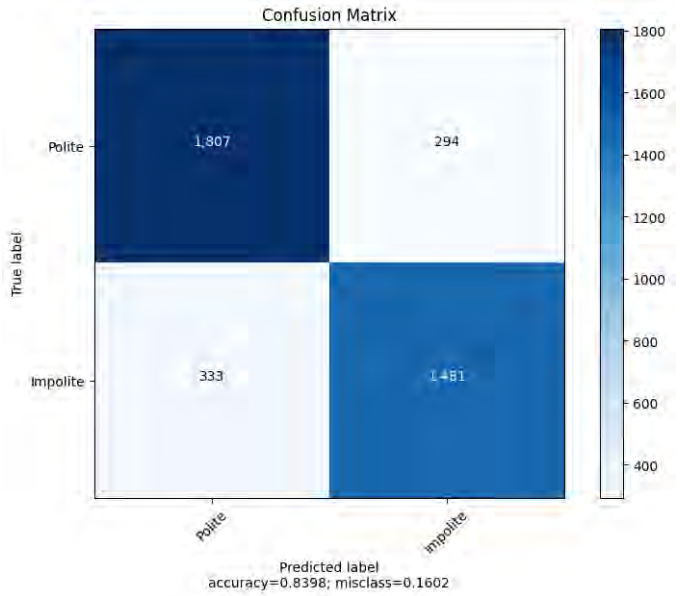



Figure 4.4: Confusion Matrix of BanglaBERT

Now, for both sentiments, the precision value, recall value, and f1 score are the highest which is given in the below table.

Table 4.4: Classification Result of BanglaBERT

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.84 | 0.86 | 0.85 | 2101 |
| Impolite | 0.83 | 0.82 | 0.83 | 1814 |
| Accuracy | | | 0.84 | 3915 |
| Macro Avg | 0.84 | 0.84 | 0.84 | 3915 |
| Weighted Avg | 0.84 | 0.84 | 0.84 | 3915 |

### BERT Multilingual base (uncased)

This is also a part of the BERT (Bidirectional Encoder Representations from Transformers) family for multilingual texts which does not save any case information that converts all text to lowercase while tokenizing. The accuracy for this model is 83%. This model can accurately predict 1811 polite labels and 1420 impolite labels. The confusion matrix for the BERT multilingual base (uncased) is given below.

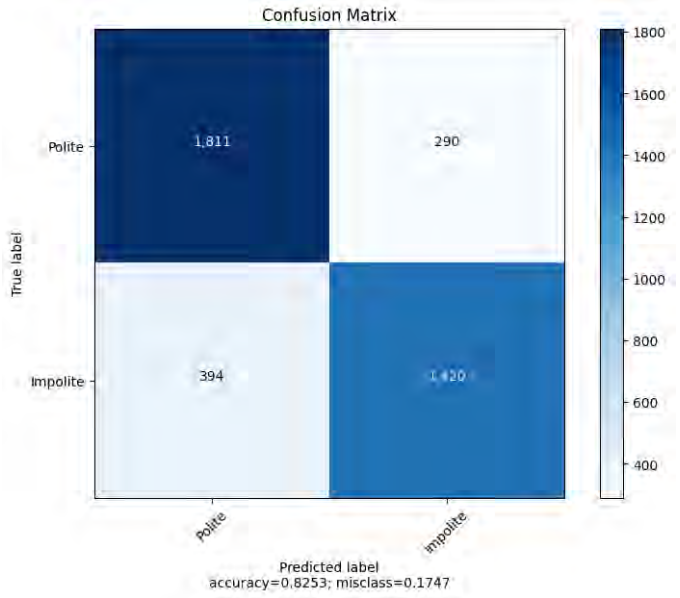



Figure 4.5: Confusion Matrix of BERT multilingual base (uncased)

Now, for both sentiments, the precision value, recall value, and f1-score vary which is given in the below table.

Table 4.5: Classification Result of BERT multilingual base (uncased)

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.82 | 0.86 | 0.84 | 2101 |
| Impolite | 0.83 | 0.78 | 0.81 | 1814 |
| Accuracy | | | 0.83 | 3915 |
| Macro Avg | 0.83 | 0.82 | 0.82 | 3915 |
| Weighted Avg | 0.83 | 0.83 | 0.82 | 3915 |

### BERT Multilingual base (cased)

This model belongs to the BERT (Bidirectional Encoder Representations from Transformers) family and can process multiple languages. This case-type variant maintains text and handles capital and lowercase letters as separate tokens. The accuracy of this model is 82%. This model can accurately predict 1776 polite labels and 1419 impolite labels. The confusion matrix for this model is given below.

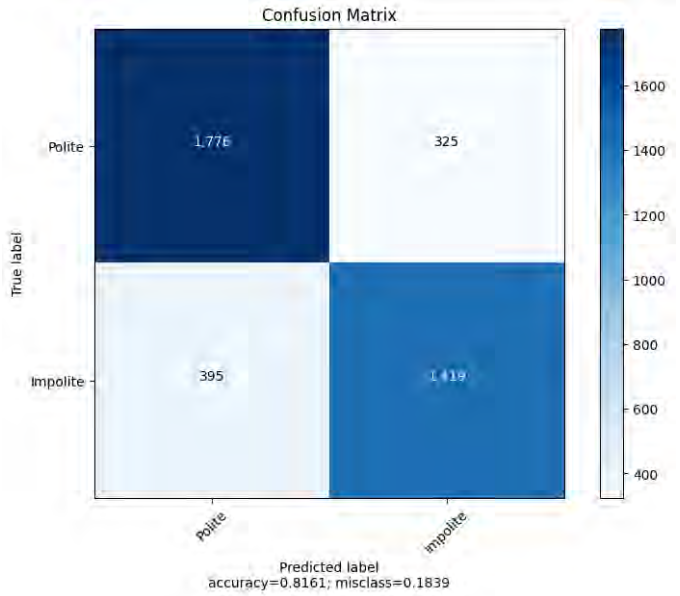



Figure 4.6: Confusion Matrix of BERT multilingual base (cased)

Now, for both sentiments, the precision value, recall value, and f1-score vary which is given in the below table.

Table 4.6: Classification Result of BERT multilingual base (cased)

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.82 | 0.85 | 0.83 | 2101 |
| Impolite | 0.81 | 0.78 | 0.80 | 1814 |
| Accuracy | | | 0.82 | 3915 |
| Macro Avg | 0.82 | 0.81 | 0.81 | 3915 |
| Weighted Avg | 0.82 | 0.82 | 0.82 | 3915 |

### BanglaBERT Base

This model is also another part of the BERT (Bidirectional Encoder Representations from Transformers) family model and it is mainly based on transformer architecture and NLP-based works. The accuracy for this model is 82%. This model can accurately predict 1781 polite labels and 1412 impolite labels. The confusion matrix for Bangla BERT Base is given below.

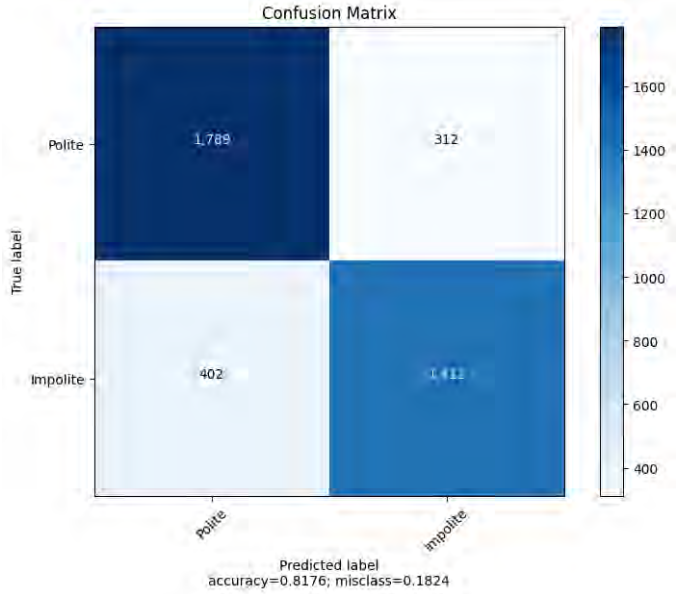



Figure 4.7: Confusion Matrix of BanglaBERT Base

Now, for both sentiments, the precision value, recall value, and f1-score vary which is given in the below table.

Table 4.7: Classification Result of BanglaBERT Base

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.82 | 0.85 | 0.83 | 2101 |
| Impolite | 0.82 | 0.78 | 0.80 | 1814 |
| Accuracy | | | 0.82 | 3915 |
| Macro Avg | 0.82 | 0.81 | 0.82 | 3915 |
| Weighted Avg | 0.82 | 0.82 | 0.82 | 3915 |

### DistilBERT base (uncased)

This model is part of a model that is known as a small and faster version of the BERT (Bidirectional Encoder Representations from Transformers) and that is distilbert. As this is also an uncased variant, it does not save case information. The accuracy for this model is 81%. This model can accurately predict 1763 polite labels and 1412 impolite labels. The confusion matrix for the DistilBERT base (uncased) is given below.

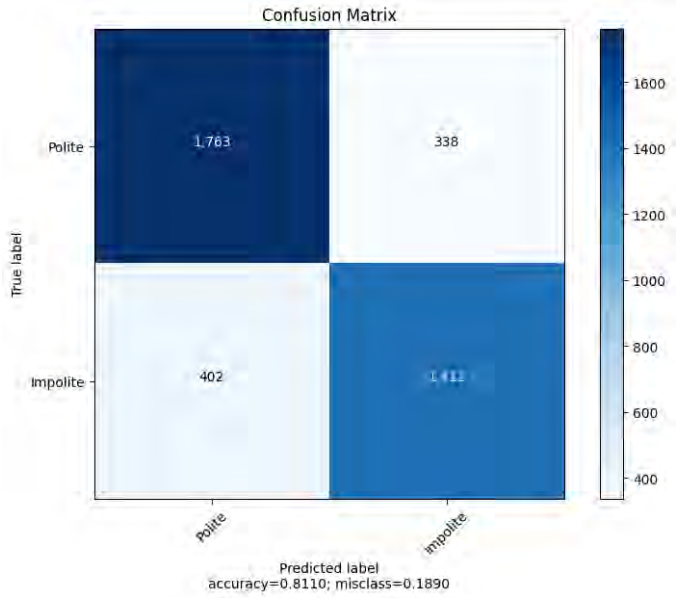



Figure 4.8: Confusion Matrix of DistilBERT base (uncased)

Now, for both sentiments, the precision value, recall value, and f1-score vary which is given in the table below.

Table 4.8: Classification Result of DistilBERT base (uncased)

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.81 | 0.84 | 0.83 | 2101 |
| Impolite | 0.81 | 0.78 | 0.79 | 1814 |
| Accuracy | | | 0.81 | 3915 |
| Macro Avg | 0.81 | 0.81 | 0.81 | 3915 |
| Weighted Avg | 0.81 | 0.81 | 0.81 | 3915 |

### BERT base (uncased)

This is similar to before mentioned that it's also a part of the BERT (Bidirectional Encoder Representations from Transformers) family and this does not save case information as it is an uncased variant. The accuracy for this model is 81%. This model can accurately predict 1758 polite labels and 1414 impolite labels. The confusion matrix for the BERT base (uncased) is given below.

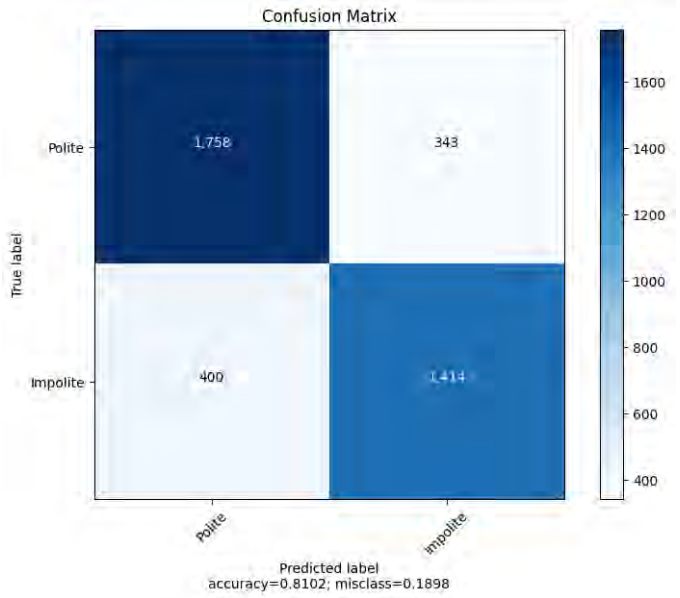



Figure 4.9: Confusion Matrix of BERT base (uncased)

Now, for both sentiments, the precision value, recall value, and f1-score vary which is given in the table below.

Table 4.9: Classification Result of BERT base (uncased)

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Polite | 0.81 | 0.84 | 0.83 | 2101 |
| Impolite | 0.80 | 0.78 | 0.79 | 1814 |
| Accuracy | | | 0.81 | 3915 |
| Macro Avg | 0.81 | 0.81 | 0.81 | 3915 |
| Weighted Avg | 0.81 | 0.81 | 0.81 | 3915 |

### 4.2.3 Comparative Analysis

A summary categorization result of all models is shown in the table below. Here, for each category, the following metrics are listed: precision, recall, f1-score, support values, accuracy, macro average, and weighted average.

Table 4.10: Classification Result for the selected model, transformer architecture base-deep learning approach, and Deep Learning Approach

| Models | Class | Polite | Impolite | Accuracy | Macro AVG | Weighted AVG |
|---|---|---|---|---|---|---|
| BanglaBERT (Selected Model) | Precision | 0.84 | 0.83 | - | 0.84 | 0.84 |
| | Recall | 0.86 | 0.82 | - | 0.84 | 0.84 |
| | F1 Score | 0.85 | 0.83 | 0.84 | 0.84 | 0.84 |
| | Support | 2101 | 1814 | 3915 | 3915 | 3915 |
| BERT Multilingual base (uncased) | Precision | 0.82 | 0.83 | - | 0.83 | 0.83 |
| | Recall | 0.86 | 0.78 | - | 0.82 | 0.83 |
| | F1 Score | 0.84 | 0.81 | 0.83 | 0.82 | 0.82 |
| | Support | 2101 | 1814 | 3915 | 3915 | 3915 |
| BERT Multilingual base (cased) | Precision | 0.82 | 0.81 | - | 0.82 | 0.82 |
| | Recall | 0.85 | 0.78 | - | 0.81 | 0.82 |
| | F1 Score | 0.83 | 0.80 | 0.82 | 0.81 | 0.82 |
| | Support | 2101 | 1814 | 3915 | 3915 | 3915 |
| Bangla BERT Base | Precision | 0.82 | 0.82 | - | 0.82 | 0.82 |
| | Recall | 0.85 | 0.78 | - | 0.81 | 0.82 |
| | F1 Score | 0.83 | 0.80 | 0.82 | 0.82 | 0.82 |
| | Support | 2101 | 1814 | 3915 | 3915 | 3915 |
| DistilBERT base (uncased) | Precision | 0.81 | 0.81 | - | 0.81 | 0.81 |
| | Recall | 0.84 | 0.78 | - | 0.81 | 0.81 |
| | F1 Score | 0.83 | 0.79 | 0.81 | 0.81 | 0.81 |
| | Support | 2101 | 1814 | 3915 | 3915 | 3915 |
| BERT base (uncased) | Precision | 0.81 | 0.80 | - | 0.81 | 0.81 |
| | Recall | 0.84 | 0.78 | - | 0.81 | 0.81 |
| | F1 Score | 0.83 | 0.79 | 0.81 | 0.81 | 0.81 |
| | Support | 2101 | 1814 | 3915 | 3915 | 3915 |
| SimpleRNN | Precision | 0.74 | 0.72 | - | 0.73 | 0.73 |
| | Recall | 0.77 | 0.68 | - | 0.73 | 0.73 |
| | F1 Score | 0.76 | 0.70 | 0.73 | 0.73 | 0.73 |
| | Support | 2122 | 1793 | 3915 | 3915 | 3915 |
| GRU | Precision | 0.78 | 0.76 | - | 0.77 | 0.77 |
| | Recall | 0.81 | 0.73 | - | 0.77 | 0.77 |
| | F1 Score | 0.79 | 0.74 | 0.77 | 0.77 | 0.77 |
| | Support | 2122 | 1793 | 3915 | 3915 | 3915 |
| CNN-BiLSTM | Precision | 0.76 | 0.79 | - | 0.78 | 0.77 |
| | Recall | 0.85 | 0.68 | - | 0.76 | 0.77 |
| | F1 Score | 0.80 | 0.73 | 0.77 | 0.77 | 0.77 |
| | Support | 2122 | 1793 | 3915 | 3915 | 3915 |

## 4.3 Performance Study of Paraphraser

For paraphrasing sentences, we have used the BanglaParaphrase paraphraser. At first, we detect whether the word is an abusive sentence or not after that we paraphrase those sentences with proper affection without changing their meaning. A comparison of outputs generated, with and without the dictionary, is given below,

Table 4.11: Example of sentence composing with proper affection

| Polite/Impolite (Input) | Given Input | Output through the dictionary and BanglaParaphrase | Output through only BanglaParaphrase |
|---|---|---|---|
| Polite input | আলহামদুলিল্লাহ। এ সজীবতা নিয়ে সুস্থ্য ও নিরাপদে থাকুন এ দোয়া করি। | আলহামদুলিল্লাহ। এ সজীবতা নিয়ে সুস্থ্য ও নিরাপদে থাকুন এ দোয়া করি। | আলহামদুলিল্লাহ। আমি প্রার্থনা করি এই জীবিত অবস্থায় সুস্থ ও নিরাপদ থাকুন। |
| Impolite input | সাফা কবির কোন **** বাল...??, | সাফা কবির কোন মানুষ ...??, | সাফা কবির কোন মানুষ...? |
| Impolite input | বালের তামাশা শুরু করছস | তুমি মজা শুরু করছ? | ের তামাশা শুরু করছস |
| Impolite input | ফইন্নির ঘরের ফইন্নি | ভাইর ঘরের ভাই | ভাইয়ের বাড়ির ভাই। |
| Impolite input | ও তো অনেক কিছু না দেখে বিশ্বাস করে।আসলে বিখ্্যাত হতে চাচেছ ২ পয়সার নাচনেওয়ালী বেশ্যা। | ও তো অনেক কিছু না দেখে বিশ্বাস করে।আসলে বিখ্্যাত হতে চাচেছ ২ পয়সার আপা। | সে অনেক কিছু দেখে বিশ্বাস করে, আসলে সে ২-পেনের বোন হতে চায়। |
| Impolite input | এটা একটা এক নাম্বারের নাস্তিকদের পেইজ | এটা একটা এক নাম্বারের দের পেইজ | এটা এক নাম্বারের পেজ। |
| Polite input | চেহারা টা দেখেন, কতো ইনোসেন্ট! | চেহারা টা দেখেন, কতো ইনোসেন্ট! | তোমার চেহারা দেখো, কত সুন্দর! |
| Impolite input | মুসলমান না হলে কোন সমস্যা নেই। তোমার জন্য দোজখের আগুন ই যথেষ্ট। তবে হ্যাঁ আল্লাহ তোমাকে হেদায়েত দান করুন। | মুসলমান না হলে কোন সমস্যা নেই। তোমার জন্য দোজখের আগুন ই যথেষ্ট। তবে হ্যাঁ আল্লাহ তোমাকে হেদায়েত দান করুন। | যদি তুমি মুসলমান না হও, তাহলে কোন সমস্যা নেই। কিন্তু আল্লাহ তোমাকে পথনির্দেশ দেবেন। |
| Impolite input | তোমার মতো বেয়াদব আর নাই,তোমার কোনো ক্ষমা নাই | তোমার মতো মানুষ আর নাই,তোমার কোনো ক্ষমা নাই | তোমার মত আর কেউ নেই, তোমার কোন ক্ষমা নেই। |
| Polite input | এখন আমরা কি করতে পারি | এখন আমরা কি করতে পারি | এখন আমরা কী করতে পারি? |

## 4.4 Score Analysis and Model Selection

The two sorts of works employed by the team members in this study were data categorizing by Binary Classification of Linguistic Politeness using the Transformer Architecture base-deep learning technique and the Deep Learning approach and sentence composing (paraphrasing) with proper affection. The most accurate models for transformer architecture base-deep learning technique were BanglaBERT(84%), BERT multilingual base (uncased)(83%), BERT multilingual base (cased)(82%), Bangla BERT Base (82%), DistilBERT base (uncased) (81%), BERT base (uncased) (81%), and for Deep Learning approach the accurate results were SimpleRNN (73%), GRU (77%), CNN-BiLSTM (77%). According to this research, among all these models the best model is BanglaBERT with 84% accuracy for the transformer architecture base-deep learning approach. Here, deep learning approaches did not give good accuracy because these models were not properly fine-tuned. Moreover, from the paraphraser, it is seen that the sentence composer is implied and those input sentences and paraphrased sentences are given by the team members in this study. As an example, a polite sentence which is given in input as ``চেহারা টা দেখেন, কতো ইনোসেন্ট!", after that, it is matched with the dictionary to find the impolite words from the dictionary and it changes to ``চেহারা টা দেখেন, কতো ইনোসেন্ট!", then passing it through the paraphraser it changes to ``তোমার চেহারা দেখো, কত সুন্দর!" that is more accurate and polite. Also, an impolite sentence given as input ``ফইন্নির ঘরের ফইন্নি", after matching with the created dictionary it changes the sentence as ``ভাইর ঘরের ভাই" and after passing it through the paraphraser the final output is ``ভাইয়ের বাড়ির ভাই।", which is a more polite and appropriate sentence. The final output of the sentences was quite accurate but not fully accurate, some of the sentences gave proper output with proper affection but still, some sentences only gave the exact result except for abusing words. This was not done properly to compute the appropriate sentence with proper affection for every sentence. In the end, sentence composer was working well for a few texts but not for all and the best result from sentence classification was by the model BanglaBERT with 84% accuracy.

# Chapter 5

# Conclusion

## 5.1 Conclusion

In today's environment, we need to maintain constant communication with other individuals. The actual message is not always conveyed accurately because there are barriers to communication between the sender and the recipient. As a result, there is a requirement for a system based on intelligence that will assist in reducing the communication gap between the sender and the receiver. We have suggested using our sentence-composer approach to improve this communication's efficiency. We have given a model that consists of methodologies. The first phase allows us to identify unpleasant statements. If any impolite terms are still there, the second step allows us to map them to their polite equivalents. We plan to accomplish the aforementioned objective with the help of the sentence composer model that we have developed. The majority of the models that were implemented were based on deep learning or transformers. We found that BERT was the most accurate and provided the best outcomes when it came to detecting them. We decided to use the BanglaBERT model for our detection, and it provided us with precision, recall, and F1-score of 84%, respectively. In addition, our one-of-a-kind construction of a dictionary can transfer the sentences into a better shape and express the appropriate level of courtesy. Through the use of our research paper, we have investigated recent works and the contributions that they have made to this field. In addition to this, we learned about the most advanced models now available as well as the restrictions associated with them.

## 5.2 Future Works

As part of this research work, our objective is to determine whether or not a given statement contains impolite language. If we successfully detect impoliteness, we will run the sentence via a dictionary-based method that converts it into a more appropriate polite form. In this project, one of our goals is to complete the task in a manner that preserves the original meaning of the sentences. The sentence is rephrased into a more respectful form, but the meaning is preserved. The research that we have done has various areas in which it could be improved. For the time being, we have utilized pre-trained models in order to determine the detection part. In our future work, one of our goals is to fine-tune a wide range of state-of-the-art models and figure out which one produces the greatest outcome. In addition, as part of our technique for phrase rephrasing, we will also fine-tune the t5 base model by using a unique dataset that we have created. This model will be effective at rephrasing any kind of sentence by utilizing its training vocabulary.

# References


[1] T. N. R. Card, *Study Finds Only 27% of Middle  High School Students Proficient in Writing: How to Make Sure Your Child Excels*, https://tenneyschool.com/middle-high-school-students-proficient-writing-child-excels/, [Online; accessed September 25, 2023], 2020.

[2] A. Foundation, *American Millennials Most Likely to Engage in Trolling Behavior Finds Avast Foundation*, https://press.avast.com/american-millennials-most-likely-to-engage-in-trolling-behavior-finds-avast-foundation, [Online; accessed September 25, 2023], 2021.

[3] Z. Yuan and Y. Park, *The Psychological Toll of Rude E-mails*, https://www.scientificamerican.com/article/the-psychological-toll-of-rude-e-mails/, [Online; accessed September 25, 2023], 2020.

[4] T. E. 200, *What are the top 200 most spoken languages?* https://www.ethnologue.com/insights/ethnologue200/, [Online; accessed September 25, 2023], 2023.

[5] M. F. Ahmed, Z. Mahmud, Z. T. Biash, A. A. N. Ryen, A. Hossain, and F. B. Ashraf, "Bangla online comments dataset," *Mendeley Data*, vol. 1, 2021.

[6] C. Lam, "Linguistic politeness in student-team emails: Its impact on trust between leaders and members," *IEEE Transactions on Professional Communication*, vol. 54, no. 4, pp. 360–375, 2011.

[7] Y. Baruch, "Response rate in academic studies-a comparative analysis," *Human relations*, vol. 52, no. 4, pp. 421–438, 1999.

[8] M. S. Akter, "The effectiveness of polite language in social behavior,"

[9] A. Applebee, J. Langer, I. Mullis, A. Latham, and C. Gentile, "National assessment of educational progress 1992: Writing report card," *Washington, DC: US Government Printing Office*, 1994.

[10] J. K. Jameson, "Negotiating autonomy and connection through politeness: A dialectical approach to organizational conflict management," *Western Journal of Communication (includes Communication Reports)*, vol. 68, no. 3, pp. 257–277, 2004.

[11] X. Luo, "Quantifying the long-term impact of negative word of mouth on cash flows and stock prices," *Marketing Science*, vol. 28, no. 1, pp. 148–165, 2009.

[12] Z. Efia, *How Do Negative Words Impact People?* https://www.voicesofyouth.org/blog/how-do-negative-words-impact-people, [Online; accessed September 25, 2023], 2021.

[13] C. O. Alm, D. Roth, and R. Sproat, "Emotions from text: Machine learning for text-based emotion prediction," in *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, 2005, pp. 579–586.

[14] S. Aman and S. Szpakowicz, "Identifying expressions of emotion in text," in *International Conference on Text, Speech and Dialogue*, Springer, 2007, pp. 196–205.

[15] C. Danescu-Niculescu-Mizil, M. Sudhof, D. Jurafsky, J. Leskovec, and C. Potts, "A computational approach to politeness with application to social factors," *arXiv preprint arXiv:1306.6078*, 2013.

[16] S. T. Hsu, C. Moon, P. Jones, and N. Samatova, "A hybrid cnn-rnn alignment model for phrase-aware sentence classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 443–449.

[17] A. Madaan, A. Setlur, T. Parekh, *et al.*, "Politeness transfer: A tag and generate approach," *arXiv preprint arXiv:2004.14257*, 2020.

[18] L. Martin, B. Sagot, E. de la Clergerie, and A. Bordes, "Controllable sentence simplification," *arXiv preprint arXiv:1910.02677*, 2019.

[19] M. Han, O. Wu, and Z. Niu, "Unsupervised automatic text style transfer using lstm," in *National CCF Conference on Natural Language Processing and Chinese Computing*, Springer, 2017, pp. 281–292.

[20] V. Madaan, P. Agrawal, N. Sethi, V. Kumar, and S. K. Singh, "A novel approach to paraphrase english sentences using natural language processing," *International Journal of Control Theory and Applications*, vol. 9, no. 11, 2016.

[21] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," *arXiv preprint arXiv:1511.08198*, 2015.

[22] A. Bhattacharjee, T. Hasan, W. U. Ahmad, and R. Shahriyar, *Banglanlg and banglat5: Benchmarks and resources for evaluating low-resource natural language generation in bangla*, 2023. arXiv: 2205.11081 [cs.CL].

[23] K. Stowe, N. Beck, and I. Gurevych, "Exploring metaphoric paraphrase generation," in *Proceedings of the 25th Conference on Computational Natural Language Learning*, 2021, pp. 323–336.

[24] J. J. Bird, A. Ekárt, and D. R. Faria, "Chatbot interaction with artificial intelligence: Human data augmentation with t5 and language transformer ensemble for text classification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3129–3144, 2023.

[25] A. Bhattacharjee, T. Hasan, W. U. Ahmad, and R. Shahriyar, "Banglanlg: Benchmarks and resources for evaluating low-resource natural language generation in bangla," *arXiv preprint arXiv:2205.11081*, 2022.

[26] J. Xie, "Fine-grained emotional paraphrasing along emotion gradients," *arXiv preprint arXiv:2212.03297*, 2022.

[27] A. Akil, N. Sultana, A. Bhattacharjee, and R. Shahriyar, "Banglaparaphrase: A high-quality bangla paraphrase dataset," *arXiv preprint arXiv:2210.05109*, 2022.

[28] Z. Li, X. Jiang, L. Shang, and H. Li, "Paraphrase generation with deep reinforcement learning," *arXiv preprint arXiv:1711.00279*, 2017.

[29] T. Hasan, A. Bhattacharjee, K. Samin, *et al.*, "Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for bengali-english machine translation," *arXiv preprint arXiv:2009.09359*, 2020.