OPEN ACCESS
Vietnam Journal of Computer Science
Vol. 10, No. 3 (2023) 329–356
© The Author(s)
DOI: 10.1142/S2196888823500021

Natural Language Processing and Sentiment Analysis on Bangla Social Media Comments on Russia–Ukraine War Using Transformers

Mahmud Hasan*, Labiba Islam†, Ismat Jahan‡, Sabrina Mannan Meem§
and Rashedur M. Rahman¶

Department of Electrical and Computer Engineering
North South University, Dhaka 1229, Bangladesh
**mahmud.hasan03@northsouth.edu*
†*labiba.islam@northsouth.edu*
‡*ismat.jahan@northsouth.edu*
§*sabrina.meem@northsouth.edu*
¶*rashedur.rahman@northsouth.edu*

Received 16 January 2023
Revised 6 March 2023
Accepted 6 March 2023
Published 4 May 2023

The Bangla Language ranks seventh in the list of most spoken languages with 265 native and non-native speakers around the world and the second Indo-Aryan language after Hindi. However, the growth of research for tasks such as sentiment analysis (SA) in Bangla is relatively low compared to SA in the English language. It is because there are not enough high-quality publically available datasets for training language models for text classification tasks in Bangla. In this paper, we propose a Bangla annotated dataset for sentiment analysis on the ongoing Ukraine–Russia war. The dataset was developed by collecting Bangla comments from various videos of three prominent YouTube TV news channels of Bangladesh covering their report on the ongoing conflict. A total of 10,861 Bangla comments were collected and labeled with three polarity sentiments, namely Neutral, Pro-Ukraine (Positive), and Pro-Russia (Negative). A benchmark classifier was developed by experimenting with several transformer-based language models all pre-trained on unlabeled Bangla corpus. The models were fine-tuned using our procured dataset. Hyperparameter optimization was performed on all 5 transformer language models which include: BanglaBERT, XLM-RoBERTa-base, XLM-RoBERTa-large, DistilmBERT and mBERT. Each model was evaluated and analyzed using several evaluation metrics which include: F1 score, accuracy, and AIC (Akaike Information Criterion). The best-performing model achieved the highest accuracy of 86% with 0.82 F1 score. Based on accuracy, F1 score and AIC, BanglaBERT outperforms baseline and all the other transformer-based classifiers.

Keywords: Natural language processing; sentiment analysis; transformers; Russia–Ukraine war.

[¶] Corresponding author.

1. Introduction

Natural Language Processing (NLP) is an area of Artificial Intelligence (AI) and Linguistics, devoted in making the machines (computers) perceive the knowledge of statements and spoken words written in human language. The existence of NLP facilitated the users' wish to be able to communicate with computers (machines) in human (natural) language.[36] Its application has been extensively proliferated in various research fields among which sentiment analysis has been the dominant one in the recent past.

Sentiment analysis, also known as opinion mining, is the process of dealing and obtaining information about people's opinions, behavioral responses and attitudes, and emotions expressed towards certain events or topics of conversation represented in the form of text. Sentiments are inherently subjective.[36] People may interpret the same text having a different standpoint towards it. As every text bears some sentiment, the texts are labeled into different categories, such as positive, negative, or neutral sentiment. The usefulness of sentiment classification or sentiment analysis can be found by social media monitoring, allowing us to gather a comprehensive overview of the wider public opinion behind certain topics.[36]

With the growing interest and active participation in social media, the growth of online users using digital media for communicating and expressing themselves globally is helping the organizations and individuals extract information required to know the sentiments and opinions of users worldwide.[11] Sentiment analysis tasks have put a massive impact on various real-life applications, especially, on determining the sentiment polarity of English texts based on public opinions on real-life events. Due to the outstanding efforts and the amount of proficient work devoted in R&D for the formation of large public datasets such as Stanford Sentiment Treebank,[40] SentiWordNet,[6] IMDB review corpus,[30] Sentiment140,[15,32] and SemEval Twitter sentiment analysis corpus,[9] users are able to further experiment and explore immensely on this field.

Bangla is the first language spoken by nearly 230 million native speakers worldwide, 160 million of whom are Bangladeshi. The majority of research works and publicly available datasets found on sentiment analysis are limited to English and other resource-rich languages, while Bangla is still at a developmental phase.[18] Bangla is a morphologically rich language that has evolved over thousands of years with its everlasting tradition, including a variety of dialects.[3] Low-resource languages like Bangla lack quality Bangla datasets required for training computational models for sentiment analysis.[22] However, in recent times, the use of social networking sites is dramatically increasing and so is the participation of Bangladeshi people in online activities. Besides English-written comments, people write comments in Bangla as well. Analyzing different blogging and social media sites, the public comments regarding any specific event can be easily extracted and collected from various sources.

Addressing the aforementioned limitations behind unavailability of public resources and benchmarks in Bangla, in this paper, we created a Bangla annotated sentiment analysis dataset based on the context of Ukraine–Russia War. Till now, the datasets or corpus available in Bangla are mostly collections of public comments on news and videos, and reviews on movies and products covering multiple domain fields. However, our dataset is mainly focused on the events of war that occurred in Ukraine and Russia in the recent past, analyzing the sentiments of Bangladeshi people towards this war. This dataset is trained on multiple transformer models for sentiment classification in Bangla rather than classical machine learning and deep learning models in order to understand the difference in model and training complexity, and the required computation time. Our contributions in this paper are as follows:

- We created a sentiment analysis dataset in Bangla texts on the Ukraine–Russia war. A total of 10,861 comments were collected from YouTube videos of three most popular Bangladeshi news channels: Jamuna News, SomoyTV News and Independent News.
- To enhance the caliber of our dataset, we applied data curation techniques, carefully cleaned and then split the data into train, validation and test sets.
- We experimented on five pretrained transformer models to perform sentiment analysis in Bangla such as, multilingual BERT (mBERT), Distil-mBERT, XLM-RoBERTa, XLM-RoBERTa-large, BanglaBERT to show better performance compared to other existing deep learning and classical machine learning models.
- We aim to make our dataset publicly available for further research, more specifically, on the context of the Ukraine–Russia war.

2. Related Work

Sentiment analysis in Bangla has become one of the researchers' newest study horizons. Over the years, a considerable amount of research work has been done by the AI research community in this field. To help make distinction between the works related to sentiment analysis, Table 1 represents a summary of works that are presented as follows.

Apoorv *et al.* through their work[2] primarily make two contributions. They develop models for two distinct classification tasks: a binary task for classifying sentiment into positive and negative categories, and a three-way task for classifying sentiment into positive, negative, and neutral categories. A feature-based model, a tree kernel-based model, and a unigram model are the three types of models they have experimented with. They report an overall increase of over 4% for two classification tasks: a binary, positive against negative and a 3-way, positive versus negative versus neutral.

This paper[14] extends to the classification of customer reviews using sentiment analysis. The unigram feature extraction approach is used to analyze labeled datasets. To calculate the similarity, they used the Semantic Orientation-based WordNet and machine learning-based classification methods Naïve Bayes, Maximum entropy,

Table 1. Tabular representation of related works of sentiment classification with English and Bangla languages.

Study	Study purpose	Dataset	Methods/Techniques	Results
Apoorv *et al.*[2]	English tweets (Sentiment Analysis)	8,753 tweets collected from a commercial source	Unigram model, Tree kernel model, 100 Senti-features model, Kernel plus Senti-features, Unigram plus Senti-features	60.83% accuracy
Gautam *et al.*[14]	English tweets (Sentiment Analysis)	19,340 tweets	Naive Bayes, Maximum entropy, Support Vector Machine, and Semantic Analysis (WordNet)	89.9% accuracy
Victoria *et al.*[10]	Ukranian and Russian News (Sentiment Analysis)	2,349 comments collected from Russian news (http://censor.net.ua/) and 2,450 comments collected from Ukranian news (https://tsn.ua/)	Naive Bayes, DMNBtext, NB Multinomial, SVM, and Feature Selection methods	F1 score of 0.82
Hande *et al.*[16]	Kannada (Sentiment Analysis and Offensive Language Identification)	7,671 comments collected from YouTube	Logistic Regression, SVM, Multinomial Naive Bayes, KNN, Decision Tree, Random Forest	59% accuracy for sentiment analysis. 66% accuracy for offensive language detection.
Hoq *et al.*[21]	Bangla (Sentiment Analysis)	32,000 comments from Facebook	Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) and hybrid of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM)	90.49% accuracy and F1 score of 92.83%
Masum *et al.*[31]	Bangla (Sentiment Analysis)	9,009 unique comments collected from famous Bengali news portals	SVM, CNN, LSTM, Bi-LSTM	78.75% accuracy, 79.38% F1-score, 80.46% precision, 78.33% recall
Isaac *et al.*[29]	English tweets (Sentiment Analysis)	11.15 million tweets named "Ukraine Conflict Twitter Dataset" collected from {https://www.kaggle.com/}	RNN Model	92% accuracy
Wahid *et al.*[43]	Bangla texts (Cricket Sentiment Analysis)	10,000 comments collected from ABSA dataset and online resources such as Facebook, YouTube, Prothom-Alo, BBC Bangla and Bdnews.com	Recurrent Neural Network with an LSTM model	95% accuracy

Table 1. (*Continued*)

Study	Study purpose	Dataset	Methods/Techniques	Results
Islam *et al.*[23]	Bangla (dataset formation and Sentiment Analysis)	15,728 comments collected from social media comments on news articles and videos	SVM, RNN, pre-trained transformer-based language models	64.09% F1-score
Romim *et al.*[37]	Bangla (Dataset and detection of online Bangla hate Speech)	50,200 offensive comments from online social networking sites	SVM and Bi-LSTM models	F1-score of 91.04%
Kastrati *et al.*[25]	Low-resource languages such as Albanian language (dataset formation and Sentiment Analysis)	10,742 comments collected from official Facebook page of NIPHK	1D-CNN, BiLSTM, Hybrid 1D-CNN + BiLSTM, SVM, Naive Bayes, Decision Tree and Random Forest	F1-score of 72.09%
Nafis *et al.*[42]	Bangla (Multilabel Sentiment and Emotion Detection)	15,689 comments collected from various kinds of YouTube videos	LSTM and CNN models	95% accuracy
Rahman *et al.*[35]	Bangla (Document Classification)	Open-source data: 50,560 data from the BARD, 78,796 data from the OSBC, and 1,28,761 data from the ProthomAlo dataset	BERT, ELECTRA	96.39% accuracy, 94.18% F1-score
Arid *et al.*[17]	Bangla (Sentiment Classification)	Several publicly available datasets used	Random Forest, SVM, CNN-W2V Glove, FastText, BERT, DistilBERT, XLM-RoBERTa	Transformer-based models showed best results
Samsul *et al.*[24]	Bangla (Sentiment Analysis)	Data collected from social media platforms	Feature Extraction techniques, RNN, LSTM, GRU, CNN-BiLSTM, Bangla-BERT, mBERT	88.59% accuracy

and SVM are applied. The naïve bayes approach, which performs better than maximal entropy and SVM, is put through a unigram model, which performs better than using it alone. When WordNet's semantic analysis is followed by the aforementioned technique, the accuracy is further enhanced, rising to 89.9% from 88.2%.

In this study,[10] Victoria *et al.* investigated the issue of sentiment analysis for Ukrainian and Russian news, created a corpus of Ukrainian and Russian news, and annotated each text using one of the three categories: positive, negative, or neutral. Several techniques, including SVM and Bayesian classifiers (Naive Bayes Multinomial), were used in Machine Learning experiments to assess the annotation (Support Vector Machine). The bag of words strategy combined with feature selection generated the best results, proving that even basic learning algorithms like Naive Bayes may provide satisfactory accuracy (average F1-score of 0.73) when given the correct features.

Hande *et al.*[16] contained a multi-task learning dataset named KanCMD that was constructed by them from scratch. It is a Kannada language-based dataset which is a language of Karnataka state in India. They applied it for abusive language identification as well as sentiment classification. The dataset contains a total of 7,671 data. They evaluated the dataset total by 6 models which are Logistic Regression (LR), SVM, Multinomial Naive Bayes (MNB), KNN, Decision Tree (DT) and Random Forest (RF). LR, RF classifiers and DT performed comparatively better than others. On the other hand, SVM performed very poorly for this dataset in analyzing sentiment which was unpredicted.

To identify emotion in Bangla texts, Hoq *et al.*[21] built four models using a hybrid of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) with different Word Embeddings, including Embedding Layer, Word2Vec, Global Vectors (Glove), and Continuous Bag of Words (CBOW) (words, sentences). They have implemented a hybrid model that combines CNN and LSTM for the Bangla language and have deployed CNN for it. The model can define the three fundamental emotions-happy, sad and angry. For a viable dataset, they used Facebook Bangla comments out of 5,640 annotated data points from 32,000 comments. Incorporating a Word2Vec embedding layer and a hybrid CNN-LSTM that recognizes emotions from raw textual data, the most effective model got an F1 score of 92.83% and an accuracy of 90.49%.

Masum *et al.*[31] developed BAN-ABSA — a manually annotated high-quality Bengali dataset. A total of 9,009 comments from renowned Bengali online news sources are included in this BAN-ABSA dataset. A preliminary assessment using a deep learning model was carried out, and it was successful in classifying sentiment and extracting aspect terms with accuracy of 71.08% and 78.75%, respectively. Studies on the BAN-ABSA dataset reveal that the CNN model performs better in terms of accuracy, but the Bi-LSTM model performs much better in terms of average F1-score. With accuracy scores of 79.09% and 71.48%, respectively, CNN demonstrates greater accuracy in both of the subtasks.

Isaac *et al.*[29] performed an emotional analysis of people all across the world using data collected from twitter about the ongoing conflict in Ukraine. They have used the

"Ukraine Conflict Twitter Dataset" acquired from Kaggle version 24 in their research. They have used a RNN model with accuracy in the validation set reaching 93% and accuracy in the test set reaching 90%.

Wahid *et al.*[43] proposed a Recurrent Neural Network with an LSTM model for performing cricket sentiment analysis from Bangla texts. They constructed a dataset composed of 10,000 comments related to cricket comments in Bangla language. This dataset is an extended version of ABSA dataset (dataset containing cricket-related comments) from where only 2 of the columns (comment column and target column) were selected and rest of the data were collected from other online resources such as Facebook, YouTube, Prothom-Alo, BBC Bangla and Bdnews.com which were manually labeled. To identify the sentiments of people, the dataset is labeled with 3 categories: positive, negative, or neutral. Word embedding method was used for the vectorization purpose of each word and LSTM model was utilized for the achievement of long-term dependencies. Their proposed approach resulted in an accuracy score of 95%, which is higher than earlier proposed methods.

SentNoB, developed by Islam *et al.,*[23] is a dataset for assessing sentiment in noisy Bangla texts. This dataset consists of around 15,728 instances in total from social media comments on news articles and videos. The dataset is labeled by three polarities: positive, negative, and neutral, is assigned to one of these comments. By applying the dominant label (+ve) to every instance, the majority baseline surpasses the random baselines with a 41.24 F1 score (34.53 and 32.60). They have evaluated their dataset with several methods such as lexical feature analysis, RNN and pre-trained language models.

Romim *et al.*[37] built a Bangla Hate Speech dataset named BD-SHS which is titled as the largest Hate Speech dataset of Bangla language according to their knowledge of other available datasets. They searched contents through some keywords and also considered roasting videos and Bangla TikTok as sources of hate and toxic comments. For word embedding, informal embeddings (IFT) worked as feature extractors better than formal embeddings (BFT, MFT). Dataset was trained on SVM and Bi-LSTM models.

Kastrati *et al.*[25] presented a manually created dataset which contains comments collected from the official Facebook page of the NIPHK (National Institute of Public Health of Kosova) about the dissemination of the COVID-19 virus in the Republic of Kosova. This large-scale dataset is composed of 10,742 Facebook comments based on COVID-19 pandemic in a low-resource Albanian language. This study also proposed a deep learning-based Sentiment Analyzer named ALBANA and used it for validating their dataset. For capturing the semantics of words, an attention mechanism was utilized to distinguish between word level interactions. The experimental results show a high performance on their sentiment analysis task when a BiLSTM model was combined with an attention mechanism, resulting in an F1 score of 72.09%.

Nafis *et al.*[42] introduced deep learning-based models to detect multilabel sentiment and emotions from Bangla YouTube comments. Their created dataset consisted of 15,689 comments collected from various kinds of YouTube videos in Bangla, English, and Romanized Bangla. To classify a Bangla text, a 3-class: positive, negative, or neutral, and a 5-class: strongly positive, positive, neutral, negative, strongly negative, sentiment label is given. Also, this dataset contains 6 emotion labels: anger, disgust, fear, joy, sadness, and surprise. They as well developed an emotion analyzer to identify emotion of a Bangla text. Their proposed approach gave an accuracy of 65.97% in 3-class label sentiment and 54.24% in 5-class label sentiment. Based on their results, their model performance is much better for domain- and language-specific datasets.

Rahman *et al.*[35] published a Bangla dataset which was collected from BARD, OSBC and ProthomAlo. They used transformer-based models ELECTRA and BERT for classification of the Bangla documents. They measured the performance by calculating accuracy, recall, precision and F1 score. From the two models, ELECTRA achieved greater accuracy and F1 score than the BERT model for the data source of them.

Arid *et al.*[17] explored publicly available Bangla datasets and experimented with classical, deep neural networks (CNN, RNN) and large pre-trained models for sentiment analysis. Results in their work have indicated that finetuned BERT-based transformer models, XLM-Roberta in their case, outperform all the other models.

Samsul *et al.*[24] also explored the field of Bangla sentiment analysis. They have used data collected from several social media platforms and experimented with multiple approaches which include, feature extraction techniques (Word2Vec, Glove), the use of deep neural networks (RNN, LSTM, GRU), and lastly, transformer-based models (Bangla-BERT, mBERT). Their hybrid model (CNN-BiLSTM) has acheived an accuracy score of 88.59%.

3. Data Collection and Preprocessing

This section provides a thorough explanation of the chain of actions that were conducted to get the data, prepare the data, annotate the data and finally split the data into three sets for experimental evaluation of the dataset to develop a benchmark classifier for sentiment analysis.

3.1. *Data extraction from YouTube*

With over 2 billion monthly active users and more than 500 hours of content published every minute, YouTube is the second largest social network at the moment.[1] We chose YouTube as our primary source of data with the intention of ferreting out relevant user-generated comments which would ultimately be the building blocks of our dataset. With that goal in mind, it was convenient to select specific Bangladeshi TV news channels and extract user-generated comments from their YouTube channels. We wrote a Python script that would take a list of YouTube video IDs and extract comments from those videos with the help of the YouTube Data API. The entire data extraction process was automated except the part where we had to procure a list of YouTube video IDs manually. Our Python script was designed to take a list of YouTube video IDs and extract out different kinds of data such as username, reply count, date and most importantly comments. We sourced video IDs that were relevant to the ongoing Russia–Ukraine war from top three YouTube news channels of Bangladesh which include SOMOY TV (16.4M subscribers), Jamuna TV (11.8M subscribers) and Independent Television (7.41M subscribers). We extracted 42,911 Bangla comments from these channels alone. The channel's videos were uploaded between the time span of February to June.

3.2. *Data annotation*

We selected three annotators from our group to annotate the data. We finalized three class labels ***Pro-Ukraine***, ***Pro-Russia***, and ***Neutral***. In the context of Russia–Ukraine conflict, Pro-Russia sentiment was assigned for favoring the war with a polarity label of '**2**', Pro-Ukraine sentiment was assigned for against the war with a polarity label of '**1**', and Neutral sentiment was assigned for neutral opinion with a polarity label of '**0**'. Standing against Ukraine and supporting Russia was considered in favor of the war. Tables 2 and 3 present a tabular representation of annotated sample instances in our dataset and distribution of class labels, respectively.

In total, we annotated 10,861 comments out of which 5,011 data were Pro-Russia, 1,211 were Pro-Ukraine, and 4,639 were Neutral. From the percentage ratio, it can be observed that Bangladeshi public sentiment is biased towards Pro-Russia polarity and minimal towards Pro-Ukraine polarity. Our dataset has 12,760 distinct words in total.

Table 2. Annotated sample instances.

Annotation	Instance
Pro-Russia	[B]ইনশাআল্লাহ রাশিয়ার জয় হবে [E] (InshaAllah Russia will win)
Neutral	[B]ধ্বংস হোক অ্যামেরিকা [E] (May America perish)
Pro-Ukraine	[B]যুদ্ধ নয় শান্তি চাই [E](Want peace not war)

Table 3. Distribution of class labels.

Class	Instances
Pro-Russia	5,011 (46.1%)
Neutral	4,639 (42.7%)
Pro-Ukraine	1,211 (11.2%)
Total	10,861

3.3. *Data splitting*

After adequate data pre-processing, cleaning and data annotation, the final instance count of our dataset dropped down to 10,861. Our dataset suffers from class imbalance as shown in Table 3 where two majority classes are "Pro-Russia" (constitutes 46.14%) and "Neutral" (constitutes 42.71%). The minority class "Pro-Ukraine" only constitutes 11.15% of the entire dataset. A per class stratified split was done to establish train (60%), validation (20%), and test (20%) sets.

4. Methodology

In this section, we propose the idea of experimenting with various transfer learning classifiers. Notorious for their success in several NLP tasks such as text classification which is an umbrella term for sentiment analysis, multilingual pre-trained language models show superior performance compared to traditional machine learning approaches and deep neural networks.[4,27,28,34,39] Common machine learning algorithms used for sentiment analysis include: Maxent (Maximum Entropy), SVM (Support Vector Machine), Decision Tree, KNN (K-Nearest Neighbor), SGDC (Stochastic Gradient Descent Classifier), Random Forest classifier (RF), etc.[30] End-to-end deep neural networks such as Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), Bi-Directional Long Short Term Memory (Bi-LSTM) and Recurrent Neural Networks (RNN) and Gated Recurrent Units (GRU) were popular neural network architectures for the last several years in sentiment analysis tasks. However, the aforementioned models do have limitations in areas like contextual difference, failure to deal with out of vocabulary word embeddings, and the need of a large training corpus for better performance.[27]

On the other hand, all of such flaws that are found in machine learning or deep learning approaches are addressed using a transformer-based approach. Transfer learning is relatively a new concept in machine learning where knowledge gathered from one task is used to address other similar tasks. In the context of Bangla NLP, data are scarce and not adequate for training deep neural networks to achieve acceptable results. However, pre-trained language models leveraging transfer learning yield better results than classical machine learning or deep learning techniques.[4,34] For low-resource languages such as Bangla, multilingual pre-trained language models like, mBERT,[5] DistilBERT,[38] XLM-RoBERTa[13] after fine-tuning performs quite as good as their monolingual counterparts.

In this study, we assessed 5 distinct transfer learning classifiers to create a benchmark classifier for our own produced dataset. They are: mBERT, a distilled version of mBERT (DistilBERT-base-multilingual-cased), XLM-RoBERTa base, XLM-RoBERTa large, BanglaBERT[8] (with a fine-tuned classification head). Figure 1 illustrates our entire project workflow. In Fig. 2, a transformer model architecture illustrates the procedure from input sentence to classification. The

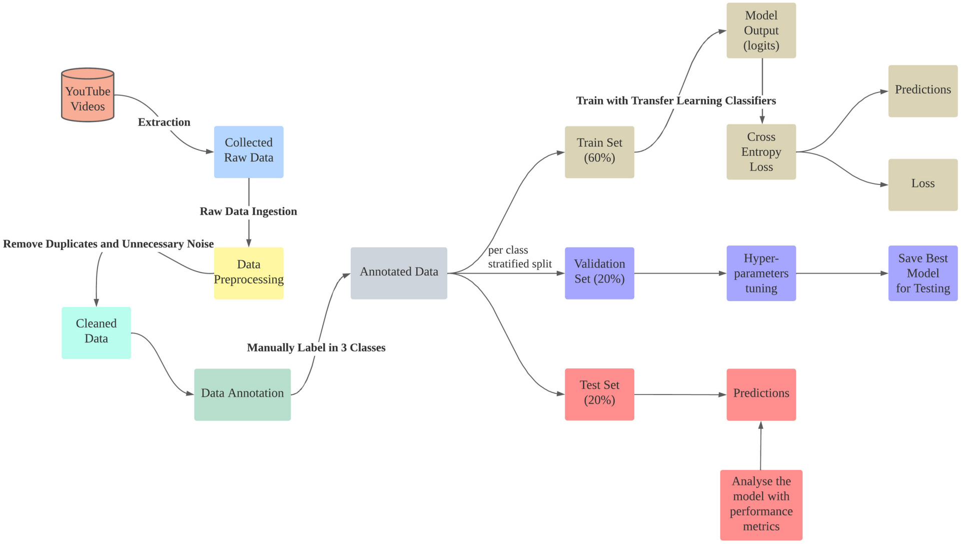


Fig. 1. Project workflow.

aforementioned pre-trained language models are publicly available in Hugging Face's transformer library.[44] In addition, to give our experiments a better perspective we used Bi-LSTM[20] recurrent neural network (RNN) as our baseline performance.

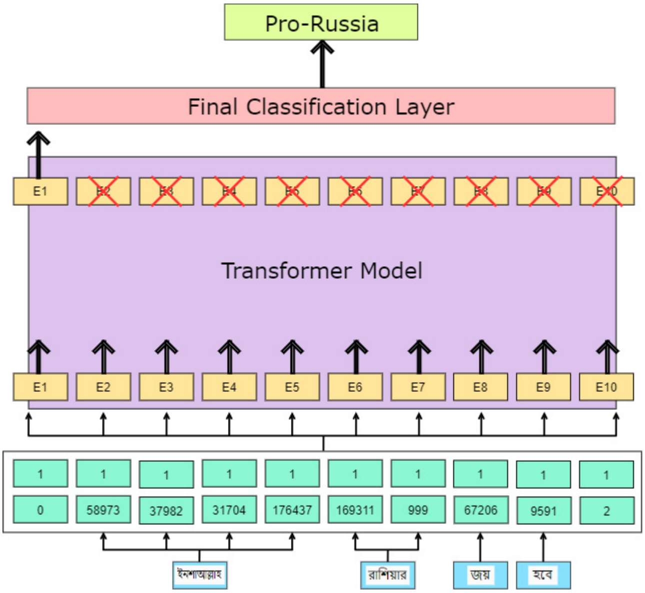


Fig. 2. Transformer model architecture.

4.1. *Types of classifiers used*

4.1.1. *Multilingual BERT (bert-base-multilingual-cased)*

Bidirectional Encoder Representation from Transformers (BERT) proposed by Jacob Devline *et al.*[5] introduced a new language representation model that is designed and trained on bidirectional representations from unlabeled datasets. The pretraining is done by combining left and right contexts of tokens in all the layers. Two training tasks were adopted: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) to pre-train the model in an unsupervised technique. Pretraining is computationally a heavy task as it requires the model to be trained on huge amounts of data. Fortunately, the BERT needs to be pre-trained only once and can be fine-tuned to perform different task specific tasks. BERT can be fine-tuned by adding an additional output layer for any downstream tasks such as question answering and text classification.[5]

We used the multilingual version of BERT, namely the model checkpoint *bert-base-multilingual-cased* which was loaded from the Hugging Face library. This model was pre-trained on 104 languages with a corpus that is the largest in Wikipedia. Multilingual BERT has a total of 177M trainable parameters. The important thing about multilingual BERT is that the data it is trained on are sampled by exponential smoothing. This results in under-sampling of high-resource languages like English which gives precedence to low-resource languages such as Bangla.

4.1.2. *DistilBERT (DistilBERT-base-multilingual-cased)*

Our experiment included a distilled version of the multilingual variant of BERT, namely the checkpoint, *distilbert-case-multilingual-cased*. Distilbert was first introduced by Victor *et al.*[38] and was asserted that it was smaller, faster, cheaper and lighter than its larger counterpart. The authors of this model, leveraged the idea of knowledge distillation in the pre-training phase and managed to reduce the size down to 40% while still retaining 97% of its language understanding capabilities.[38] After the distillation process, the total number of parameters are lowered from 177M to 135M and thus the training process is 60% faster.

4.1.3. *XLM-RoBERTa base*

XLM-RoBERTa is the multilingual variant of RoBERTa. The dataset it was pre-trained on included 100 languages and a total of 2.5TB of filtered CommonCrawl data.[13] It has a total of 278M parameters with a 12-layer, 768 hidden state architecture.

The authors of XLM-RoBERTa stated in their paper that, *XLM-RoBERTa* outperforms *mBERT* by upto 23% increased accuracy achieved in cross-lingual classification on low-resource languages.[13] XLM-RoBERTa shows impressive performance in multilingual downstream tasks like sentiment classification with good performance metrics. In our experiment, we used XLM-RoBERTa-base from the Hugging Face transformers library.

4.1.4. *XLM-RoBERTa large*

XLM-RoBERTa-large is the larger version of XLM-RoBERTa-base. It has approximately 560M of trainable parameters with a 24-layer, 1024 hidden states architecture. The authors had to train this large model on 500 Nvidia V100 GPUs each having 32GB of dedicated VRAM with a batch size of 8192.[13] Unlike its smaller counterpart, XLM-RoBERTa large takes significant computational resources and time to train our dataset. However, it achieves the highest accuracy and macro F1 score than the other transformer classifiers we used in our experiment.

4.1.5. *BanglaBERT*

BanglaBERT is a BERT-based transformer model specifically pre-trained on 27.5GB of Bangla texts.[8] BanglaBERT has the lowest trainable parameters of all the transformer models we used in our experiments. It has a parameter count of approximately 110M. Unlike BERT where it is trained with a MLM objective where the input gets replaced with mask tokens and the language model essentially reconstructs them to its original form through training, BanglaBERT is trained with an RTD objective introduced in Ref. 12 due to the fact that it is more sample-efficient than MLM objective which requires large data. BanglaBERT is distinctive in nature compared to other BERT-based models primarily for its implementation of Replaced Token Detection (RTD) approach. Excluding this distinctive pre-training objective, the rest of the architecture is similar to the standard BERT architecture. In the RTD approach, two neural networks namely, Generator and Discriminator are trained where the generator model follows standard MLM objective where a certain portion (15%) of the input text is masked-out and generates predictions by training. Next, using the generator's output distributions, the masked out tokens are corrupted (replaced) with reasonable and logical alternative tokens. The discriminator model then tries to predict which of the tokens were original and corrupted. In the replaced token detection approach (RTD), the backpropagation is done with all the tokens compared to the MLM approach where 15% of input sequence is masked out. For this reason, ELECTRA[12] is computationally less expensive and learns better than BERT model's MLM approach because it needs fewer samples to train. For this reason the authors of BanglaBERT chose the ELECTRA approach rather than the standard MLM pre-training approach used by BERT.

4.1.6. *Bi-directional long short term memory*

Bi-LSTM model[20] serves as our baseline classifier in our experiments. Bi-directional long short term memory is a recurrent neural network that is designed to encode an input text from both forward and backward directions thus, remembering sequence information of both directions. The encoding takes from both directions and results in two 2-dimensional (2D) embedding vectors which are later concatenated. Next, the Bahdanau attention mechanism[7] is applied to the concatenated 2D vector that is able to automatically search selective parts of an input text and learn to apply adequate weight that is needed for predicting words. Finally, the attention applied vectors are summed up and passed to a linear layer to predict 3 categorical classes.

4.2. *Training procedure*

All the models were trained and validated on the train and validation sets where optimization of the parameters only took place in the learning phase. On unseen data from the test set, only the top model was evaluated. The various hyper parameters and optimization steps that were experimented with are shown in Table 4. Cross entropy loss criterion was used to compute loss for each of the 5 classifiers used in the experiments. We used Adam optimizer with a linear warmup as the optimization algorithm for training the parameters and tuning the learning rate of the models as it has excellent empirical performance in deep learning.[26] Total of 8 epochs and batch size 32 and 16 were used in the experiments except for the baseline LSTM model which we trained for 35 epochs. Nvidia Tesla 12GB K80 GPU was used to train all the models used in the experiments. As suggested by the authors of BERT[5] three different learning rates were used: 5e-5, 2e-5 and 3e-5. Hyperparameter tuning was done with combinations of batch sizes and learning rates. Early stopping was applied to prevent overfitting the models in the experiments. We analyzed the distribution of lengths of tokenized texts of our dataset and perceived that the distribution ranges from 3 to 357. For encoding string inputs to model inputs, we used *AutoTokenizer* generic class which instantiates model-specific tokenizer classes as per the given model id. While encoding the input texts, maximum token lengths of 128 and 200 were fixed for certain models, truncation and padding were applied when necessary. For fine tuning our models a generic class, *AutoModelForSequenceClassification* was used which instantiates a specific model architecture with a classification head on top of the output head of the model as per the given model id. During training all model parameters were optimized. From Fig. 3, we get an overview of the training time each model took, where the elapsed training times are on the y-axis, and each of the models used in our experiments is on the x-axis. XLM-RoBERTa-large, being the largest model in our experiments expectedly takes longer time to train. In addition,

Table 4. Network training parameters of transformer models.

Network parameters, optimizers and criterion	Value
Optimizer	Adam
Criterion	Cross Entropy Loss
Learning rate	2e-5, 3e-5, 5e-5
Dropout rate	0.1
Training epoch	8
Early stopping	Yes
Batch size	16, 32
Training, Evaluation, Testing	60%, 20%, 20%
Weight decay	1e-2
Maximum length	128, 200

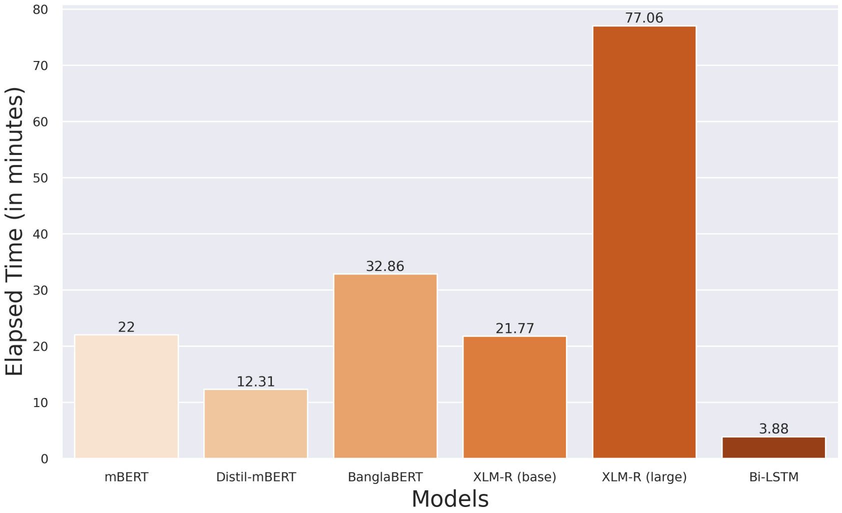


Fig. 3. Elapsed time (in minutes) during each training run of the models.

training XLM-RoBERTa-large with a mini batch size of 32 was not possible for us due to hardware constraints.

5. Experimental Results and Discussion

5.1. *Evaluation metrics*

Several evaluation metrics are defined that have been used to analyze the model performance in this study. For better identification of the performance of model, accuracy, F1 score, precision, recall, all are used in most classification tasks .[16,21,23,31,35,37]

5.1.1. *Cross entropy loss*

All the classifiers in our experiments were trained with the cross entropy loss criterion. For tasks such as multi-class sentiment classification, it is useful applying a cross entropy loss.[19] Each output z_i from each individual class from the output layer of the network is then passed into a softmax activation function. The softmax activation[33] returns a probability distribution by dividing the exponent of output $e^{(z_i)}$ with the summation of exponent of the outputs $e^{(z_j)}$ of all the classes. The output of the softmax is then sent as input to the cross entropy criterion. In Eq. (2), y_c is the truth label, p_c is the probability distribution from the softmax activation and C is the total number of classes, which in our case is 3. Therefore, cross entropy is often called Softmax Loss due to the fact that it is combined with a softmax activation. Equations (1) and (2) represent the mathematical representations of softmax activation and cross entropy loss, respectively,

$$\sigma(\bar{z}) = \frac{e^{(z_i)}}{\sum_{j=1}^{k} e^{(z_j)}}, \tag{1}$$

$$\text{CE} = -\sum_{c=1}^{C} y_c \log(p_c). \tag{2}$$

5.1.2. *Accuracy*

Accuracy in a classification task is defined as the proportion of instances that were properly predicted over all observations. The mathematical representation of accuracy is presented in Eq. (3). Moreover, TP, TN, FP and FN refer to the number of true positives, true negatives, false positives and false negatives. Accuracy is a measure for balanced dataset but as our dataset suffers from class imbalance issue, we did not consider it as the principal performance metric.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{3}$$

5.1.3. *Precision and recall*

The proportion of accurately categorized occurrences to all instances that were identified is known as precision. For a multiclass classification problem such as ours, a high precision score for Negative class refers to the number of instances that were correctly characterized as Negatives over the total Negative classified instances. Recall on the other hand refers to the ratio of correctly predicted instances over the total number of observations of that particular class. High recall score denotes the ability of classifying a particular class is high. The mathematical representation of precision and recall is shown in Eqs. (4) and (5).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5}$$

5.1.4. *Macro averaged F1-score*

Since our dataset is unbalanced, macro-F1 score is used as our principal evaluation metric. It is because macro F1 gives all the classes equal importance. F1 score for a particular class follows Eq. (6).

$$\text{F1 score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{6}$$

5.1.5. *Akaike information criterion*

Akaike information criterion (AIC) is mostly used for model selection. AIC is a statistical function introduced by Ref. 41 that estimates the out-of-sample prediction error. A lower value of AIC indicates that a model is best fit for the model. We used the corrected Akaike information criterion (AICc) to compare the model performances obtained from the analysis of macro-F1 score and accuracy of each model with the value of AIC calculated for each model. The mathematical representation for calculating AIC value is shown in Eq. (7).

$$\mathrm{AIC}_C = n \ln \frac{1}{n}\left(\sum_{i=1}^{n} (\mathrm{pred} - \mathrm{actual})^2\right) + 2k + \frac{2k(k+1)}{n-k-1}. \tag{7}$$

5.2. *Results*

Five transfer learning classifiers: Multilingual BERT (mBERT), Distil Multilingual BERT (Distil-mBERT), BanglaBERT, XLM-RoBERTa base and XLM-RoBERTa large were experimented with three different learning rates, namely 3e-5, 2e-5 and 5e-5 with batch sizes of 16 and 32. Several experiments were conducted with the combinations of learning rate and batch size hyperparameters with the models to selectively ferret out the best 5 models based on their macro F1 score and accuracy and AIC score. The detailed results of the experiments with combinations of 16 and 32 batch sizes with 3 different learning rates applied in 5 transfer learning classifiers are shown in Table 5. From Table 5, we observed that certain models gave better results with certain combinations of batch size and learning rate. This behavior is expected because transformer-based classifiers are known to be highly sensitive to learning rates and batch sizes, mainly due to the fact that these models have been pre-trained with large corpuses. Considering the hyperparameter batch size, it plays an important role in all 5 models in our experiments. Distil-mBERT, XLM-RoBERTa-large and mBERT performed best when batch size is 16. Again, XLM-RoBERTa-base and BanglaBERT showed best performance individually coupled with a batch size of 32. Learning rate is also a very significant hyperparameter, especially for transformer-based models as they are very sensitive to it. Interestingly, Distil-mBERT, XLM-RoBERTa-large and XLM-RoBERTa-base showed best performance with a learning rate of 3e-5. Our experiment showed that XLM-RoBERTa responds well to 3e-5 learning rate. Multilingual BERT (mBERT) and the overall best-performing model, BanglaBERT performed their individual best performance with a learning rate of 5e-5. As mentioned earlier, it was not possible due to hardware constraints to train XLM-RoBERTa-large with batch size of 32 thus, this particular experiment was not included in the results. Additionally, Bi-LSTM model was used as a baseline performance to evaluate our dataset. However, no hyperparameter tuning was performed in the case of Bi-LSTM. It was trained with a batch size of 256 and a learning rate of 0.01.

A detailed evaluation of the best-performing models after hyper parameter tuning along with the baseline model is shown with their class-wise precision, recall and F1 score in Table 6. BanglaBERT is the best-performing model out of all the 6 classifiers with a macro F1 score of 0.82 and accuracy of 86%. From Figs. 5 and 6, relative

Table 5. Comparative analysis of 16 and 32 batch sizes with 3 different learning rates on test set. (Models with best performance are highlighted with bold font.)

Batch size	Learning rate	Model	F1-Score	Accuracy
16	2e-5	mBERT	0.73	80%
		Distil-mBERT	0.73	80%
		BanglaBERT	0.78	83%
		XLM-RoBERTa base	0.80	84%
		XLM-RoBERTa large	0.21	45%
	3e-5	mBERT	0.74	80%
		Distil-mBERT	**0.76**	**81%**
		BanglaBERT	0.80	83%
		XLM-RoBERTa base	0.78	83%
		XLM-RoBERTa large	**0.79**	**82%**
	5e-5	**mBERT**	**0.77**	**82%**
		Distil-mBERT	0.73	80%
		BanglaBERT	0.82	85%
		XLM-RoBERTa base	0.76	82%
		XLM-RoBERTa large	0.21	45%
32	2e-5	mBERT	0.76	81%
		Distil-mBERT	0.70	78%
		BanglaBERT	0.76	82%
		XLM-RoBERTa base	0.77	82%
		XLM-RoBERTa large	N/A	N/A
	3e-5	mBERT	0.74	80%
		Distil-mBERT	0.71	79%
		BanglaBERT	0.79	83%
		XLM-RoBERTa base	**0.81**	**84%**
		XLM-RoBERTa large	N/A	N/A
	5e-5	mBERT	0.70	76%
		Distil-mBERT	0.72	79%
		BanglaBERT	**0.82**	**86%**
		XLM-RoBERTa base	0.78	83%
		XLM-RoBERTa large	N/A	N/A
256	0.01	Bi-LSTM	0.64	74%

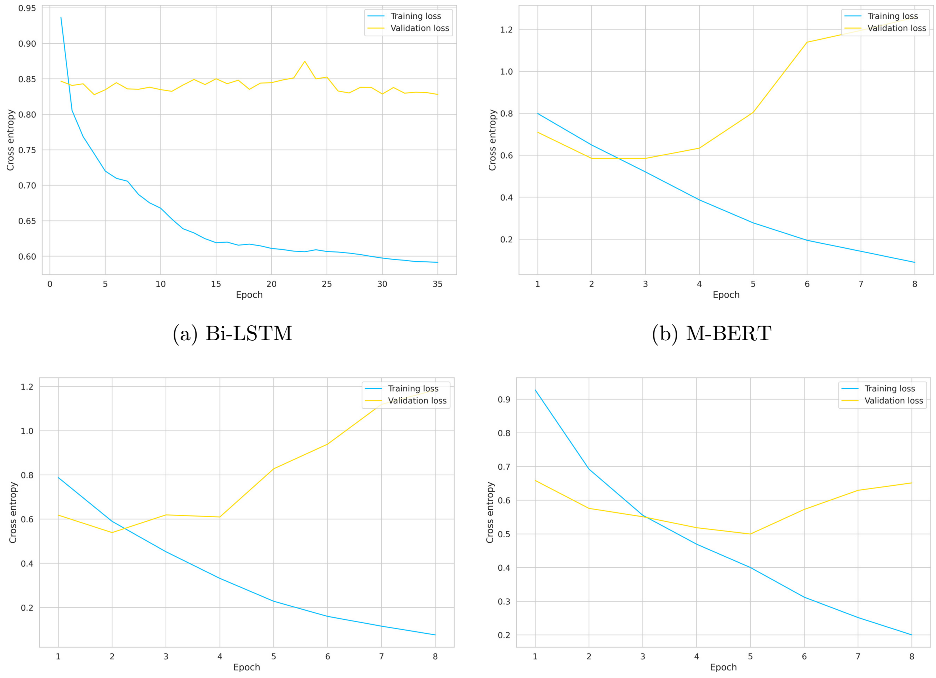
Table 6. Comparative analysis of the best-performing models after hyper parameter tuning. Support (S), Precision (P), Recall (R) and F1 score is listed for each sentiment polarity. (Model with best performance is highlighted with bold font.)

Batch size	Learning rate	Models	Class labels	Support	Precision	Recall	F1	Macro-F1	Accuracy
16	5e-5	mBERT	neutral	1001	0.85	0.84	0.84	0.77	82%
			positive	220	0.67	0.60	0.63		
			negative	952	0.81	0.84	0.83		
16	3e-5	Distil-mBERT	neutral	1001	0.83	0.84	0.83	0.76	81%
			positive	220	0.70	0.58	0.63		
			negative	952	0.81	0.82	0.82		

Table 6. (*Continued*)

Batch size	Learning rate	Models	Class labels	Support	Precision	Recall	F1	Macro-F1	Accuracy
32	**5e-5**	Bangla BERT	neutral	1001	0.85	0.89	0.87	**0.82**	**86%**
			positive	220	0.72	0.72	0.72		
			negative	952	0.90	0.86	0.88		
32	3e-5	XLM-R (base)	neutral	1001	0.88	0.84	0.86	0.81	84%
			positive	220	0.69	0.70	0.69		
			negative	952	0.84	0.88	0.86		
16	3e-5	XLM-R (large)	neutral	1001	0.85	0.82	0.84	0.79	82%
			positive	220	0.71	0.68	0.70		
			negative	952	0.82	0.86	0.84		
256	0.01	Bi-LSTM	neutral	849	0.73	0.79	0.76	0.64	74%
			positive	232	0.55	0.29	0.38		
			negative	967	0.77	0.81	0.79		

macro F1 scores and accuracies of the transformer models and baseline classifier are shown in detailed view. BanglaBERT outperforming other transformer models and baseline Bi-LSTM was expected due to the fact that it was pre-trained on large amounts of Bangla texts (27.5GB) whereas other classifiers were not so much trained



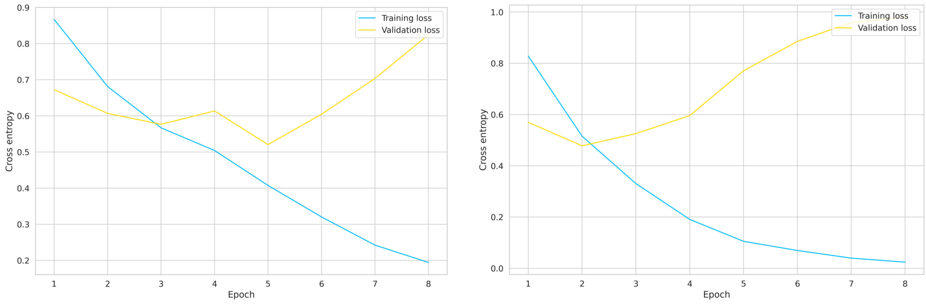
(a) Bi-LSTM

(b) M-BERT

(c) Distil-M-BERT

(d) XLM-RoBERTa-base

Fig. 4. Loss curves of the models.



(e) XLM-RoBERTa-large

(f) BanglaBERT

Fig. 4. (*Continued*)

on the Bangla language. XLM-RoBERTa-base performs close to BanglaBERT with 0.81 F1 and 84% of accuracy. XLM-RoBERTa at its base version is notable for being good at classification tasks and was reported to done well in low-resource languages when fine-tuned properly. In our experiment baseline Bi-LSTM achieves 0.64 F1 score with no pre-training. This indicates that our baseline requires more data to project acceptable results. Distil-mBERT and mBERT show near similar results although Distil-mBERT is comparatively much faster and computationally cheaper than mBERT. XLM-RoBERTa-large performs the lowest out of all the transformers with 79 macro F1 score despite being the deepest model out of all. In our

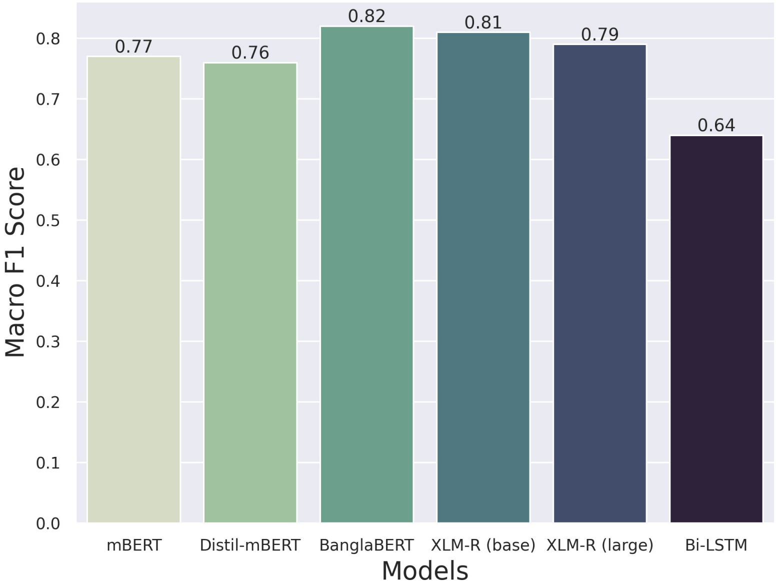


Fig. 5. Relative Macro-F1 score of each model.

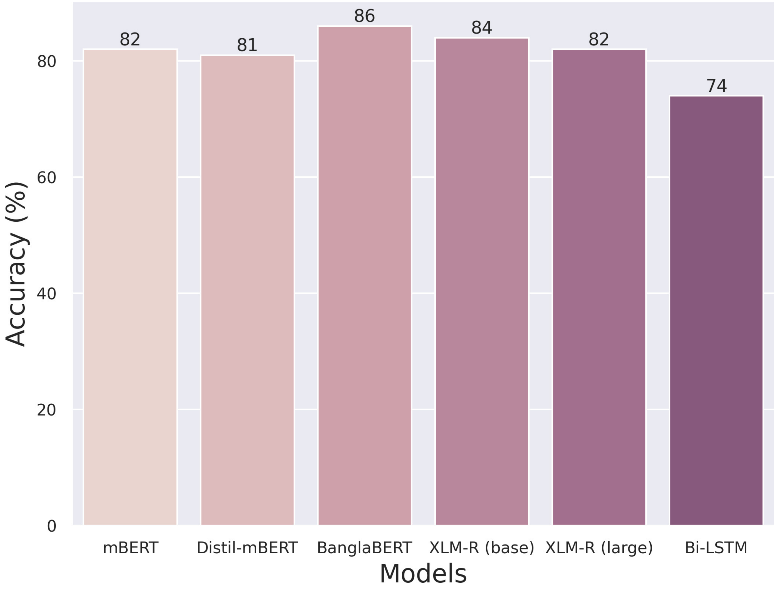


Fig. 6. Relative Accuracy of each model.

experiments, we perceived that the XLM-RoBERTa-large needs to be trained on larger epochs to reach the levels of other models, however that might lead to overfitting.

In Fig. 4, the respective loss graphs of the models are shown. By observing the loss curves of each of the models used in the experiments, we see a similar pattern of when these models start to overfit. Each model was trained on 8 epochs and the loss graphs of the models show a number of models starting to overfit particularly at epochs 2 and 3. Therefore, early stopping was applied to prevent these models from overfitting.

Furthermore, we evaluated each transformer model with the corrected Akaike Information Criterion[41] to select the model that best fit our dataset. The evaluation of AIC is shown in Table 7. According to Table 7, the smallest value obtained for AIC is given by BanglaBERT. From these analyses, it can be observed that Bangla BERT is the best-performing model with an AIC value of −6295.59. The smaller the

Table 7. AIC value of each model. (Model with best performance is highlighted with bold font.)

Model	AIC
mBERT	−5774.66
Distil-mBERT	−5572.14
BanglaBERT	**−6295.59**
XLM-R (base)	−6157.75
XLM-R (large)	−5728.80
Bi-LSTM	−5183.28

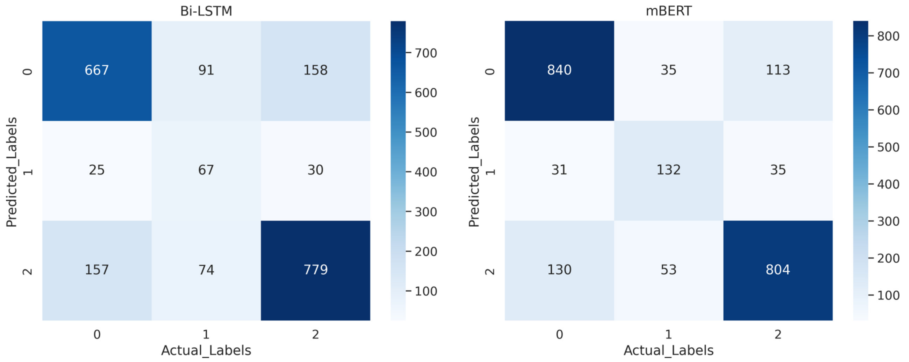
value of AIC, the better the model fits the data. Therefore, we conclude that according to the macro-F1 score and the AIC, BanglaBERT is the best-performing classifier for our dataset. Our work also highlights this fact that compared to other approaches such as feature extraction, deep neural networks (RNN, LSTM, GRU) or hybrid models like CNN-BiLSTM [17,21,23,24,29,31,42,43] that have been reported so far in the last few years, transformer-based large language models, in our case BanglaBERT has performed exceptionally well in Bangla sentiment analysis.

5.3. *Error analysis*

Our experimental results have shown that BanglaBERT is the best-performing model out of all the transformer models and is selected as our benchmark classifier for evaluating our dataset. In this section, we apply a thorough error-analysis of the best-performing model to find out its drawbacks and mistakes. Furthermore, both quantitative and qualitative analysis are discussed in this section to have a better understanding of the model. Figure 7 represents the confusion matrix of all the models that were experimented with. From Fig. 8, we can see the class-wise confusion matrix of BanglaBERT.

5.3.1. *Quantitative analysis*

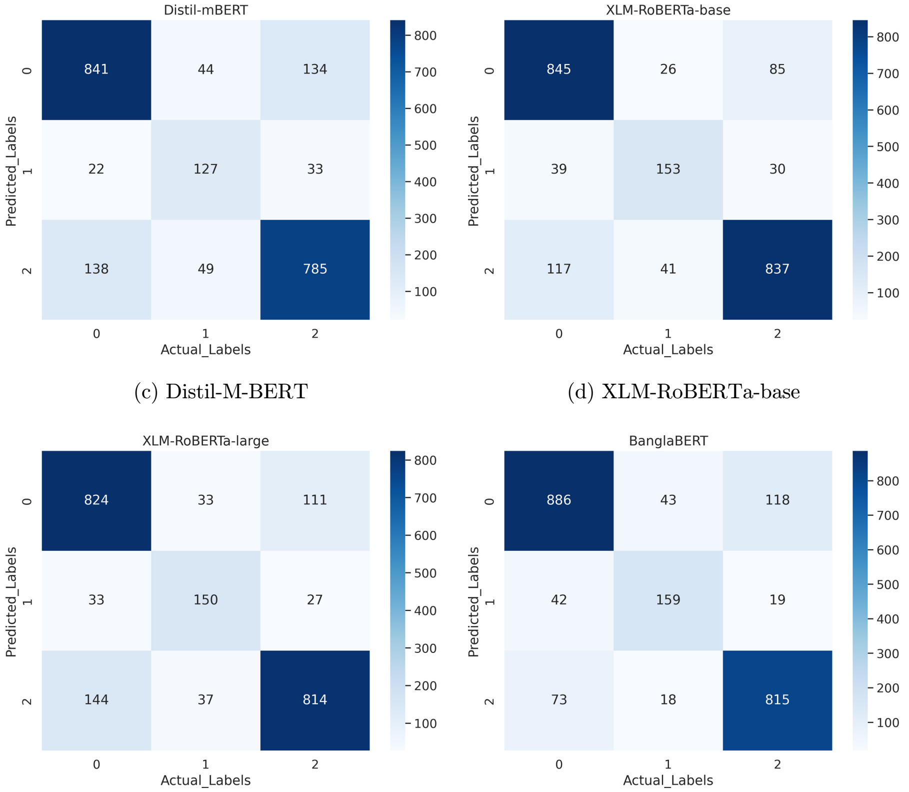
It is observed that, for the Neutral class BanglaBERT misclassified 161 (out of 1,047) and 115 (out of 1,126) instances. For the Pro-Ukraine class, it mistakenly classified 61 (out of 220) and 61 (out of 1,953) instances, respectively. Furthermore, it falsely predicted 91 (out of 906) and 137 (out of 1,267) comments. Thus, it is evident that the number of incorrect predictions are greater when predicting comments that are Pro-Ukraine (61 out of 220). This low performance regarding positive instances can be attributed to the fact that the number of comments labeled pro-Ukraine in our dataset is less (11.4%) compared to pro-Russia (46.4%) and neutral classes (42.2%).



(a) Bi-LSTM

(b) M-BERT

Fig. 7. Confusion matrices of each model.



(c) Distil-M-BERT

(d) XLM-RoBERTa-base

(e) XLM-RoBERTa-large

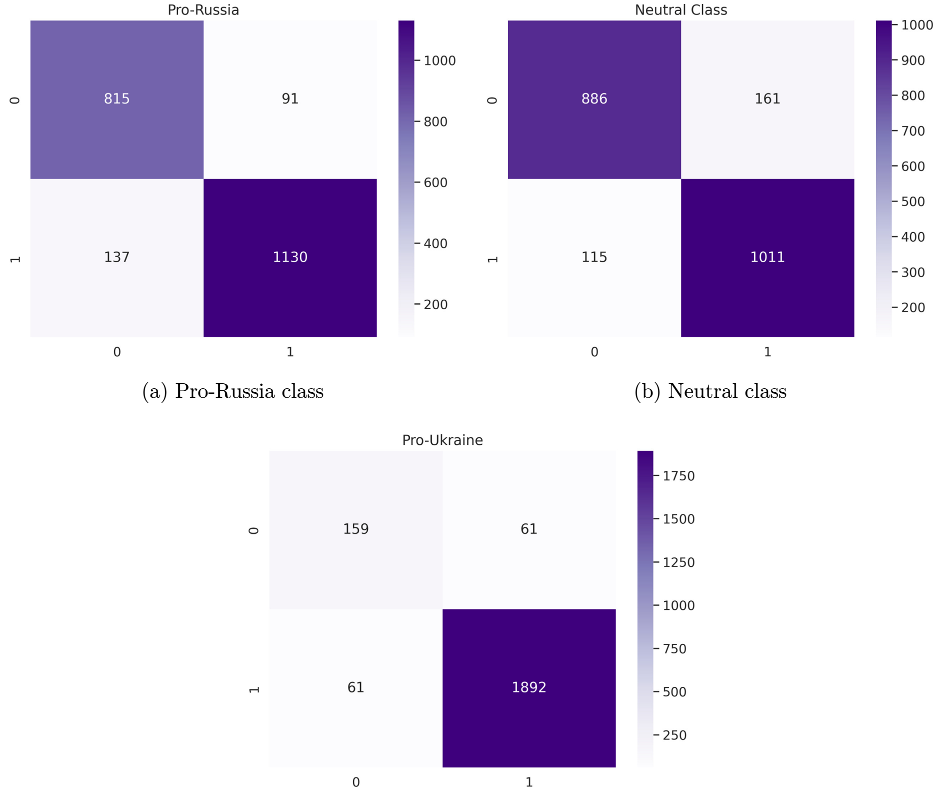
(f) BanglaBERT

Fig. 7. (*Continued*)

Furthermore, from our analysis we found that there is a high 27.8% FNR (False Negative Rate) related to the Pro-Ukraine class compared to other classes that have a FNR rate of 11.4% (Neutral) and 14.3% (Pro-Russia). This is also due to the fact that our dataset suffers from class imbalance. Therefore, it is perceived that the model is likely to falsely predict any Pro-Ukraine comments 27.8% of the time. To counter this problem, adding more training samples of Pro-Ukraine class is the ideal approach.

5.3.2. *Qualitative analysis*

Table 8 shows four prediction examples comprising both correct and incorrect classifications obtained from BanglaBERT. The first two examples are correctly classified by the model and the later two are misclassifications. The last two examples illustrate that the model incorrectly identifies pro-Ukraine (Positive) comments as neutral comments. However, in the second example it correctly classifies a



(a) Pro-Russia class

(b) Neutral class

(c) Pro-Ukraine class

Fig. 8. Class-wise confusion matrices of BanglaBERT.

pro-Ukraine comment. From our analysis, the reason for these false predictions stems from having less number of training examples of the pro-Ukraine class. Our analysis also finds that comments that do not explicitly express a certain sentiment but show implicit expression of that particular sentiment are misclassified. For example, the

Table 8. Examples of correctly classified and misclassified predictions of BanglaBERT model.

Text	Actual	Prediction
'শাবাশ রাশিয়া শাবাশ পুতিন সাহেব কে এগিয়ে যান শুভকামনা রইল রাশিয়ার জন্য' (Bravo Russia bravo to Putin sir best wishes for Russia)	Pro-Russia	Pro-Russia
'আল্লাহ পাক আমাদের সবাই কে হেফাজত করুন আমরা যুদ্ধ চাই না শান্তি চাই' (May God protect us all we do not want war we want peace)	Pro-Ukraine	Pro-Russia
'রাশিয়ার যুদ্ধের কারণে ইউক্রেনে সাধারণ মানুষ নিহত আহত হচ্ছে' (Because of Russia's war innocent people in Ukraine are getting killed and wounded)	Pro-Ukraine	Neutral
'পৃথিবীর উচিত ইউক্রেনের সৈন্যদের থেকে স্বাধীনতা রক্ষার লড়াই শিখা' (The world should learn to fight from the soldiers of Ukraine to protect its independence)	Pro-Ukraine	Neutral

Table 9. An example of correct prediction of BanglaBERT model and misprediction of other models.

Text	Actual	BanglaBERT	XLM-R (base)	XLM-R (large)	M-BERT	Distil-MBERT
'খুব ভালো খবর রাশিয়া কে পৃথিবীর মানচিত্র থেকে মুছে ফেলা উচিত' (Very good news Russia should be wiped off the world map)	Pro-Ukraine	Pro-Ukraine	Pro-Russia	Pro-Russia	Pro-Russia	Pro-Russia

latter two examples shown in Table 8 exhibit pro-Ukraine sentiments but not in a clear manner thus, making it harder for the model to predict correctly. However, in some instances BanglaBERT excels all the transformer models in predicting the correct class. To illustrate, in Table 9, an example is shown where only BanglaBERT correctly classified a comment when all the other models misclassified it. This indicates that adding more training examples of pro-Ukraine class would enhance the predicting ability of BanglaBERT and partially eradicate this underlying problem.

6. Conclusion and Future Work

Sentiment analysis is the procedure of understanding and acquiring details of public opinions, their attitudes, responses and expressions towards a particular situation or incident presented in the form of text. Being one of the most extensively spoken languages worldwide, Bangla needs to be more valued and researched in this area of sentiment analysis. Unfortunately, there is a shortage of Bangla datasets that are essential for research in Bangla natural language processing. Therefore, we created a dataset composed of Bangla comments about the ongoing Russia–Ukraine war for sentiment analysis in this domain. In addition, we developed a benchmark classifier by utilizing our dataset that can perform sentiment analysis in the context of the Russia–Ukraine war. We are making our own dataset and models publicly available at https://github.com/mahmudhasankhan/Bangla-SA-On-Russia-Ukraine-War for advancement in Bangla NLP.

In future, we plan to enhance the quality of our dataset by eradicating the class imbalance issue it has so that it can be more useful to fellow researchers in the natural language processing community. It is hoped that, adding more training samples in the dataset would produce models with better generalization capability.

Acknowledgment

This work was supported by the Faculty Research Grant [CTRG-22-SEPS-07], North South University, Bashundhara, Dhaka 1229, Bangladesh.

References

1. Youtube for press, https://blog.youtube/press/.
2. A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, Sentiment analysis of twitter data, in *Proc. Workshop on Languages in Social Media* (ACL, 2011), pp. 30–38.
3. F. Alam, A. Hasan, T. Alam, A. Khan, J. Tajrin, N. Khan and S. A. Chowdhury, A review of bangla natural language processing tasks and the utility of transformer models, preprint (2021), arXiv:2107.03844.
4. T. Alam, A. Khan and F. Alam, Bangla text classification using transformers, preprint (2020), arXiv:2011.04446.
5. T. T. Aurpa, R. Sadik and M. S. Ahmed, Abusive bangla comments detection on facebook using transformer-based deep learning models, *Soc. Netw. Anal. Min.* **12** (2022) 24, doi: 10.1007/s13278-021-00852-x.
6. S. Baccianella, A. Esuli and F. Sebastiani, Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in *Proc. Seventh Int. Conf. Language Resources and Evaluation (LREC 2010)*, Vol. **10** (European Language Resources Association, 2010), pp. 2200–2204.
7. D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, preprint (2014), arXiv:1409.0473.
8. A. Bhattacharjee, T. Hasan, K. S. Mubasshir, M. Rahman, A. Iqbal and R. Shahriyar, Banglabert: Combating embedding barrier for low-resource language understanding, (2021), https://arxiv.org/abs/2101.00204.
9. U. Bhaumik, D. K. Yadav, Sentiment analysis using twitter. In: J.K. Mandal, I. Mukherjee, S. Bakshi, S. Chatterji, P. K. Sa (eds.) *Computational Intelligence and Machine Learning*, pp. 59–66. Springer Singapore, Singapore (2021).
10. V. Bobicev, O. Kanishcheva and O. Cherednichenko, Sentiment analysis in the Ukrainian and Russian news, in *Proc. 1st Ukraine Conf. Electrical and Computer Engineering (UKRCON - 2017)* (IEEE, 2017), pp. 1050–1055, doi: 10.1109/UKRCON.2017.8100410.
11. Y. Chandra and A. Jana, Sentiment analysis using machine learning and deep learning, in *2020 7th Int. Conf. Computing for Sustainable Global Development (INDIACom)* (IEEE, 2020), pp. 1–4, doi: 10.23919/INDIACom49435.2020.9083703.
12. K. Clark, M. T. Luong, Q. V. Le and C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, preprint (2020), arXiv:2003.10555.
13. A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzman, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in *Proc. 58th Annual Meeting of the Association for Computational Linguistics* (ACL, 2020), pp. 8440–8451, doi: 10.18653/v1/2020.acl-main.747.
14. G. Gautam and D. Yadav, Sentiment analysis of twitter data using machine learning approaches and semantic analysis, in *2014 Seventh Int. Conf. Contemporary Computing (IC3)* (IEEE, 2014), pp. 437–442, doi: 10.1109/IC3.2014.6897213.
15. A. Go, R. Bhayani and L. Huang, Twitter sentiment classification using distant supervision, *Processing* **150** (2009) 1–6.
16. A. Hande, R. Priyadharshini and B. R. Chakravarthi, Kancmd: Kannada codemixed dataset for sentiment analysis and offensive language detection, in *Proc. Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media* (ACL, 2020), pp. 54–63.
17. M. A. Hasan, J. Tajrin, S. A. Chowdhury and F. Alam, Sentiment classification in bangla textual content: A comparative study, in *2020 23rd Int. Conf. Computer and Information Technology (ICCIT)* (IEEE, 2020), pp. 1–6, doi: 10.1109/ICCIT51783.2020.9392681.

18. A. Hassan, N. Mohammed and A. Azad, Sentiment analysis on bangla and romanized bangla text (BRBT) using deep recurrent models, preprint (2016), arXiv:1610.00369.
19. Y. Ho and S. Wookey, The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling, *IEEE Access* **8** (2019) 4806–4813, doi: 10.1109/ACCESS.2019.2962617.
20. S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9**(8) (1997) 1735–1780.
21. M. Hoq, P. Haque and M. N. Uddin, Sentiment analysis of bangla language using deep learning approaches, in *Computing Science, Communication and Security. COMS2 2021* (Springer, 2021), pp. 140–151, doi: 10.1007/978-3-030-76776-1_10.
22. K. Islam, M. S. Islam and M. Amin, Sentiment analysis in bengali via transfer learning using multi-lingual BERT, in *2020 23rd Int. Conf. Computer and Information Technology (ICCIT)* (IEEE, 2020), pp. 1–5, doi: 10.1109/ICCIT51783.2020.9392653.
23. K. Islam, M. S. Islam and M. Amin, Sentnob: A dataset for analysing sentiment on noisy bangla texts, in *Findings of the Association for Computational Linguistics: EMNLP 2021* (ACL, 2021), pp. 3265–3271, doi: 10.18653/v1/2021.findings-emnlp.278.
24. S. Islam, M. J. Islam, M. M. Hasan, S. M. S. M. Ayon and S. S. Hasan, Bengali social media post sentiment analysis using deep learning and bert model, in *2022 IEEE Symp. Industrial Electronics & Applications (ISIEA)* (IEEE, 2022), pp. 1–6, doi: 10.1109/ISIEA54517.2022.9873680.
25. Z. Kastrati, L. Ahmedi, A. Kurti, F. Kadriu, D. Murtezaj and F. Gashi, A deep learning sentiment analyser for social media comments in low-resource languages, *Electronics* **10** (2021) 1133, doi: 10.3390/electronics10101133.
26. D. Kingma and J. Ba, Adam: A method for stochastic optimization, in *3rd International Conference on Learning Representations, ICLR 2015* (San Diego, CA, USA, May 7–9, 2015).
27. S. Kokab, S. Asghar and S. Naz, Transformer-based deep learning models for the sentiment analysis of social media data, *Array* **14** (2022) 100157, doi: 10.1016/j.array.2022.100157.
28. H. Lu, L. Ehwerhemuepha and C. Rakovski, A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance, *BMC Medical Research Methodology* **22** (2022) 181, doi: 10.1186/s12874-022-01665-y.
29. I. López Ramírez and J. Méndez Vargas, A sentiment analysis of the ukraine-russia conflict tweets using recurrent neural networks (2022). https://www.researchgate.net/publication/361275253_A_sentiment_analysis_of_the_Ukraine-Russia_conflict_tweets_using_Recurrent_Neural_Networks
30. A. Maas, R. Daly, P. Pham, D. Huang, A. Ng and C. Potts, Learning word vectors for sentiment analysis, in *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (ACL, 2011), pp. 142–150.
31. M. Masum, S. Ahmed, A. Tasnim and M. S. Islam, Ban-absa: An aspect-based sentiment analysis dataset for bengali and it's baseline evaluation, *Proceedings of International Joint Conference on Advances in Computational Intelligence* (2020), pp. 385–395.
32. S. Mohammad, S. Kiritchenko and X. Zhu, NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets, in *Proc. Seventh Int. Workshop on Semantic Evaluation (SemEval 2013)* (ACL, 2013), pp. 321–327.
33. C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, Activation functions: Comparison of trends in practice and research for deep learning, preprint (2018), arXiv:1811.03378.

34. R. Qasim, W. Bangyal, M. Alqarni and A. Almazroi, A fine-tuned bert-based transfer learning approach for text classification, *J. Healthcare Eng.* **2022** (2022) 1–17, doi: 10.1155/2022/3498123.
35. M. Rahman, M. Pramanik, R. Sadik, M. Roy and P. Chakraborty, Bangla documents classification using transformer based deep learning models, in *2020 2nd Int. Conf. Sustainable Technologies for Industry 4.0 (STI)* (IEEE, 2021), pp. 1–6, doi: 10.1109/STI50764.2020.9350394.
36. M. Rahman, S. Haque and Z. Saurav, Identifying and categorizing opinions expressed in bangla sentences using deep learning technique, *Int. J. Comput. Appl.* **176** (2020) 13–17, doi: 10.5120/ijca2020920119.
37. N. Romim, M. Ahmed, M. S. Islam, A. Sharma, H. Talukder and M. Amin, Bd-shs: A benchmark dataset for learning to detect online bangla hate speech in different social contexts, preprint (2022), arXiv:2206.00372.
38. V. Sanh, L. Debut, J. Chaumond and T. Wolf, Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter, *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS* (2019).
39. O. Sen, M. Fuad, M. N. Islam, J. Rabbi, M. Masud, M. Hasan, M. Awal, A. Fime, M. T. Fuad, D. Sikder and A. R. Iftee, Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning based methods, *IEEE Access* **10** (2022) 38999–39044, doi: 10.1109/ACCESS.2022.3165563.
40. R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng and C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in *Proc. 2013 Conf. Empirical Methods in Natural Language Processing* (ACL, 2013), pp. 1631–1642.
41. N. Sugiura, Further analysts of the data by akaike' s information criterion and the finite corrections: Semantic scholar (1978), https://www.semanticscholar.org/paper/Further-analysts-of-the-data-by-akaike'-s-criterion-Sugiura/0b3995e33daf8fea7a772585ff7dac925af7fae9.
42. N. Tripto and M. E. Ali, Detecting multilabel sentiment and emotions from bangla youtube comments, in *2018 Int. Conf. Bangla Speech and Language Processing (ICBSLP)* (IEEE, 2018), pp. 1–6, doi: 10.1109/ICBSLP.2018.8554875.
43. M. Wahid, M. J. Hasan and M. S. Alom, Cricket sentiment analysis from bangla text using recurrent neural network with long short term memory model, in *2019 Int. Conf. Bangla Speech and Language Processing (ICBSLP)* (IEEE, 2019), pp. 1–4, doi: 10.1109/ICBSLP47725.2019.201500.
44. T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Scao, S. Gugger and A. Rush, Transformers: State-of-the-art natural language processing, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing: System Demonstrations* (ACL, 2020), pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6.