BLP 2023

The First Workshop on Bangla Language Processing (BLP-2023)

Proceedings of the Workshop

December 7, 2023

The BLP organizers gratefully acknowledge the support and sponsorship provided by the following organizations.

Supported By

Gold Sponsor

Bronze Sponsors

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 979-8-89176-058-5

Preface

Welcome to the First Workshop on Bangla Language Processing (BLP 2023) collocated with the EMNLP 2023, hosted in Singapore.

In this first edition, the program is rich and varied, featuring a keynote talk, four paper presentation sessions, two poster sessions, a panel discussion, and an industry presentation.

At this first edition of the BLP workshop, we received 24 submissions comprising 16 long and 8 short papers. Each paper was rigorously peer-reviewed by at least three expert reviewers in the field. From these submissions, 16 papers were accepted, including 10 long and 6 short papers, all of which have been selected for oral presentation. Note that we made no distinction in the quality between long and short papers, or between oral and poster presentations.

The workshop featured two shared tasks: (1) Task 1: Violence Inciting Text Detection (VITD), and (2) Task 2: Sentiment Analysis of Bangla Social Media Posts. Both tasks were well-received, with robust participation. For Task 1, we had 27 team registrations, out of which 16 submitted system description papers. For task 2 we had registrations from 71 teams, with 29 and 30 teams participating in the development and evaluation phases, respectively, culminating in 15 system description papers.

Each system description paper for the shared tasks peer-reviewed by at least three expert reviewers and each system description papers were reviewed by two reviewers. The proceedings include these papers along with two comprehensive overview papers, which will be presented in an oral session at the workshop.

We were fortunate to secure sponsorship funding for the workshop, which has been instrumental in subsidizing registrations for students and aspiring young researchers.

Finally, we would like to thank all the contributors of papers and the 81 members of the Program Committee for their dedication to ensuring the delivery of high-quality reviews in a timely manner.

Firoj Alam, Sudipta Kar, and Shammur Absar Chowdhury
On Behalf of the BLP Workshop Organizing Committee
Workshop website at https://blp-workshop.github.io/

Organizing Committee

Organizers

Firoj Alam, Qatar Computing Research Institute, Qatar
Sudipta Kar, Amazon Alexa AI, USA
Shammur Absar Chowdhury, Qatar Computing Research Institute, Qatar
Farig Sadeque, BRAC University, Bangladesh
Ruhul Amin, Fordham University, USA

Program Committee

Abhik Bhattacharjee, BUET
Adnan Ahmad, Technische Universität Berlin
Avijit Mitra, University of Massachusetts
Avisha Das, University of Houston
Ayan Bhunia, iSize
Biddut Sarker Bijoy, Shahjalal University of Science and Technology
Biplob Biswas, Ohio State University
Debanjan Ghosh, Educational Testing Service
Farhana Diba, The George Washington University
Ercong Nie, Ludwig-Maximilians-Universität München
Fardin Ahsan Sakib, George Mason University
Farzana Islam Adiba, University of Florida
Guneet Singh Kohli, GreyOrange
Hasan Mesbaul Ali Taher, Chittagong University of Engineering and Technology
Jeniya Tabassum, Amazon
Kallol Naha, University of Idaho
Ketan Kumar Todi, Google
Khondoker Ittehadul Islam, Shahjalal University of Science and Technology
Krishno Dey, University of New Brunswick
Labiba Jahan, Southern Methodist University
M Saiful Bari, Nanyang Technological University
Madhusudan Basak, Dartmouth College
Matin Saad Abdullah, BRAC University
Md Mushfiqur Rahman, George Mason University
M. Moshiul Hoque, CUET
Md Nishat Raihan, George Mason University
Md Rashad Al Hasan Rony, BMW Group
Md Rizwan Parvez, Qatar Computing Research Institute
Md Saiful Islam, University of Alberta
Md Tahmid Rahman Laskar, Dialpad Inc.
Md Tanvirul Alam, Rochester Institute of Technology
Md Taufiq Nasseef, Prince Sattam Bin Abdulaziz University
Md Zobaer Hossain, Université de Lorraine
Md. Arid Hasan, University of New Brunswick
Md. Mahadi Hassan, Auburn University
Md. Rafiul Biswas, HBKU
Md. Sanzidul Islam, King Abdul Aziz University
Md. Towhidul Absar Chowdhury, Rochester Institute of Technology
Mehedi Hasan Bijoy, Bangladesh University of Business & Technology
Mohammod Akib Khan, BRAC University
Naira Khan, University of Dhaka
Nasheen Nur, Florida Institute of Technology
Navid Ayoobi, University of Houston
Niladri Sekhar Dash, ISI
Rabindra Nath Nandi, Hishab Singapure Pte. Ltd
Rifat Shahriyar, Bangladesh University of Engineering and Technology
Sabit Hassan, University of Pittsburgh
Sadat Shahriar, University of Houston
Saddam Hossain, United International University
Sadid A. Hasan, Microsoft

Sadiya Sayara Chowdhury Puspo, George Mason University
Sajjadur Rahman, Megagon Labs
Sanaul Haque, Jönköping University
Sarah Masud Preum, Dartmouth College
Shafiq Rayhan Joty, NTU and Salesforce
Shafiqul Islam, University of Essex
Sheak Rashed Haider Noori, Daffodil University
Shamik Roy, Amazon
Shamsuddeen Hassan Muhammad, Bayero University
Shervin Malmasi, Amazon
Shubhra (Santu) Karmaker, Auburn University
Sivaji Bandyopadhyay, Jadavpur University
Somnath Banarjee, University of Tartu
Soujanya Poria, Singapore University of Technology and Design
Soumya Sarkar, Microsoft
Souvika Sarkar, Auburn University
Subham De, Meta
Surendrabikram Thapa, Virginia Polytechnic Institute and State University
Syeda Jannatus Saba, State University of New York at Stony Brook
Sudip Kumar Naskar, Jadavpur University
Syed Mobassir Hossen, Apsis Solutions
Tahmid Hasan, Bangladesh University of Engineering and Technology
Tamanna Hossain-Kay, University of California
Tanmoy Chakraborty, IIT Delhi
Tanvirul Alam, RIT
Tashin Ahmed, Smart Studios
Tawhida Jahan, University of Dhaka
Wasi Uddin Ahmad, Amazon
Yassine El Kheir, KTH Royal Institute of Technology
Zeerak Talat, Mohamed bin Zayed University of Artificial Intelligence

Keynote Talk: NLP in Mexican Spanish: A path through shared tasks

Manuel Montes-y-Gómez
2023-12-07 09:20:00 – Room: **Pisces 2 & 3**

Abstract: Although Spanish is one of the most spoken languages in the world, it was only until very recently that the development of linguistic technologies for it had a strong boost. However, this is not entirely true for some of its Latin American variants, such as the Mexican Spanish, which show phonetic, and also some lexical and semantic differences with respect to peninsular Spanish.
This talk will focus on presenting the development of NLP for Mexican Spanish, emphasizing the path taken through the organization of different evaluation campaigns. It will present some data about Mexican Spanish as well as about the impact of the organization of shared tasks in the context of IberLEF for the development of the NLP area in our country, first as a mechanism to motivate more students to get involved, and then as a vehicle to build resources and design and implement specific methods. The talk will conclude by exposing some of the obstacles faced, our main achievements, and some plans for the coming years.

Bio: Manuel Montes is Full Professor at the National Institute of Astrophysics, Optics and Electronics (INAOE) of Mexico. His research is on automatic text processing. He is author of more than 250 journal and conference papers in the fields of information retrieval, text mining and authorship analysis.
He has been visiting professor at the Polytechnic University of Valencia (Spain), and the University of Alabama (USA). He is also member of the Mexican Academy of Sciences (AMC), and founding member of the Mexican Academy of Computer Science (AMEXCOMP), and the Mexican Association of Natural Language Processing (AMNLP). In the context of the latter, he has been the organizer of the National Workshop on Language Technologies (from 2004 to 2016), the Mexican Workshop on Plagiarism Detection and Authorship Analysis (2016-2020), the Mexican Autumn School on Language Technologies (2015 and 2016), and shared tasks on author profiling, aggressiveness analysis and fake news detection in Mexican Spanish at IberLEF (2018-2021).

Keynote Talk: Towards Transforming the Landscape of Indian Language Technology

Mitesh Khapra
2023-12-07 15:05:00 – Room: **Pisces 2 & 3**

Abstract: In this talk, I will reflect on our journey towards transforming the landscape of Indian language technology. I will delve on our engineering-heavy approach in addressing the initial scarcity of data for Indian languages, while gradually establishing the necessary human resources to gather high-quality data on a larger scale through Bhashini. The objective is to share our insights into developing high quality open-source technology for Indian languages. This involves curating extensive data from the internet, constructing multilingual models for transfer learning, and crafting high-quality datasets for fine-tuning and evaluation. I will then transition into how our experiences can benefit the broader AI community, particularly as India aspires to create Language Model Models (LLMs) for Indic languages.

Bio: Mitesh M. Khapra is an Associate Professor in the Department of Computer Science and Engineering at IIT Madras. He heads the AI4Bharat Research Lab at IIT Madras which focuses on building datasets, tools, models and applications for Indian languages. His research work has been published in several top conferences and journals including TACL, ACL, NeurIPS, TALLIP, EMNLP, EACL, AAAI, etc. He has also served as Area Chair or Senior PC member in top conferences such as ICLR and AAAI. Prior to IIT Madras, he was a Researcher at IBM Research India for four and a half years, where he worked on several interesting problems in the areas of Statistical Machine Translation, Cross Language Learning, Multimodal Learning, Argument Mining and Deep Learning. Prior to IBM, he completed his PhD and M.Tech from IIT Bombay in Jan 2012 and July 2008 respectively. His PhD thesis dealt with the important problem of reusing resources for multilingual computation. During his PhD he was a recipient of the IBM PhD Fellowship (2011) and the Microsoft Rising Star Award (2011). He is also a recipient of the Google Faculty Research Award (2018), the IITM Young Faculty Recognition Award (2019), the Prof. B. Yegnanarayana Award for Excellence in Research and Teaching (2020) and the Srimathi Marti Annapurna Gurunath Award for Excellence in Teaching (2022).

Table of Contents

Program

Thursday, December 7, 2023

09:00 - 09:20 *Opening Remarks*

09:20 - 09:50 *Invited Talk 1: NLP in Mexican Spanish: A Path Through Shared Tasks*

09:50 - 10:26 *Oral Presentation I (long papers)*

BSpell: A CNN-Blended BERT Based Bangla Spell Checker
Chowdhury Rafeed Rahman, MD.Hasibur Rahman, Samiha Zakir, Mohammad Rafsan and Mohammed Eunus Ali

BLP-2023 Task 1: Violence Inciting Text Detection (VITD)
Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed and Mohammad Ruhul Amin

BLP-2023 Task 2: Sentiment Analysis
Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das and Afiyat Anjum

10:30 - 11:00 *Coffee Break*

11:00 - 12:00 *Poster Session*

12:00 - 13:00 *Lunch Break*

13:00 - 14:00 *Oral Presentation II (long papers)*

Low-Resource Text Style Transfer for Bangla: Data & Models
Sourabrata Mukherjee, Akanksha Bansal, Pritha Majumdar, Atul Kr Ojha and Ondrej Dusek

Vio-Lens: A Novel Dataset of Annotated Social Network Posts Leading to Different Forms of Communal Violence and its Evaluation
Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahouti, Syed Ishtiaque Ahmed, Nabeel Mohammed and Mohammad Ruhul Amin

Pseudo-Labeling for Domain-Agnostic Bangla Automatic Speech Recognition
Rabindra Nath Nandi, Mehadi Hasan Menon, Tareq Al Muntasir, Sagor Sarker, Quazi Sarwar Muhtaseem, Md. Tariqul Islam, Shammur Absar Chowdhury and Firoj Alam

Thursday, December 7, 2023 (continued)

Contextual Bangla Neural Stemmer: Finding Contextualized Root Word Representations for Bangla Words
Md Fahim, Amin Ahsan Ali, M Ashraful Amin and Akmmahbubur Rahman

Crosslingual Retrieval Augmented In-context Learning for Bangla
Xiaoqian Li, Ercong Nie and Sheng Liang

14:00 - 14:10 *Break*

14:10 - 15:05 *Oral Presentation III (long + short papers)*

Advancing Bangla Punctuation Restoration by a Monolingual Transformer-Based Method and a Large-Scale Corpus
Mehedi Hasan Bijoy, Mir Fatema Afroz Faria, Mahbub E Sobhani, Tanzid Ferdoush and Swakkhar Shatabda

BanglaCHQ-Summ: An Abstractive Summarization Dataset for Medical Queries in Bangla Conversational Speech
Alvi Aveen Khan, Fida Kamal, Mohammad Abrar Chowdhury, Tasnim Ahmed, Md Tahmid Rahman Laskar and Sabbir Ahmed

Offensive Language Identification in Transliterated and Code-Mixed Bangla
Md Nishat Raihan, Umma Hani Tanmoy, Anika Binte Islam, Kai North, Tharindu Ranasinghe, Antonios Anastasopoulos and Marcos Zampieri

Intent Detection and Slot Filling for Home Assistants: Dataset and Analysis for Bangla and Sylheti
Fardin Ahsan Sakib, A H M Rezaul Karim, Saadat Hasan Khan and Md Mushfiqur Rahman

Assessing Political Inclination of Bangla Language Models
Surendrabikram Thapa, Ashwarya Maratha, Khan Md Hasib, Mehwish Nasim and Usman Naseem

SynthNID: Synthetic Data to Improve End-to-end Bangla Document Key Information Extraction
Syed Mostofa Monsur, Shariar Kabir and Sakib Chowdhury

BEmoLexBERT: A Hybrid Model for Multilabel Textual Emotion Classification in Bangla by Combining Transformers with Lexicon Features
Ahasan Kabir, Animesh Chandra Roy and Zaima Sartaj Taheri

15:05 - 15:35 *Invited Talk 2: Towards Transforming the Landscape of Indian language Technology*

Thursday, December 7, 2023 (continued)

15:35 - 16:00 *Coffee Break*

16:00 - 16:36 *Oral Presentation IV (long papers)*

BaTEClaCor: A Novel Dataset for Bangla Text Error Classification and Correction
Nabilah Tabassum Oshin, Syed Mohaiminul Hoque, Md Fahim, Amin Ahsan Ali, M Ashraful Amin and Akmmahbubur Rahman

Investigating the Effectiveness of Graph-based Algorithm for Bangla Text Classification
Farhan Noor Dehan, Md Fahim, Amin Ahsan Ali, M Ashraful Amin and Akmmahbubur Rahman

Pipeline Enabling Zero-shot Classification for Bangla Handwritten Grapheme
Linsheng Guo, Md Habibur Rahman Sifat and Tashin Ahmed

16:36 - 17:15 *Panel Discussion*

17:15 - 17:30 *Industry Talk*

17:30 - 17:45 *Awards and Ending Remarks*

Offensive Language Identification in Transliterated and Code-Mixed Bangla

Md Nishat Raihan[1], Umma Hani Tanmoy[1], Anika Binte Islam[1], Kai North[1]
Tharindu Ranasinghe[2], Antonios Anastasopoulos[1], Marcos Zampieri[1]
[1]George Mason University, USA
[2]Aston University, UK
mraihan2@gmu.edu

Abstract

Identifying offensive content in social media is vital for creating safe online communities. Several recent studies have addressed this problem by creating datasets for various languages. In this paper, we explore offensive language identification in texts with transliterations and code-mixing, linguistic phenomena common in multilingual societies, and a known challenge for NLP systems. We introduce TB-OLID, a transliterated Bangla offensive language dataset containing 5,000 manually annotated comments. We train and fine-tune machine learning models on TB-OLID, and we evaluate their results on this dataset. Our results show that English pre-trained transformer-based models, such as fBERT and HateBERT achieve the best performance on this dataset.

1 Introduction

As the popularity of social media continues to grow, the spread of offensive content in these platforms has increased substantially, motivating companies to invest heavily in content moderation strategies and robust models to detect offensive content. We have observed a growing interest in this topic, evidenced by popular shared tasks at SemEval (Basile et al., 2019) and other venues. Apart from a few notable exceptions (Mandl et al., 2020), most of the work on this topic has not addressed the question of transliteration and code-mixing, two common phenomena in social media.

Code-mixing is the phenomenon of embedding linguistic units such as phrases, words, or morphemes of one language into another language (Myers-Scotton, 1997; Muysken et al., 2000). Code-mixed texts often feature transliterations where speakers use an alternative script to the language's official or standard script by mapping from one writing system (e.g., Hindi and its original Devanagari script) to another one (e.g., Latin transliteration of Hindi) based on phonetic similarity. Transliterated texts are widely used in social media platforms as transliteration allows users to write in their native language using a script that may not be supported by the platform and/or using Latin-based default keyboards. Furthermore, the use of transliteration also allows users to easily switch between languages with otherwise different scripts (e.g., English and Hindi). As discussed in a recent survey (Winata et al., 2022), however, processing code-mixing datasets is a challenge that hinders performance in a variety of NLP tasks, thus deserving special attention.

Code-mixing and transliteration are common in various languages, including Bangla (Das and Gambäck, 2015; Jamatia et al., 2015). Related work on Bangla offensive language identification (Wadud et al., 2021), however, has mostly focused on standard Bangla script. As such, the performance of offensive language identification models on code-mixing and transliterated Bangla remains largely unexplored. To address this shortcoming, we create TB-OLID, a manually annotated transliterated Bangla offensive language dataset. TB-OLID was annotated following the popular OLID hierarchical taxonomy (Zampieri et al., 2019a), allowing cross-lingual experiments. To the best of our knowledge, the dataset is the first of its kind for Bangla, opening exciting new avenues for future research.

The main contributions of this paper are as follows:

1. We introduce TB-OLID, an offensive language identification corpus containing 5,000 Facebook comments.[1]
2. We provide a comparative analysis of various machine learning models trained or fine-tuned on TB-OLID.

WARNING: This paper contains examples that are offensive in nature.

[1]https://github.com/LanguageTechnologyLab/TB-OLID

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 1–6

Comment	C|T	O|N	I|G|U
BN: *Tui to 1 ta rastar chele mother chod.* EN: You are a motherfucking street vagabond	**C**	**O**	**I**
BN: *O to Manush na sokun* EN: He/She is not a person but a vulture	**T**	**O**	**I**
BN: *R kichudin por kanglu ra Uth ar mut diye cha banabe* EN: After some days, the barbarians will make tea with camel piss	**T**	**O**	**G**
BN: *Dhoren r dog ar baccha gulo ke gono dholai diye pongu kore den* EN: Capture these son of bitches and beat them to their death	**C**	**O**	**G**
BN: *Pagole kina bole chagole kina khai* EN: A mad man and an animal have no difference	**T**	**O**	**U**
BN: *Ami kintu parlam na hojom korte* EN: I cannot fathom it anymore	**T**	**N**	

Table 1: Examples from TB-OLID in Bangla along with an English translation. The labels included are C (transliterated code-mixed), T (transliterated Bangla), O (offensive), N (not-offensive), I (offensive posts targeted at an individual), G (offensive posts targeted at a group), and U (untargeted offensive posts).

2 Data

Data Collection We collect data from Facebook, the most popular social media platform in Bangladesh. We compile a list of the most popular Facebook pages in Bangladesh using Fanpage Karma[2] and scraped comments from each of the top 100 most followed Facebook pages using the publicly available Facebook scraper tool.[3] This results in an initial corpus of over 100,000 comments. We exclude all comments not written with non-Latin script. We search the corpus using keywords for transliterated hate speech and offensive language. We select keywords from the list of 175 offensive Bangla terms by Karim et al. (2021). As the dataset by Karim et al. (2020) contains standard Bangla, we convert keywords into transliterated Bangla using the Indic-transliteration tool.[4] Using these keywords we randomly select a set of 5,000 comments for annotation.

Annotation Guidelines We prepare the TB-OLID annotation guidelines containing labels and examples. The first step is to label whether a comment is transliterated Bangla or transliterated code-mixed Bangla. If the comment contains at least one English word along with other Bangla transliterated words, we consider it as transliterated code-mixed. Next, we consider the offensive vs. non-offensive distinction and, in the case of offensive posts, its target or lack thereof. Table 1 presents six annotated instances included in TB-OLID.

We adopt the guidelines introduced by the popular OLID annotation taxonomy (Zampieri et al., 2019a) used in the OffensEval shared task (Zampieri et al., 2019b) and replicated in multiple other datasets in languages such as Danish (Sigurbergsson and Derczynski, 2020), Greek (Pitenis et al., 2020), Marathi (Gaikwad et al., 2021; Zampieri et al., 2022), Portuguese (Sigurbergsson and Derczynski, 2020), Sinhala (Ranasinghe et al., 2022) and Turkish (Çöltekin, 2020). We choose OLID due to the flexibility provided by its three-level hierarchical taxonomy that allows us to model different types of offensive and abusive content (e.g., hate speech, cyberbulling, etc.) using a single taxonomy. OLID's taxonomy considers whether an instance is offensive (level A), whether an offensive post is targeted or untargeted (level B), and what is the target of an offensive post (level C). As the second level of the TB-OLID annotation we consider OLID level A as follows.

- **Offensive:** Comments that contain any form of non-acceptable language or a targeted offense, including insults, threats, and posts containing profane language
- **Non-offensive:** Comments that do not contain any offensive language

Finally, the third level of the TB-OLID annotation merges OLIDs level B and C. We label whether a post is untargeted or, when targeted, whether it is labeled at an individual or a group as follows:

- **Individual:** Comments targeting any individ-

[2] https://www.fanpagekarma.com/
[3] https://github.com/kevinzg/facebook-scraper
[4] https://github.com/sanskrit-coders/indic_transliteration_py

ual, such as mentioning a person with his/her name, unnamed participants, or famous personality.

- **Group:** Comments targeting any group of people of common characteristics, religion, gender, etc.
- **Untargeted:** Comments containing unacceptably strong language or profanities that are not targeted.

Ensuring Annotation Quality Three annotators working on this project are tasked to annotate TB-OLID. They are PhD students in Computing aged 22-28 years old, 1 male and 2 female, all native speakers of Bangla and fluent speakers of English. The first step of the annotation process involves a pilot annotation study, where 300 comments are assigned to all three annotators to calculate initial agreement and refine the annotation guidelines according to their feedback. After this pilot experiment, we annotate an additional 4,700 Facebook comments totaling 5,000 instances which are subsequently split into 4,000 and 1,000 instances for training and testing, respectively. The instances in TB-OLID are annotated by at least two annotators, with the third one serving as adjudicator. We calculate pairwise inter-annotator agreement on 1,000 instances using Cohen's Kappa, and we report Cohen's Kappa score of 0.77 and 0.72 for levels 1 (code-mixed vs. transliterated) and 2 (offensive vs. non-offensive), which is generally considered substantial agreement. We report Cohen's Kappa score of 0.66 on level 3, considered moderate agreement.

Dataset Statistics We calculated the frequency of each label in the dataset namely code-mixed and transliterated, offensive and non-offensive, targeted and untargeted, and target types in the dataset. The dataset statistics are presented in Table 2.

Level	Label	Instances	Percentage
1	T	2,959	59.18%
	C	2,041	41.82%
2	O	2,381	47.62%
	N	2,619	52.38%
3	I	1,192	23.84%
	G	954	19.08%
	U	235	4.70%

Table 2: TB-OLID per level and per class statistics. Percentage calculated considering the total number of instances in the dataset (5,000).

Finally, we run an analysis of the code-mixed data using ad-hoc Python scripts. We observe that English is by far the most common language included in the code-mixed instances mixed with Bangla followed by Hindi. We report that 38.42% of all tokens in the code-mixed (C) class are English.

3 Baselines and Models

Baselines We report the results of three baselines: (1) Google's Perspective API[5], a free API developed to detect offensive comments widely used as a baseline in this task (Kaati et al., 2022; Fortuna et al., 2020); (2) prompting GPT 3.5 turbo providing the model with TB-OLID's annotation guidelines; and (3) a majority class baseline. Due to the API's limitations, Perspective API was used only for offensive language identification and not for for target classification.

General Models We experiment with pre-trained language models fine-tuned on TB-OLID. As our dataset is transliterated Bangla and contains English code-mixed, we experiment with BERT (Devlin et al., 2019), roBERTa (Liu et al., 2020) which are trained on English, and Bangla-BERT (Kowsher et al., 2022), which is trained on Bangla. We also use cross-lingual models such as mBERT (Devlin et al., 2019) and xlm-roBERTa (Conneau et al., 2020) which are trained in multiple languages.

Task-specific Models We also experiment with task-specific fined-tuned models like HateBERT (Caselli et al., 2021), and fBERT (Sarkar et al., 2021). These models were also further fined-tuned on TB-OLID.

4 Results and Discussion

We use F1-score to evaluate the performance of all models. The training and test sets are obtained by the aforementioned 80-20 random split on the entire TB-OLID dataset. We further subdivide the test set into transliterated code-mixed (C), transliterated (T), and all instances. We present results for offensive text classification (offensive vs. non-offensive) in Table 3.

We observe that the standard BERT model performs well over the baselines, whereas the Bangla-BERT model performs less well. BERT achieves F1-score of 0.71, whereas Bangla-BERT obtains F1-score of 0.42. We believe this is due to the

[5] https://perspectiveapi.com/

Model	C	T	All
fBERT	0.73	0.70	0.72
HateBERT	0.74	0.69	0.72
BERT	0.73	0.68	0.71
m-BERT	0.70	0.68	0.69
GPT 3.5	0.65	0.64	0.64
Majority Class Baseline	0.57	0.57	0.57
Perspective API	0.53	0.50	0.51
Bangla-BERT	0.42	0.42	0.42
xlm-roBERTa	0.40	0.41	0.41
roBERTa	0.41	0.41	0.41

Table 3: Offensive Language Identification - F1-score of all models trained and/or fine-tuned on TB-OLID. We report results on the transliterated code-mixed (C), transliterated (T), and All test set. Baselines in italics.

fact that many instances in the dataset are in Latin script, which means that BanglaBERT frequently struggles with out-of-vocabulary tokens. The low performance of Bangla-BERT in this task requires further examination. Models pre-trained specifically on offensive language identification perform very well with fBERT and Hate-BERT coming out on top, both with an F1 score of 0.72. Finally, we observe that the top-5 performing models perform better on the code-mixing data compared to transliterated data. This is likely due to the heavy presence of English words in the code-mixing data where we observe the presence of 38% of English words.

Finally, Table 4 presents the results of target type classification (individual, group, or untargeted).

Model	C	T	All
HateBERT	0.69	0.66	0.68
m-BERT	0.72	0.64	0.67
BERT	0.72	0.64	0.67
fBERT	0.66	0.64	0.65
roBERTa	0.73	0.60	0.65
GPT 3.5	0.39	0.46	0.43
Majority Class Baseline	0.48	0.63	0.55
xlm-roBERTa	0.61	0.51	0.55
Bangla-BERT	0.59	0.47	0.51

Table 4: Target Classification - F1-score of all models trained and/or fine-tuned on TB-OLID. We report results on the transliterated code-mixed (C), transliterated (T), and All test sets. Baselines in italics.

Overall, target classification is a more challenging task than offensive language identification due to the presence of three classes instead of two. Therefore, all results are substantially lower for this task. HateBERT performs better than all other models with an F1 score of 0.68. roBERTa achieved more competitive performance for target classification than for offensive language identification whereas Bangla-BERT did not perform well in both tasks. Finally, similar to the previous task, the best-performing models achieved higher F1 scores on the code-mixed data than on the transliterated data.

One key observation is that the transformer-based models do not perform very well, since most of them are not pre-trained on transliterated Bangla. Among the models that we experiment with, only xlm-roBERTa is pre-trained with a comparatively small set of Romanized Bangla. However, the lack of any standard rules for spelling in transliterated Bangla makes TB-OLID very challenging.

5 Conclusion and Future Work

In this work, we introduced TB-OLID, a transliterated Bangla offensive language dataset containing 5,000 instances retrieved from Facebook. Three native speakers of Bangla have annotated the dataset with respect to the presence of code-mixing, the presence of offensive language, and its target according to the OLID taxonomy. TB-OLID opens exciting new avenues for research on offensive language identification in Bangla.

We performed experiments with multiple models such as general monolingual models like BERT (Devlin et al., 2019), roBERTa (Liu et al., 2020) and Bangla-BERT (Kowsher et al., 2022); cross-lingual models like mBERT (Devlin et al., 2019) and xlm-roBERTa (Conneau et al., 2020); and models fine-tuned for offensive language identification like HateBERT (Caselli et al., 2021), and fBERT (Sarkar et al., 2021)). The best results were obtained by the task-specific models.

In future work, we would like to extend the TB-OLID dataset and annotate the offense type (e.g., religious offense, political offense, etc.). This would help us identify the common targets in various platforms. Furthermore, we would like to pre-train and fine-tune a Bangla transliterated BERT model to see how it performs on TB-OLID. Finally, in future work, we would like to evaluate the performance of other recently released large language models (LLMs) (e.g., GPT 4.0, Llama 2) on TB-OLID. The first baseline results using GPT 3.5 indicate that general-purpose LLMs still struggle with the transliterated and code-mixed content presented in TB-OLID.

Acknowledgments

We thank the anonymous workshop reviewers for their insightful feedback. Antonios Anastasopoulos is generously supported by NSF award IIS-2125466.

Ethics Statement

The generation and annotation procedure of TB-OLID adheres to the ACL Ethics Policy and seeks to make a valuable contribution to the realm of online safety. The technology in question possesses the potential to serve as a beneficial instrument for the moderation of online content, thereby facilitating the creation of safer digital environments. However, it is imperative to exercise caution and implement stringent regulations to prevent its potential misuse for purposes such as monitoring or censorship.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of SemEval*.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. Hatebert: Retraining bert for abusive language detection in english. In *Proceedings of WOAH*.

Çağrı Çöltekin. 2020. A Corpus of Turkish Offensive Language on Social Media. In *Proceedings of LREC*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Amitava Das and Björn Gambäck. 2015. Code-mixing in social media text: The last language identification frontier? *Revue TAL - Association pour le Traitement Automatique des Langues (ATALA)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Paula Fortuna, Juan Soler, and Leo Wanner. 2020. Toxic, hateful, offensive, or abusive? what are we really classifying? an empirical analysis of hate speech datasets. In *Proceedings of LREC*.

Saurabh Sampatrao Gaikwad, Tharindu Ranasinghe, Marcos Zampieri, and Christopher Homan. 2021. Cross-lingual offensive language identification for low resource languages: The case of marathi. In *Proceedings of RANLP*.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. In *Proceedings of RANLP*.

Lisa Kaati, Amendra Shrestha, and Nazar Akrami. 2022. A machine learning approach to identify toxic language in the online space. In *Proceedings of ASONAM*.

Md Rezaul Karim, Bharathi Raja Chakravarthi, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In *Proceedings of DSAA*.

Md Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Md Azam Hossain, and Stefan Decker. 2021. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *Proceedings of DSAA*.

M Kowsher, Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, 10:91855–91870.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Roberta: A robustly optimized bert pretraining approach. In *Proceedings of ACL*.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german. In *Proceedings of FIRE*.

Pieter Muysken et al. 2000. *Bilingual speech: A typology of code-mixing*. Cambridge University Press.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Zesis Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive language identification in greek. In *Proceedings of LREC*.

Tharindu Ranasinghe, Isuri Anuradha, Damith Premasiri, Kanishka Silva, Hansi Hettiarachchi, Lasitha Uyangodage, and Marcos Zampieri. 2022. SOLD: Sinhala offensive language dataset. *arXiv preprint arXiv:2212.00851*.

Diptanu Sarkar, Marcos Zampieri, Tharindu Ranasinghe, and Alexander Ororbia. 2021. fbert: A neural transformer for identifying offensive content. In *Findings of the ACL*.

Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2020. Offensive Language and Hate Speech Detection for Danish. In *Proceedings of LREC*.

Md Anwar Hussen Wadud, Md Abdul Hamid, Muhammad Mostafa Monowar, and Atif Alamri. 2021. L-boost: Identifying offensive texts from social media post in bengali. *Ieee Access*, 9:164681–164699.

Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2022. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In *Findings of the ACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of SemEval*.

Marcos Zampieri, Tharindu Ranasinghe, Mrinal Chaudhari, Saurabh Gaikwad, Prajwal Krishna, Mayuresh Nene, and Shrunali Paygude. 2022. Predicting the type and target of offensive social media posts in Marathi. *Social Network Analysis and Mining*, 12(1).

BSpell: A CNN-Blended BERT Based Bangla Spell Checker

Chowdhury Rafeed Rahman
National University of Singapore
e0823054@u.nus.edu

MD. Hasibur Rahman
United International University

Samiha Zakir and **Mohammed Rafsan**
University of Texas Rio Grande Valley

Mohammed Eunus Ali
Bangladesh University of Engineering and Technology

Abstract

Bangla typing is mostly performed using English keyboard and can be highly erroneous due to the presence of compound and similarly pronounced letters. Spelling correction of a misspelled word requires understanding of word typing pattern as well as the context of the word usage. A specialized BERT model named *BSpell* has been proposed in this paper targeted towards word for word correction in sentence level. *BSpell* contains an end-to-end trainable CNN sub-model named *SemanticNet* along with specialized auxiliary loss. This allows *BSpell* to specialize in highly inflected Bangla vocabulary in the presence of spelling errors. Furthermore, a hybrid pretraining scheme has been proposed for *BSpell* that combines word level and character level masking. Comparison on two Bangla and one Hindi spelling correction dataset shows the superiority of our proposed approach. *BSpell* is available as a Bangla spell checking tool via GitHub: *https://github.com/Hasiburshanto/Bangla-Spell-Checker*.

1 Introduction

Bangla is the native language of 228 million people which makes it the sixth most spoken language in the world [1]. This Sanskrit originated language has 11 vowels, 39 consonants, 11 modified vowels and 170 compound characters (Sifat et al., 2020). There is vast difference between Bangla grapheme representation and phonetic utterance for many commonly used words. As a result, fast typing of Bangla yields frequent spelling mistakes. Almost all Bangla native speakers type using English QWERTY layout keyboard (Noyes, 1983) which makes it difficult to type Bangla compound characters, phonetically similar single characters and similar pronounced modified vowels correctly. Thus Bangla typing speed, if error-free typing is desired, is slow. An accurate spell checker (SC) can be a solution to this problem.

Existing Bangla SCs include phonetic rule (Uz-Zaman and Khan, 2004, 2005) and clustering based methods (Mandal and Hossain, 2017). These methods do not take misspelled word context into consideration. Another N-gram based Bangla SC (Khan et al., 2014) takes only short range previous context into consideration. Recent state-of-the-art (SOTA) spell checkers have been developed for Chinese language, where a character level confusion set (similar characters) guided sequence to sequence (seq2seq) model has been proposed by Wang et al. (2019). Another research used similarity mapping graph convolutional network in order to guide BERT based character by character parallel correction (Cheng et al., 2020). Both these methods require external knowledge and assumption about confusing character pairs existing in the language. The most recent Chinese SC offers an assumption free BERT architecture where error detection network based soft-masking is included (Zhang et al., 2020). This model takes all N characters of a sentence as input and produces the correct version of these N characters as output in a parallel manner.

Incorrect	Correct
পরিকা (প+র+ ি+ক+া)	পরীক্ষা (প+র+ী+ক+ ্+ষ+া): Exam
বিশশ (ব+ি+শ+শ)	বিশ্ব (ব+ি+শ+ ্+ব): World
ভাদর (ভ+া+দ+র)	ভাদ্র (ভ+া+দ+ ্+র): month name

Figure 1: Heterogeneous character number between error word and corresponding correctly spelled word

One of the limitations in developing Bangla SC using SOTA BERT based implementation (Zhang et al., 2020) is that number of input and output characters in BERT has to be exactly the same. Such scheme is only capable of correcting substitution type errors. As compound characters are common in Bangla words, an error made due to the

[1] https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 7–17

substitution of such characters also changes word length (see the table in Figure 1). So, we introduce word level prediction in our proposed BERT based model.

Correct	Incorrect
সৈনিক ঘোড়া চড়ে যুদ্ধে গেল। (Soldier went to war riding a horse)	সৈনিক ঘোরা চড়ে যুদ্ধে গেল। (Soldier went to war riding a visit)
আসামী দোষ স্বীকার করল। (The criminal confessed crime)	আসামী দোষ শিকার করল। (The criminal hunted crime)
কাল আমাদের বার্ষিক পরীক্ষা। (Tomorrow is our final exam)	কাল আমাদের বার্ষিক পরিখা। (Tomorrow is our final trench)

Figure 2: ample words that are correctly spelled accidentally, but are context-wise incorrect.

The table shown in Figure 2 illustrates the importance of context in Bangla SC. Although the red marked words of this figure are the misspelled versions of the corresponding green marked correct words, these red words are valid Bangla words. But if we check these red words based on sentence semantic context, we can realize that these words have been produced accidentally because of spelling error. An effective SC has to consider word pattern, its prior context and its post context.

Misspelled: গরাম ক্রিশির অরর নিরভরশিল
Correct: গ্রাম কৃষির ওপর নির্ভরশীল
Meaning: Villages are dependent on agriculture

Misspelled	Correct	Context
গরাম	গ্রাম (village)	কৃষির
ক্রিশির	কৃষির (Agriculture)	গ্রাম, নির্ভরশীল
অরর	ওপর (on)	নির্ভরশীল
নিরভরশিল	নির্ভরশীল (dependent)	ওপর

Figure 3: Necessity of understanding existing erroneous words for spelling correction of misspelled words

Spelling errors often span up to multiple words in a sentence. Figure 3 provides an example where all four words have been misspelled. The correction of each word has context dependency on a few other words of the same sentence. The problem is that these words that form the correction context are also misspelled. The table in the figure shows the words to look at in order to correct each misspelled word. In the original sentence (colored in red), all these words that need to be looked at for context are misspelled. If a SC cannot understand the approximate underlying meaning of these misspelled words, then we lose all context for correcting each misspelled word which is undesirable.

We propose a word level BERT (Devlin et al., 2018) based model *BSpell*. This model is capable of learning prior and post context dependency through the use of multi-head attention mechanism of stacked Transformer encoders (Vaswani et al., 2017). The model uses CNN based learnable *SemanticNet* sub-model to capture semantic meaning of both correct and misspelled words. *BSpell* also uses specialized auxiliary loss to facilitate word level pattern learning and vanishing gradient problem removal. We introduce *hybrid pretraining*for *BSpell* to capture both context and word error pattern. We perform detailed evaluation on three error datasets that include a real life Bangla error dataset. Our evaluation includes detailed analysis on possible LSTM based SCs, SC variants of BERT and existing classic Bangla SCs.

2 Related Works

Several studies on Bangla SC development have been conducted in spite of Bangla being a low resource language. A phonetic encoding oriented Bangla word level SC based on Soundex algorithm was proposed by UzZaman and Khan (2004). This encoding scheme was later modified to develop a Double Metaphone encoding based Bangla SC (UzZaman and Khan, 2005). They took into account major context-sensitive rules and consonant clusters while performing their encoding scheme. Another word level Bangla SC able to handle both typographical and phonetic errors was proposed by Mandal and Hossain (2017). An N gram model was proposed by Khan et al. (2014) for checking sentence level Bangla word correctness. An encoder-decoder based seq2seq model was proposed by Islam et al. (2018) for Bangla sentence correction task which involved bad arrangement of words and missing words, though this work did not include incorrect spelling. A recent study has included Hindi and Telugu SC development, where mistakes are assumed to be made at character level (Etoori et al., 2018). They have used attention based encoder-decoder modeling as their approach.

SOTA research in this domain involves Chinese SCs as it is an error prone language due to its confusing word segmentation, phonetically and visually similar but semantically different characters. A seq2seq model assisted by a pointer network was employed for character level spell checking where the network is guided by externally generated character confusion set (Wang et al., 2019). Another research incorporated phonological and visual similarity knowledge of Chinese characters into BERT based SC model by utilizing graph

convolutional network (Cheng et al., 2020). A recent BERT based SC has taken advantage of GRU (Gated Recurrent Unit) based soft masking mechanism and has achieved SOTA performance in Chinese character level SC in spite of not providing any external knowledge to the network (Zhang et al., 2020). Another external knowledge free approach namely FASPell used BERT based seq2seq model (Hong et al., 2019). HanSpeller++ is notable among initially implemented Chinese SCs (Xiong et al., 2015). It was an unified framework utilizing a hidden Markov model.

3 Our Approach

3.1 Problem Statement

Suppose, an input sentence consists of n words – $Word_1$, $Word_2$, ..., $Word_n$. For each $Word_i$, we have to predict the right spelling, if $Word_i$ exists in the top-word list of our corpus. If $Word_i$ is a rare word (Proper Noun in most cases), we predict UNK token denoting that we do not make any correction to such words. For correcting a particular $Word_i$ in a paragraph, we only consider other words of the same sentence for context information.

3.2 BSpell Architecture

Figure 4 shows the details of *BSpell* architecture. Each input word of the sentence is passed through the ***SemanticNet* sub-model**. This sub-model returns us with a *SemanticVec* vector representation for each input word. These vectors are then passed onto two separate branches (**main branch** and **secondary branch**) simultaneously. The main branch is similar to BERT_Base architecture (Gong et al., 2019). This branch provides us with the n correct words corresponding to the n input sentence words at its output side. The secondary branch consists of an output dense layer. This branch is used for the sole purpose of imposing **auxiliary loss** to facilitate *SemanticNet* sub-model learning of misspelled word patterns.

3.2.1 SemanticNet Sub-Model

Correcting a particular word requires the understanding of other relevant words in the same sentence. Unfortunately, those relevant words may also be misspelled. As humans, we can understand the meaning of a word even if it is misspelled because of our deep understanding at word syllable level and our knowledge of usual spelling error pattern. We want our model to have similar semantic level understanding of the words. We propose *SemanticNet*, a sequential 1D CNN sub-model that is employed at each individual word level with a view to learning intra word syllable pattern. Details of individual word representation has been shown in the bottom right corner of Figure 4. We represent each input word by a matrix (each character represented as a one hot vector). We apply global max pooling on the final convolution layer output feature matrix of *SemanticNet* which gives us the *SemanticVec* vector representation of the input word. We get a similar *SemanticVec* representation from each of our input words by independently applying the same *SemanticNet* sub-model on each of their matrix representations.

3.2.2 BERT_Base as Main Branch

Each of the *SemanticVec* vector representations obtained from the input words are passed parallelly on to our first Transformer encoder. 12 such Transformer encoders are stacked on top of each other. Each Transformer employs multi head attention mechanism, layer normalization and dense layer specific modification on each input vector. The attention mechanism applied on the word feature vectors in each transformer layer helps the words of the input sentence interact with one another extracting sentence context. We pass the final Transformer layer output vectors to a dense layer with Softmax activation function applied on each vector in an independent manner. So, now we have n probability vectors from n words of the input sentence. Each probability vector contains len_P values, where len_P is one more than the total number of top words considered (the additional word represents rare words). The top word corresponding to the index of the maximum probability value of i^{th} probability vector represents the correct word for $Word_i$ of the input sentence.

3.2.3 Auxiliary Loss in Secondary Branch

Gradient vanishing problem is a common phenomena in deep neural networks, where weights of the shallow layers are not updated sufficiently during backpropagation. With the presence of 12 Transformer encoders on top of the *SemanticNet* sub-model, the layers of this sub-model certainly lie in a shallow position. Although *SemanticNet* constitutes a small initial portion of *BSpell*, this portion is responsible for word pattern learning, an important task of SC. In order to eliminate gradient vanishing problem of *SemanticNet* and to turn it into an ef-

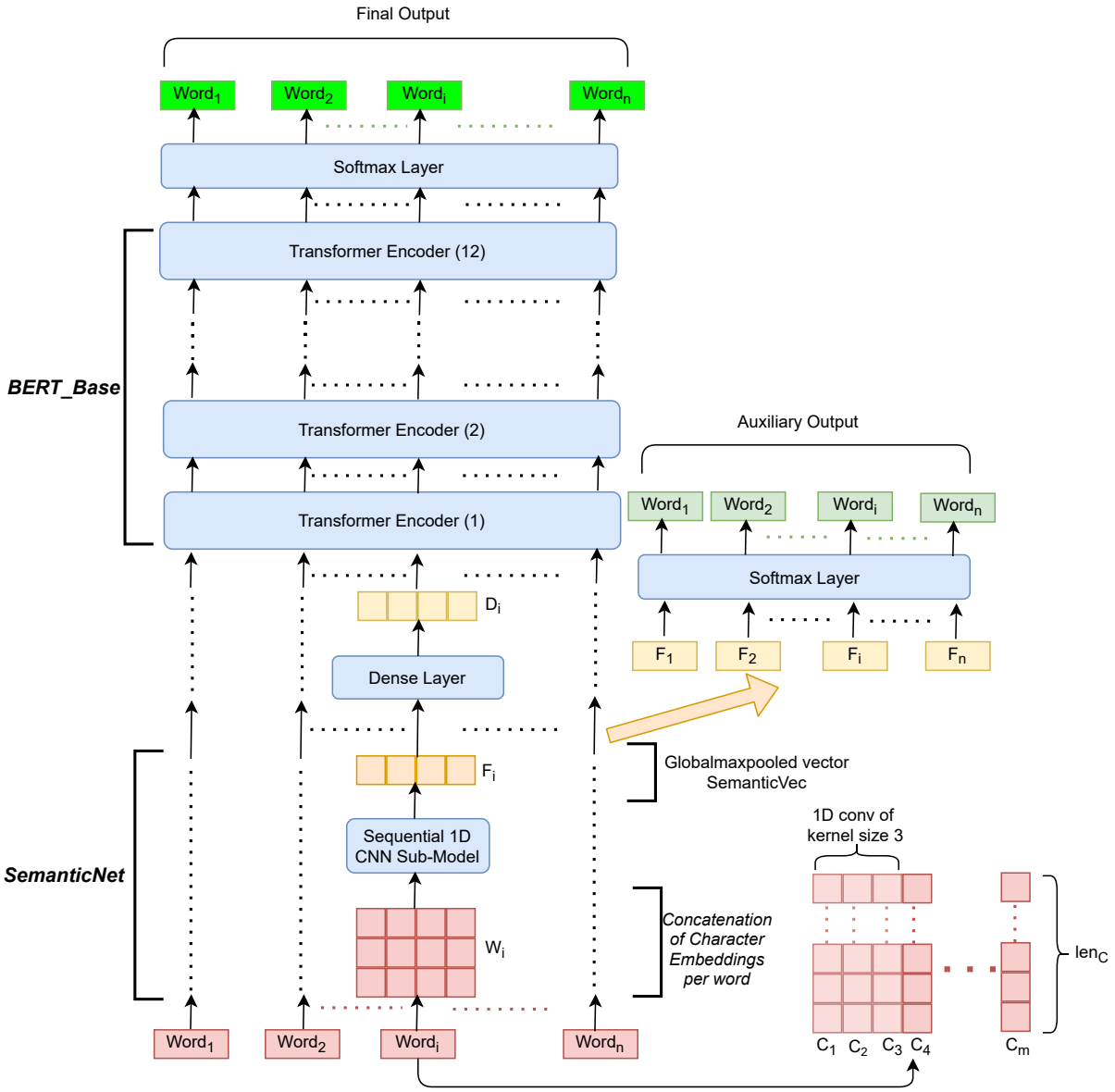


Figure 4: BSpell architecture details

fective pattern based word level spell checker, we introduce an auxiliary loss based secondary branch in *BSpell*. Each of the n *SemanticVecs* obtained from the n input words are passed parallelly on to a Softmax layer without any further modification. The outputs obtained from this branch are probability vectors similar to the main branch output. The total loss of *BSpell* can be expressed as: $L_{Total} = L_{Final} + \lambda \times L_{Auxiliary}$. We want our final loss to have greater impact on model weight update as it is associated with the final prediction made by *BSpell*. Hence, we impose the constraint $0 < \lambda < 1$. This secondary branch of *BSpell* does not have any Transformer encoders through which the input words can interact to produce context information. The prediction made from this branch is dependent solely on misspelled word pattern extracted by *SemanticNet*. This enables *SemanticNet* to learn more meaningful word representation.

3.3 BERT Hybrid Pretraining

In contemporary BERT pretraining methods, each input word $Word_i$ maybe kept intact or maybe replaced by a default mask word in a probabilistic manner (Devlin et al., 2018; Liu et al., 2019). BERT has to predict the masked words. Mistakes from the BERT side will contribute to loss value accelerating backpropagation based weight update. In this process, BERT learns to fill in the gaps, which in turn teaches the model language context.

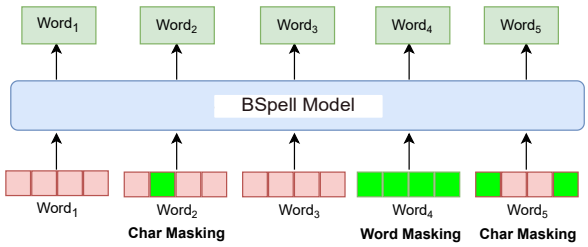


Figure 5: BERT hybrid pretraining

Sun et al. (2020) proposed incremental ways of pretraining the model for new NLP tasks. We take a more task specific approach for masking. In SC, recognizing noisy word pattern is important. But there is no provision for that in contemporary pretraining schemes and so, we propose hybrid masking (see Figure 5). Among n input words in a sentence, we randomly replace n_W words with a mask word $Mask_W$. Among the remaining $n - n_W$ words, we choose n_C words for character masking. We choose m_C characters at random from a word having m characters to be replaced by a mask character $Mask_C$ during character masking. Such masked characters introduce noise in words and helps BERT to understand the probable semantic meaning of noisy/ misspelled words.

4 Experimental Setup

4.1 Implemented Pretraining Schemes

We have experimented with three types of masking based pretraining schemes. During **word masking** we randomly select 15% words of a sentence and replace those with a fixed mask word. During **character masking**, we randomly select 50% words of a sentence. For each selected word, we randomly mask 30% of its characters by replacing each of them with a special mask character. Finally, during **hybrid masking**, we randomly select 15% words of a sentence and replace them with a fixed mask word. We randomly select 40% words from the remaining words. For these selected words, we randomly mask 25% of their characters.

4.2 Dataset Specification

We have used one Bangla and one Hindi corpus with over 5 million (5 M) sentences for BERT pretraining (see Table 1). Bangla pretraining corpus consists of Prothom Alo [2] articles dated from 2014-2017 and BDnews24 [3] articles dated from 2015-2017. The Hindi pretraining corpus consists of Hindi Oscar Corpus [4], preprocessed Wikipedia articles [5], HindiEnCorp05 dataset [6] and WMT Hindi News Crawl data [7] (all of these are publicly available corpus). We have used Prothom-Alo 2017 online newspaper dataset for Bangla SC training and validation purpose. Our errors in this corpus have been produced synthetically using the probabilistic algorithm described by Sifat et al. (2020). We further validate our baselines and proposed methods on Hindi open source SC dataset, namely ToolsForIL (Etoori et al., 2018). For real error dataset, we have collected a total of 6300 sentences from Nayadiganta [8] online newspaper. Then we have distributed the dataset among ten participants. They have typed (in regular speed) each correct sentence using English QWERTY keyboard producing natural spelling errors. It has taken 40 days to finish the labeling. Top words have been taken such that they cover at least 95% of the corresponding corpus.

4.3 *BSpell* Architecture Hyperparameters

SemanticNet sub-model of *BSpell* consists of a character level embedding layer producing a 40 size vector from each character, then 5 consecutive layers each consisting of 1D convolution (batch normalization and Relu activation in between each pair of convolution layers) and finally, a 1D global max pooling in order to obtain *SemanticVec* representation from each input word. The five 1D convolution layers consist of $(64, 2), (64, 3), (128, 3), (128, 3), (256, 4)$ convolution, respectively. The first and second element of each tuple denote number of convolution filters and kernel size, respectively. We provide a weight of 0.3 (λ value of loss function) to the auxiliary loss. The main branch of *BSpell* is similar to BERT_Base (Gong et al., 2019) in terms of stacking 12 Transformer encoders. Attention outputs from each Transformer is passed through a dropout layer (Srivastava et al., 2014) with a dropout rate of 0.3 and then layer normalized (Ba et al., 2016). We use *Stochastic Gradient Descent (SGD)* Optimizer with a learning rate of 0.001 for our model weight update. We clip our gradient value and keep it below 5.0 to avoid gradient exploding problem.

[2] https://www.prothomalo.com/
[3] https://bangla.bdnews24.com/
[4] https://www.kaggle.com/abhishek/hindi-oscar-corpus
[5] https://www.kaggle.com/disisbig/hindi-wikipedia-articles-172k
[6] http://hdl.handle.net/11858/00-097C-0000-0023-625F-0
[7] https://www.aclweb.org/anthology/W19-5301
[8] https://www.dailynayadiganta.com/

Datasets	Unique Word	Unique Char	Top Word	Train Sample	Validation Sample	Unique Error Word	Error Word Percentage
Prothom-Alo Bangla Synthetic Error	262 K	73	35 K	1 M	200 K	450 K	52%
Bangla Real Error	14.5 K	73	_	4.3 K	2 K	10 K	36%
Bangla Pretrain Corpus	513 K	73	40 K	5.5 M	_	_	_
Hindi Synthetic Error Corpus (ToolsForIL)	20.5 K	77	15 K	75 K	16 K	5 K	10%
Hindi Pretrain Corpus	370 K	77	40 K	5.5 M	_	_	_

Table 1: Dataset specification details

5 Results and Discussion

5.1 Training and Validation Details

In case of Bangla SC, we randomly initialize the weights of model M. We use our large Bangla pretrain corpus for hybrid pretraining and get pretrained model M_{pre}. Next we split our benchmark synthetic spelling error dataset (Prothom-Alo) into 80%-20% training-validation set. We fine tune M_{pre} using the 80% training portion (obtaining fine tuned model M_{fine}) and report performance on the remaining 20% validation portion. We use the Bangla real spelling error dataset in two ways - (1) We do not fine tune M_{fine} on any of part of this data and use the entire dataset as an independent test set (result reported with the title *real error (no fine tune)*) (2) We split this real error dataset into 80%-20% training-validation and fine tune M_{fine} further using the 80% portion, then validate on the remaining 20% (result reported with the title *real error (fine tuned)*). In case of Hindi, the first two steps (pretraining and fine tuning) are the same. We have not constructed any real life spelling error dataset for Hindi. So, results are reported on the 20% held out portion of the benchmark dataset.

5.2 BSpell vs Contemporary BERT Variants

We start with **BERT Seq2seq** where the encoder and decoder portion consist of 12 stacked Transformers (Devlin et al., 2018). Predictions are made at character level. Similar architecture has been used in *FASpell* (Hong et al., 2019) for Chinese SC. A word is considered wrong if even one of its characters is predicted incorrectly. Hence character level seq2seq modeling achieves poor result (see Table 2). Moreover, in most cases during sentence level spell checking, the correct spelling of the i^{th} word of input sentence has to be the i^{th} word in the output sentence as well. Such constraint is difficult to follow through such architecture design. **BERT Base** consisting of stacked Transformer encoders has two differences from the design proposed by Cheng et al. (2020) - (i) We make predictions at word level instead of character level (ii) We do not incorporate any external knowledge about Bangla SC since such knowledge is not well established in the field. This approach achieves good performance in all four cases. **Soft Masked BERT** learns to apply specialized synthetic masking on error prone words in order to push the error correction performance of *BERT Base* further. The error prone words are detected using a GRU sub-model and the whole architecture is trained end to end. Although Zhang et al. (2020) implemented this architecture to make corrections at character level, our implementation does everything in word level. We have used popular FastText (Athiwaratkun et al., 2018) word representation for both *BERT Base* and *Soft Masked BERT*. **BSpell** shows decent performance improvement in all cases.

5.3 Comparing *BSpell* Pretraining Schemes

We have implemented three different pretraining schemes (details provided in Subsection 4.1) on *BSpell* before fine tuning on spell checker dataset. **Word masking** teaches *BSpell* context of a lan-

Spell Checker Architecture	Synthetic Error (Prothom-Alo)		Real-Error (No Fine Tune)		Real-Error (Fine Tuned)		Synthetic Error (Hindi)	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
BERT Seq2seq	31.6%	0.305	24.5%	0.224	29.3%	0.278	22.8%	0.209
BERT Base	91.1%	0.902	83%	0.823	87.6%	0.855	93.8%	0.923
Soft Masked BERT	92%	0.919	84.2%	0.832	88.1%	0.862	94%	0.933
BSpell	**94.7%**	**0.934**	**86.1%**	**0.859**	**90.1%**	**0.898**	**96.2%**	**0.96**

Table 2: Comparing BERT based variants. Typical word masking based pretraining has been used on all these variants. Real-Error (Fine Tuned) denotes fine tuning of the Bangla syn- thetic error dataset trained model on real error dataset, while Real-Error (No Fine Tune) means directly validating synthetic error dataset trained model on real error dataset without any further fine tuning.

Pretraining Scheme	Synthetic Error (Prothom-Alo)		Real-Error (No Fine Tune)		Real-Error (Fine Tuned)		Synthetic Error (Hindi)	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Word Masking	94.7%	0.934	86.1%	0.859	90.1%	0.898	96.2%	0.96
Character Masking	95.6%	0.952	85.3%	0.851	89.2%	0.889	96.4%	0.963
Hybrid Masking	**97.6%**	**0.971**	**87.8%**	**0.873**	**91.5%**	**0.911**	**97.2%**	**0.97**

Table 3: Comparing *BSpell* exposed to various pretraining schemes

guage through a fill in the gaps sort of approach. SC is not all about filling in the gaps. It is also about what the writer wants to say, i.e. being able to predict a word even if some of its characters are blank (masked). **Character masking** takes a more drastic approach by completely eliminating the fill in the gap task. This approach masks a few of the characters residing in some of the input words of the sentence and asks *BSpell* to predict these noisy words' original correct version. The lack of context in such pretraining scheme puts negative effect on performance over real error dataset experiments, where harsh errors exist and context is the only feasible way of correcting such errors (see Table 3). **Hybrid masking** focuses both on filling in word gaps and on filling in character gaps through prediction of correct word and helps *BSpell* achieve SOTA performance.

5.4 *BSpell* vs Possible LSTM Variants

BiLSTM is a many to many bidirectional LSTM (two layers) that takes in all n words of a sentence at once and predicts their correct version as output (Schuster and Paliwal, 1997). During SC, *BiLSTM* takes in both previous and post context into consideration besides the writing pattern of each word and shows reasonable performance (see Table 4). In **Stacked BiLSTM**, we stack twelve many to many bidirectional LSTMs instead of just two. We see marginal improvement in SC performance in spite of such large increase in parameter number. **Attn_Seq2seq** LSTM model utilizes attention mechanism at decoder side (Bahdanau et al., 2014). This model takes in misspelled sentence characters as input and provides the correct sequence of characters as output (Etoori et al., 2018). Due to word level spelling correction evaluation, this model faces the same problems as *BERT Seq2seq* model discussed in Subsection 5.2. Proposed **BSpell** outperforms these models by a large margin.

5.5 Ablation Study

BSpell has three unique features - (1) secondary branch with auxiliary loss (possible to remove this branch), (2) 1D CNN based SemanticNet submodel (can be replaced by simple *Byte Pair Encoding (BPE)* (Vaswani et al., 2017)) and (3) hybrid pretraining (can be replaced by word masking based pretraining). Table 5 demonstrates the results we obtain after removing any one of these features. In all cases, the results show a downward trend compared to the original architecture.

5.6 Existing Bangla Spell Checkers vs *BSpell*

Phonetic rule based SC takes a Bangla phonetic rule based hard coded approach (Saha et al., 2019), where a hybrid of Soundex (UzZaman and Khan, 2004) and Metaphone (UzZaman and Khan, 2005) algorithm has been used. *Clustering* based SC on the other hand follows some predefined rules

Spell Checker Architecture	Synthetic Error (Prothom-Alo)		Real-Error (No Fine Tune)		Real-Error (Fine Tuned)		Synthetic Error (Hindi)	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
BiLSTM	81.9%	0.818	78.3%	0.781	81.1%	0.809	81.2%	0.809
Stacked BiLSTM	83.5%	0.832	80.1%	0.80	82.4%	0.822	82.7%	0.824
Attn_Seq2seq (Char)	20.5%	0.178	15.4%	0.129	17.3%	0.152	22.7%	0.216
BSpell	**97.6%**	**0.971**	**87.8%**	**0.873**	**91.5%**	**0.911**	**97.2%**	**0.97**

Table 4: Comparing LSTM based variants with hybrid pretrained *BSpell*. FastText word representation has been used with LSTM portion of each architecture.

BSpell Variants	Synthetic Error (Prothom-Alo)		Real-Error (No Fine Tune)		Real-Error (Fine Tuned)		Synthetic Error (Hindi)	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Original	**97.6%**	**0.971**	**87.8%**	**0.873**	**91.5%**	**0.911**	**97.2%**	**0.97**
No Aux Loss	96.3%	0.96	86.9%	0.865	90.5%	0.90	95.4%	0.949
No SemanticNet	94.5%	0.94	85.7%	0.848	89.2%	0.885	95.2%	0.95
No Hybrid Pretrain	94.7%	0.934	86.1%	0.859	90.1%	0.898	96.2%	0.96

Table 5: Comparing *BSpell* with its variants created by removing one of its novel features

Spell Checker	Synthetic Error (Prothom-Alo)		Real-Error (No Fine Tune)	
	ACC	F1	ACC	F1
Phonetic	61.2%	0.582	43.5%	0.401
Clustering	52.3%	0.501	44.2%	0.412
BSpell	**97.6%**	**0.971**	**87.8%**	**0.873**

Table 6: Existing Bangla spell checkers vs *BSpell*

on word cluster formation, distance measurement and correct word suggestion (Mandal and Hossain, 2017). Since these two SCs are not learning based, fine tuning is not applicable for them. They do not take misspelled word context into consideration while correcting that word. As a result, their performance is poor especially in Bangla real error dataset (see Table 6). *BSpell* outperforms these Bangla SCs by a wide margin.

5.7 Is *BSpell* Language Specific?

BSpell has originally been designed keeping the unique characteristics of Sanskrit originated languages such as Bangla and Hindi in mind. Here we see how this model performs on English which is very different from Bangla in terms of structure. We experiment on an English spelling error dataset published by Jayanthi et al. (2020). The training set consists of 1.6 million sentences. The authors created a confusion set consisting of 109K misspelled-correct word pairs for 17K popular English words. 20% of the words of the training set have been converted to spelling error based on this confusion set. The authors created BEA-60K test set from BEA-2019 shared task consisting of natural English spelling errors. The best correction rate achieved by the authors was around 80% using LSTM based ELMo model, whereas *BSpell* has achieved a correction rate of 86.2%. We have also experimented with *BERT_Base* model on this test set where we have used byte pair encoding as word representation. *BERT_Base* has achieved an error correction rate of 85.6%. It is clear that *BSpell* and *BERT_Base* do not have that much difference in performance when it comes to English compared to Bangla and Hindi.

5.8 Effectiveness of *SemanticNet*

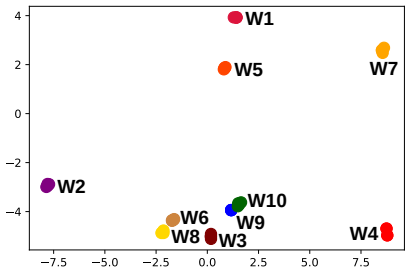


Figure 6: Visualizing *SemanticVec* representation of 10 popular words with their error variants

The main motivation behind the inclusion of

SemanticNet in *BSpell* is to obtain vector representations of error words as close as possible to their corresponding correct words. We take 10 frequently occurring Bangla words and collect three real life error variations of each of these words. We produce *SemanticVec* representation of all 40 of these words using *SemanticNet*. We use principal component analysis (PCA) (Shlens, 2014) on each of these *SemanticVecs* and plot them in two dimensions. Finally, we implement K-Means Clustering algorithm using careful initialization with $K = 10$ (Chen and Xia, 2009). Figure 6 shows the 10 clusters obtained from this algorithm. Each cluster consists of a popular word and its three error variations. In all cases, the correct word and its three error versions are so close in the graph plot that they almost form a single point.

6 Conclusion

In this paper, we have proposed a SC named *BSpell* for Bangla and Hindi language. *BSpell* uses *SemanticVec* representation of input misspelled words and a specialized auxiliary loss for the enhancement of spelling correction performance. The model exploits the concept of hybrid masking based pretraining. We have also investigated into the limitations of existing Bangla SCs as well as other SOTA SCs proposed for high resource languages. *BSpell* has two main limitations - (a) it cannot handle accidental merge or split of words and (b) it cannot correct misspelled rare words. A potential research direction can be to eradicate these limitations by designing models that can perform prediction at sub-word level which includes white space characters and punctuation marks.

7 Limitations

BSpell model provides a word for word correction, i.e., number of input words and number of output words have to be exactly the same. Unfortunately, during accidental word merging or word splitting, number of input and output words differ and so in such cases *BSpell* will fail in resolving such errors. This type of error is more common in Chinese language. The advantage for us is that this type of error is rare in Bangla and Hindi as the words of these languages are clearly spaced in sentences. So, people will rarely perform accidental merge or split of words. Another limitation is that *BSpell* has been trained to correct only the top Bangla and Hindi words that cover 95% of the entire corpus. As a result, this spell checker will face problems while correcting spelling errors in rare words. For such rare words, *BSpell* simply provides *UNK* as output which means that it is not sure what to do with these words. An advantage here is that most of these rare words are some form of proper nouns which should not be corrected and should ideally be left alone as they are. For example, someone may have an uncommon name. We do not want our model to correct that person's name to some commonly used name.

An immediate research direction is to overcome the limitations of the proposed method. A straightforward way of dealing with the word merge, word split and rare word correction problem is to model spelling errors at character level (sequence-to-sequence type approach). We have taken this trivial attempt and have failed miserably (see the performance reported in the first row of Table 2). Solving these problems while maintaining the current spelling correction performance of *BSpell* can be a challenge. Another interesting future direction is to investigate on personalized Bangla and Hindi spell checker which has the ability to take user personal preference and writing behaviour into account. The main challenge here is to effectively utilize user provided data that must be collected in an online setting. Recently, deep learning based automatic grammatical error correction has gained a lot of attention in English language (Chollampatt and Ng, 2018), (Chollampatt and Ng, 2017), (Stahlberg and Kumar, 2021). SOTA grammar correction models developed for English can be trained and tested on Bangla and Hindi spell checking tasks as part of future research effort. Such benchmarking studies can play a vital role in pushing the boundaries of low resource language correction automation.

References

Ben Athiwaratkun, Andrew Gordon Wilson, and Anima Anandkumar. 2018. Probabilistic fasttext for multi-sense word embeddings. *arXiv preprint arXiv:1806.02901*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Zhang Chen and Shixiong Xia. 2009. K-means cluster-

ing algorithm with improved initial center. In *2009 Second International Workshop on Knowledge Discovery and Data Mining*, pages 790–792. IEEE.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. Spellgcn: Incorporating phonological and visual similarities into language models for chinese spelling check. *arXiv preprint arXiv:2004.14166*.

Shamil Chollampatt and Hwee Tou Ng. 2017. Connecting the dots: Towards human-level grammatical error correction. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 327–333.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. 2018. Automatic spelling correction for resource-scarce languages using deep learning. In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152, Melbourne, Australia. Association for Computational Linguistics.

Linyuan Gong, Di He, Zhuohan Li, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Efficient training of bert by progressively stacking. In *International Conference on Machine Learning*, pages 2337–2346. PMLR.

Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu. 2019. Faspell: A fast, adaptable, simple, powerful chinese spell checker based on dae-decoder paradigm. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 160–169.

Sadidul Islam, Mst Farhana Sarkar, Towhid Hussain, Md Mehedi Hasan, Dewan Md Farid, and Swakkhar Shatabda. 2018. Bangla sentence correction using deep neural network based sequence to sequence learning. In *2018 21st International Conference of Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. Neuspell: A neural spelling correction toolkit. *arXiv preprint arXiv:2010.11085*.

Nur Hossain Khan, Gonesh Chandra Saha, Bappa Sarker, and Md Habibur Rahman. 2014. Checking the correctness of bangla words using n-gram. *International Journal of Computer Application*, 89(11).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Prianka Mandal and BM Mainul Hossain. 2017. Clustering-based bangla spell checker. In *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 1–6. IEEE.

Jan Noyes. 1983. The qwerty keyboard: A review. *International Journal of Man-Machine Studies*, 18(3):265–281.

Sourav Saha, Faria Tabassum, Kowshik Saha, and Marjana Akter. 2019. *BANGLA SPELL CHECKER AND SUGGESTION GENERATOR*. Ph.D. thesis, United International University.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Jonathon Shlens. 2014. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

Md Habibur Rahman Sifat, Chowdhury Rafeed Rahman, Mohammad Rafsan, and Hasibur Rahman. 2020. Synthetic error dataset generation mimicking bengali writing pattern. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 1363–1366. IEEE.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. *arXiv preprint arXiv:2105.13318*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975.

Naushad UzZaman and Mumit Khan. 2004. A bangla phonetic encoding for better spelling suggesions. Technical report, BRAC University.

Naushad UzZaman and Mumit Khan. 2005. A double metaphone encoding for approximate name searching and matching in bangla.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Dingmin Wang, Yi Tay, and Li Zhong. 2019. Confusionset-guided pointer networks for Chinese spelling check. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5780–5785, Florence, Italy. Association for Computational Linguistics.

Jinhua Xiong, Qiao Zhang, Shuiyuan Zhang, Jianpeng Hou, and Xueqi Cheng. 2015. Hanspeller: a unified framework for chinese spelling correction. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language*.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked bert. *arXiv preprint arXiv:2005.07421*.

Advancing Bangla Punctuation Restoration by a Monolingual Transformer-Based Method and a Large-Scale Corpus

Mehedi Hasan Bijoy[1,†], **Mir Fatema Afroz Faria**[2,†], **Mahbub E Sobhani**[3]
Tanzid Ferdoush[3] **and Swakkhar Shatabda**[3,*]
[1]Aalto University, [2]North South University, [3]United International University
{mehedi.bijoy@aalto.fi, afroz.fariaa@gmail.com, msobhani171134@bscse.uiu.ac.bd,
ferdoushtanzid@gmail.com, and swakkhar@cse.uiu.ac.bd}
† denotes equal contributions
∗ denotes corresponding author

Abstract

Punctuation restoration is the endeavor of reinstating and rectifying missing or improper punctuation marks within a text, thereby eradicating ambiguity in written discourse. The Bangla punctuation restoration task has received little attention and exploration, despite the rising popularity of textual communication in the language. The primary hindrances in the advancement of the task revolve around the utilization of transformer-based methods and an openly accessible extensive corpus, challenges that we discovered remained unresolved in earlier efforts. In this study, we propose a baseline by introducing a monolingual transformer-based method named Jatikarok[1] , where the effectiveness of transfer learning has been meticulously scrutinized, and a large-scale corpus containing 1.48M source-target pairs to resolve the previous issues. The Jatikarok attains accuracy rates of 95.2%, 85.13%, and 91.36% on the BanglaPRCorpus, Prothom-Alo Balanced, and BanglaOPUS corpora, thereby establishing itself as the state-of-the-art method through its superior performance compared to BanglaT5 and T5-Small. Jatikarok and BanglaPRCorpus are publicly available at https://github.com/mehedihasanbijoy/Jatikarok-and-BanglaPRCorpus.

1 Introduction

The continuous effort to bridge the linguistic gap between human natural language and digital devices has propelled natural language processing (NLP) to its current level of advancement. Despite these advances in NLP, Bangla language processing continues to present significant challenges including multimodal complexities stemming from intricate language rules. Proper punctuation placement, particularly in the Bangla language, plays a pivotal role in further reducing this barrier and facilitating downstream Bangla natural language processing (BNLP) tasks.

Previous studies have highlighted the dominance of transformer-based models such as BERT (Fu et al., 2021), RoBERTa (Nagy et al., 2021), and ALBERT (Shi et al., 2021) and have showcased their efficacy in leveraging contextual information for punctuation restoration in high-resource languages. Additionally, architectural enhancements such as attention mechanisms has also shown good performance (Yi and Tao, 2019). Following the trend of domain-specific fine-tuning of pretrained models, alongside post-processing techniques, has also achieved close to adequate performance (Chordia, 2021). Cross-lingual augmentation strategies enhance transformer models for languages with diverse resources, which is nonexistent in (Alam et al., 2020). The study conducted by (Rahman et al., 2023) only restored four types of punctuation marks in the Bangla language. However, there are at least nine more punctuation marks that need to be addressed to exhaustively capture the meaning. Moreover, it should be pointed out that deep learning models may not be capable of covering a significant proportion of punctuation in cases where the corpus is comparably small (Monsur et al., 2022). However, transformer-centric approaches have started demonstratring impressive performance in various BNLP tasks, including grammar and spelling error correction (Bijoy et al., 2022). Surprisingly, transformer-based methods have yet to be applied in any studies for the Bangla punctuation restoration task. Consequently, in this study, we leverage the impressive capabilities of transformers and initiate an investigation into their unexplored potential for the task.

In this study, we propose a transformer-based method named Jatikarok for the task with a uniquely tailored architecture of six encoder and decoder layers, optimizing the balance between

[1]যতিকারক

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 18–25

model complexity and computational efficiency for Bangla punctuation restoration, while enhancing its performance through transfer learning, which consequently renders it a monolingual method. Furthermore, we introduce BanglaPRCorpus, a large-scale parallel corpus for the task consisting of 1.48 million source-target pairs. The contributions of this paper are summarized below:

- A monolingual transformer-based method called Jatikarok has paved the way for the first-ever monolingual transformer-based baseline in the Bangla punctuation restoration task.
- We benchmarked our proposed method on various corpora, and it has emerged as the state-of-the-art approach on two additional corpora, namely Prothom-Alo Balanced and BanglaOPUS, in addition to ours.
- The effectiveness of transfer learning from the Bangla grammatical error correction task has been scrutinized for its ability to capture intricate linguistic patterns within this specific task.
- A large-scale parallel corpus comprising 1.48M source-target pairs has been developed by incorporating 16 Bangla punctuation marks and made publicly available, making Bangla no longer a resource-scarce language for the task.

The subsequent sections of the paper are organized as follows: Section 2 presents an in-depth analysis of the background of Bangla punctuation restoration; the process of constructing our corpus is expounded upon in Section 3; Section 4 elucidates the architecture of our proposed method; Section 5 presents the tangible results derived from our empirical study; Section 6 culminates our investigation by offering concluding remarks and outlining potential avenues for future research.

2 Literature Review

The task of punctuation restoration has garnered widespread attention, leading to the emergence of novel insights within methods and datasets. We delve into an examination of the recent studies conducted for punctuation restoration. Our extensive studies identified several contemporary transformer-based and deep learning methods in the realm of Bangla punctuation restoration tasks and in various high and low-resource languages such as Transformer (Lai et al., 2023; Nguyen et al., 2019; Wu et al., 2022), RNN (Rahman et al., 2023; Kim, 2019) and Hybrid (Yi et al., 2020; Bakare et al., 2023).

Among RNN-based methods (Rahman et al., 2023) proposed a novel approach comprised of a bidirectional recurrent neural network (BRNN) model with an attention mechanism. The authors trained a large Bangla dataset focusing specifically on predicting the exclamation mark and achieved 96.8% accuracy with various post-processing techniques. Likewise, (Kim, 2019) took a similar approach to solve the task.

The advent of NLP has seen the employment of transformer-based methods where M-BERT, BERT, RoBERTa, BioBERT, and ELECTRA have been utilized (Sunkara et al., 2020; Huang et al., 2021). (Alam et al., 2020) explored transformer-based language models to restore punctuation and improved Bangla training and evaluation data whereas (Monsur et al., 2022) utilized inadequate supervision and proposed a unique method for acquiring dialogue data in languages with few resources and evaluated the dataset by finetuning BanglaBERT (Bhattacharjee et al., 2022). Predicting punctuation for sequences instead of individual tokens by utilizing RoBERTa-base, (Courtland et al., 2020) proposed an innovative approach to solving the task. (Guerreiro et al., 2021) followed a homogeneous approach and proposed a contextual embedding-based punctuation prediction model. RoBERTa outperformed other transformer-based models in the comparison.

Our study has also revealed that the implementation of hybrid models has yielded exceptional results in addressing complex natural language processing tasks. By taking advantage of the evaluation of different BERT transformer models using LSTM and GRU with a linear neural network layer (Bakare et al., 2023) proposed a robust punctuation restoration algorithm. Besides, (Makhija et al., 2019) proposed a LSTM-CRF(Conditional Random Field) model that uses pre-trained BERT embeddings to make tagging decisions that take step interdependence into account to solve the punctuation restoration problem.

A thorough analysis found that transformer-based methods outperform RNN-based ones.

While RNNs may struggle with feature coverage and handling large datasets, transformers do not face these challenges. However, a downside of transformer-based approaches is that they require huge datasets to perform effectively.

3 Corpus Creation

We consider a total of 16 distinct punctuation marks, including period('।'), comma(','), exclamation mark ('!'), question mark ('?'), semicolon (';'), Bangla colon ('ঃ'), colon (':'), double quotation mark ("), single quotation mark ('), hyphen ('-'), opening parenthesis ('('), closing parenthesis (')'), opening curly brace ('{'), closing curly brace ('}'), opening square bracket ('['), and closing square bracket(']'), to curate the corpus. The details of these punctuation marks are delineated as follows:

Period(।): A definitive halt, denoting the terminus of a sentence in Bangla.

Comma(,): An eloquent separator, orchestrating rhythm within lists, crafting succinct pauses, and clarifying sentence structure.

Exclamation Mark(!): A linguistic exclamation point, amplifying emphasis, evoking astonishment or fervor, typically crowning the culmination of sentences.

Question Mark(?): An inquisitive note, framing direct queries. Its presence, positioned at sentence conclusions, signifies an inquest for insight.

Semicolon(;): A poised pause, surpassing a comma's subtlety yet shying from a full stop's grandeur. It adroitly links kindred concepts.

Bangla Colon(ঃ): A signal which indicates that what comes next is elaborating on, explaining, or providing examples related to the preceding clause or phrase.

Colon(:): An introducer of elucidation, explanations, and verbatim passages within sentences, colonizes text with structured context.

Double Quotation Mark("): A textual embrace for direct discourse or citations in Bangla script, encapsulating borrowed expressions.

Single Quotation Mark('): An enigmatic gesture, encircling quotes within quotes or indicating nuanced semantics, an annotation of depth.

Hyphen(-). A linguistic bridge, tethering word parts, fusing compound lexemes, and demarcating ranges with subtle precision.

Opening Parenthesis(( ): A grammatical cradle, ensconcing auxiliary or clarifying content, nurturing intricate sentence ecosystems.

Closing Parenthesis( )): A tender closure, rounding out preceding parenthetical, nurturing textual harmony and enclosure.

Opening Curly Brace({): A technical flourish, sometimes corralling supplementary information or code within contexts of expertise.

Closing Curly Brace(}): A counterpart to the opening brace, it brings closure, marking the ambit of enclosed insights or code.

Opening Square Bracket([): A gateway to lists, references, and augmented text in Bangla, welcoming expanded textual horizons.

Closing Square Bracket(]): The ultimate gatekeeper, sealing the opening bracket's portal, concluding augmented textual exploration.

3.1 Data Sourcing

We source our data from a publicly available Bangla paraphrase corpus (Akil et al., 2022). This dataset comprises approximately 466,000 carefully produced pairs of artificially created rephrased sentences in the Bangla language. These rephrased sentences have been meticulously crafted to uphold both the meaning's coherence and the diversity of sentence structure, guaranteeing their outstanding quality.

3.2 Data Preprocessing

We consider 72 distinct characters that frequently occur in Bangla text denoted as $DC = \{DC_1, DC_2, ..., DC_{72}\}$, in addition to 16 Bangla punctuation marks represented by $PM = \{PM_1, PM_2, ..., PM_{16}\}$ and a space SP, resulting in a set of 89 Bangla characters represented by $C = \{DC + PM + SP\} = \{C_1, C_2, ..., C_{89}\}$. Next, we take into account each of the sentences indicated as $S = \{S_1, S_2, ..., S_N\}$, where N represents the number of characters in the sentence. We iterate through each of the characters $S_i \in S$ and remove any character that is not present in the unique character set C.

3.3 Punctuation Removal Procedure

We randomly remove N punctuation marks from a sentence S, based on their availability, where $N >= 1$ & $N <= 10$. To achieve this, we follow these steps: (Step 1) Initially, we count the number of punctuation marks, P_{count}, present in the sentence. If P_{count} is less than the number of punctuation marks we intend to remove from the sentence, we simply skip the sentence. (Step 2) Otherwise,

to remove a punctuation mark $PM_i \in PM$, we begin by shuffling the list of punctuations, PM. (Step 3) Proceeding to the next step, we iterate through the list of punctuations, PM, and determine whether the sentence contains the specific punctuation mark, PM_i. (Step 4) If the punctuation mark is present such that $PM_i \in S$, we remove it from the sentence and continue with the process. (Step 5) Finally, we repeat these steps from 1 to 4 for N times to achieve the removal of the desired N punctuation marks from a sentence.

3.4 Corpus Statistic

Our proposed Bangla punctuation restoration corpus (BanglaPRcorpus) consists of 1.48 million source-target pairs. In these pairs, the source sentences lack punctuation, while the target sentences are the corrected versions where missing punctuation is restored. To do so, we systematically eliminated punctuation marks in varying quantities, ranging from 1 to 10, within each sentence. Moreover, the minimum, maximum, and average number of words in a sentence of our corpus is 2, 127, and 12.9, respectively.

4 Methodology

4.1 Problem Formulation & Overview

Consider two sequences of tokens, $X_I = \{x_1, x_2, ..., x_n\}$, and $Y_I = \{y_1, y_2, ..., y_k\}$, where X_I represents an erroneous input sequence with missing punctuation marks, and Y_I represents the corresponding corrected sequence with punctuation marks restored. The encoder ($E(\cdot)$) of our method takes an erroneous input sentence X_I, which is first tokenized using a pre-trained tokenizer ($T(\cdot)$), and generates a representative vector of the sentence, denoted as $V = [V_1, V_2, ..., V_{512}]$. Subsequently, the decoder ($D(\cdot)$) utilizes the representative vector V, along with the previously generated tokens, to autoregressively generate the corresponding correct sentence. The entire procedure can mathematically be abbreviated as follows:

$$\hat{Y} = D((E(T([X_I]), W^E), D_{out}^{t-1}), W^D) \quad (1)$$

4.2 Motivations

Bangla is the fifth (Bhattacharyya et al., 2023) most spoken language, considering the number of speakers. Beyond mere documentation, Bangla serves multifarious communicative purposes, highlighting its diverse utility. A method aimed at enhancing typing proficiency by rectifying misused punctuation could offer substantial advantages. The endeavor of punctuation restoration holds significant importance to its enhancement of text lucidity and interpretability, thus circumventing potential ambiguities. Consequently, it contributes to the amelioration of downstream NLP tasks.

4.3 Jatikarok

In this section, we provide the details on Jatikarok.

4.3.1 Encoder

Given an input sequence of tokens $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, where n is the sequence length, we assigned unique discrete values to each word. We ensured uniform input dimension by expanding each input sequence $\mathbf{X}_i$ by incorporating padding. Subsequently, each token, x_i, undergoes an embedding layer $\mathbf{E}$ to convert discrete inputs into continuous vector representations using a trainable matrix in a latent space, such that $\mathbf{E}_i = \mathbf{Embed}(\mathbf{x_i})$. Notably, these matrices are fine-tuned via backpropagation during training to minimize the loss. These embeddings are subsequently combined with positional encodings $\mathbf{PE}$ to account for token order where $\mathbf{PE}_i$ represents the positional encoding for x_i. The combined embeddings, denoted as $\mathbf{Z}_i = \mathbf{E}_i + \mathbf{PE}_i$, are then fed into a stack of K identical layers, each composed of two main components: a multi-head self-attention mechanism and position-wise feedforward networks. The self-attention mechanism computes weighted representations for each token by attending to all tokens in the sequence $\mathbf{X}$ using learnable query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) vectors. This self-attention mechanism is defined as follows (Vaswani et al., 2017):

$$\mathbf{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{\mathbf{d_k}}})\mathbf{V} \quad (2)$$

The self-attention mechanism calculates weighted representations of each token by considering interactions with all other tokens in the sequence, enabling the capture of contextual dependencies. The position-wise feedforward networks introduce non-linearity through two linear transformations followed by a non-linear activation function, ReLU, enhancing the acquired representations. The outputs of each layer are sequentially

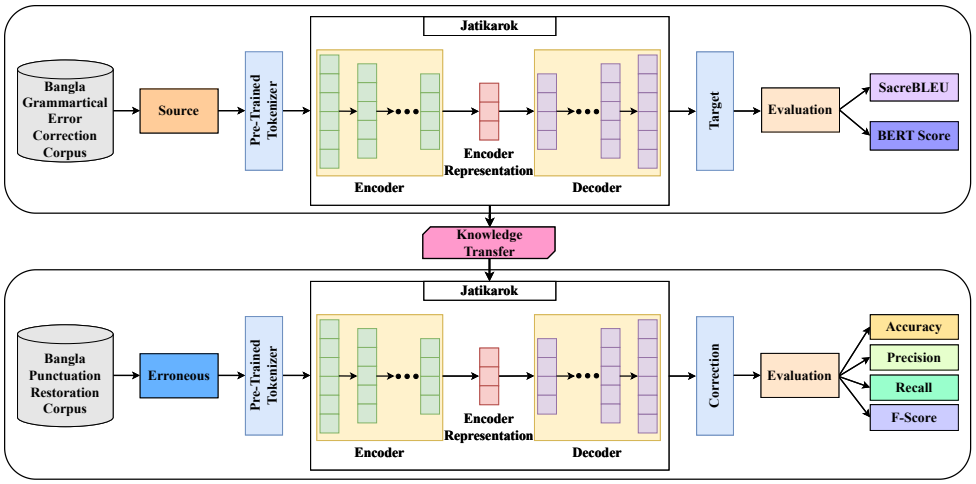


Figure 1: **(Top)** Jatikarok is initially trained on the Bangla Grammatical Error Correction (BGEC) task. **(Middle)** The insights acquired during the BGEC training are preserved for subsequent knowledge transfer to the Bangla Punctuation Restoration (BPR) task. **(Bottom)** Jatikarok is then fine-tuned on BPR corpora, leveraging the knowledge gleaned from the BGEC task.

propagated through the stack of K identical layers, yielding refined representations that encode both local and global dependencies, incorporating rich contextualized portrayals of the input sequence $\mathbf{X}$.

4.3.2 Decoder

Firstly, the target sequence, which is denoted as $\mathbf{Y} = \{y_1, y_2, \ldots, y_m\}$, where m is the sequence length, is embedded into a latent space using learned embeddings: $\mathbf{E}y_i = \mathbf{Embed}(y_i)$. To convey information about token order, positional encodings $\mathbf{PE}y_i$ are added to these embeddings ($\mathbf{E}y_i$). The resulting embeddings $\mathbf{Z}y_i = \mathbf{E}y_i + \mathbf{PE}y_i$ are then passed through a stack of L similar decoder layers, each composed of two primary components: a masked multi-head self-attention mechanism and position-wise feedforward networks. The computation of masked multi-head self-attention follows the same equation as regular multi-head self-attention (as given in Equation 2), with the crucial distinction that it enforces a restriction preventing the model from attending to tokens that occur in the future within the sequence. In contrast, the position-wise feedforward networks introduce non-linearity through linear transformations followed by a non-linear activation function (ReLU), enhancing the learned representations similar to the encoder's feedforward networks. The obtained representations from each layer are sequentially propagated through the stack of L similar decoder layers, resulting in refined target sequence representations denoted as $\mathbf{Y}$.

4.3.3 Hyperparameters

To maintain consistency, a hidden size dimension of 512 is employed across all layers within the encoder and decoder. Moreover, the feedforward neural network layers, which consist of 2048 neurons, contribute significantly to the model's depth and capacity. In order to mitigate the risks of overfitting, a dropout ratio of 0.1 is applied, thereby promoting robust and effective learning. The incorporation of the ReLU activation function introduces essential non-linearity to the network's computations. Throughout the training process, a learning rate of 5×10^{-5} is applied, and the model undergoes 100 epochs of training using the AdamW optimizer. This optimization process is carefully guided by the categorical cross-entropy loss function, which effectively steers the model towards achieving the desired translation outcomes.

5 Experimental Analysis

5.1 Datasets

- **BanglaPRCorpus (Ours).** It consists of 1,481,149 (1.48M) source-target pairs. We split the corpus into training and test sets, keeping 85% of the data in the training set and 15% in the test set, with each type of erroneous sentence, based on the number of punctuation removed, to maintain a balanced distribution. As a result, our training and test sets comprise 1,258,977 (1.26M) and 222,172 (222.1K) source-target instances, re-

Method	#Params.	BanglaPRCorpus					Prothom-Alo Balanced					BanglaOPUS				
		ACC	PR	RE	F1	F0.5	ACC	PR	RE	F1	F0.5	ACC	PR	RE	F1	F0.5
BiLSTM (Rahman et al., 2023)	**11.54M**	–	–	–	–	–	–	0.594	0.44	0.506	–	–	0.546	0.394	0.458	–
BanglaT5	247.53M	83.94%	0.839	0.839	0.839	0.841	76.53%	0.77	0.77	0.77	0.783	80.66%	0.806	0.806	0.806	0.813
T5-Small	60.51M	72.67%	0.728	0.727	0.727	0.728	74.95%	0.74	0.75	0.75	0.761	74.81%	0.748	0.748	0.748	0.754
Jatikarok	74.36M	**95.2%**	**0.953**	**0.952**	**0.952**	**0.955**	**85.13%**	**0.85**	**0.851**	**0.845**	**0.852**	**91.36%**	**0.914**	**0.914**	**0.914**	**0.92**

Table 1: The juxtaposition of the quantitative performance of different existing methods across various corpora.

spectively.

- **Prothom-Alo Balanced (Rahman et al., 2023).** It encompasses a total of 80150 source-target pairs after our meticulous preprocessing. The corpus was partitioned into training and test sets by maintaining an 85% and 15% split. Consequently, the resultant training set and test set comprise 68128 and 12022 source-target pairs, respectively.

- **Bangla OPUS (Tiedemann, 2012).** Following a comprehensive text preprocessing phase, we identified a total of 877,299 source-target pairs within the corpus. Subsequently, we divided the corpus in an 85:15 ratio to establish distinct training and test sets. This division resulted in 745,705 pairs within the training set, while the test set comprised 131,594 pairs.

5.2 Baselines

- **BanglaT5(Akil et al., 2022).** It is a pre-trained language model developed by fine-tuning the T5(Raffel et al., 2020) architecture specifically for the purpose of Bangla paraphrase task.

- **T5-Small(Raffel et al., 2020).** It is a variant of the Text-To-Text Transfer Transformer (T5) architecture featuring a smaller number of parameters (≈70M) compared to larger versions of T5 (220M).

5.3 Performance Evaluation

We evaluate the effectiveness of our model in restoring punctuations with accuracy, precision, recall, and the F-beta score. Mathematically, accuracy(Eq. 3), precision(Eq. 4), recall(Eq. 5), and the Fβ score(Eq. 6) can be defined as follows:

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

$$Precision(PR) = \frac{TP}{TP + FP} \tag{4}$$

$$Recall(RE) = \frac{TP + TN}{TP + FN} \tag{5}$$

$$f\beta\ score = (1 + \beta^2) \times \frac{PR \times RE}{\beta^2 \times PR + RE} \tag{6}$$

Where TP, TN, FP, and FN mean True Positive, True Negative, False Positive, and False Negative.

5.4 Main Results

5.4.1 Quantitative Results

The quantitative performance of different transformer-based methods on various corpora has been presented in Table 1. Our proposed model, Jatikarok, demonstrates significant performance superiority over both BanglaT5 and T5-Small across all three corpora, establishing itself as the new state-of-the-art method. It surpasses BanglaT5 and T5-Small in all evaluation measures, including accuracy, precision, recall, F1 score, and F0.5 score. Our method outperforms BanglaT5, which is the second-best model in comparison, despite having a parameter size three times smaller. It achieves 11.26%, 8.6%, and 10.7% higher accuracy scores on the BanglaPRCorpus, Prothom-Alo Balanced, and BanglaOPUS corpora, respectively. However, for multiple punctuation marks removed in a sentence, we did not consider them in the metrics individually, rather we calculated the overall accuracy, precision, recall, and F-scores considering the whole sentence.

5.4.2 Qualitative Results

The qualitative performance of BanglaT5, T5-Small, and Jatikarok has been juxtaposed in Table 2, effectively highlighting the superiority of our Jatikarok over BanglaT5 and T5-Small. The examples in the table explicitly illustrate that as the number of missing punctuation marks increases in a sentence, the performance of other methods decreases, while our Jatikarok maintains better accuracy. For instance, all methods performed well when only one punctuation mark was missing in a sentence. As the number increases to two, only our Jatikarok correctly corrects the sentence. However, when punctuation marks increase rapidly, all

(Input) স্বাভাবিকভাবেই এটা তিনি মেনে নিতে পারেন নি
(BanglaT5) স্বাভাবিকভাবেই এটা তিনি মেনে নিতে পারেন নি। (✓)
(T5-Small) স্বাভাবিকভাবেই এটা তিনি মেনে নিতে পারেন নি। (✓)
(Jatikarok) স্বাভাবিকভাবেই এটা তিনি মেনে নিতে পারেন নি। (✓)
(Input) তিনি অনুর্ধ্ব১৯ বিশ্বকাপ খেলেন ২০০০ সালে
(BanglaT5) তিনি অনুর্ধ্ব১৯ বিশ্বকাপ খেলেন ২০০০ সালে। (×)
(T5-Small) তিনি অনুর্ধ্ব১৯ বিশ্বকাপ খেলেন ২০০০ সালে। (×)
(Jatikarok) তিনি অনুর্ধ্ব-১৯ বিশ্বকাপ খেলেন ২০০০ সালে। (✓)
(Input) কিছু বিবরণ অনুসারে এই সংখ্যা ১২০০০০১৩০০০০
(BanglaT5) কিছু বিবরণ অনুসারে, এই সংখ্যা ১২০০০০১৩০০০০। (×)
(T5-Small) কিছু বিবরণ অনুসারে এই সংখ্যা ১২০০০০১৩০০০০। (×)
(Jatikarok) কিছু বিবরণ অনুসারে, এই সংখ্যা ১২০,০০০,১৩০,০০০। (×)

Table 2: The qualitative performance of different transformer-based methods.

methods fail, as demonstrated in the last example. For an erroneous input "কিছু বিবরণ অনুসারে এই সংখ্যা ১২০০০০১৩০০০০", Jatikarok generated output "কিছু বিবরণ অনুসারে, এই সংখ্যা ১২০,০০০,১৩০,০০০।", where the actual correction is "কিছু বিবরণ অনুসারে, এই সংখ্যা ১২০,০০০-১৩০,০০০!", which is superior to the corrections made by the other two methods. It accurately reinstated a comma between two words in the middle of the sentence (...অনুসারে, এই...), a task where the other two methods failed. Moreover, it also added commas in the number (১২০,০০০,১৩০,০০০) to enhance readability, a feat the other two methods did not accomplish.

5.5 Ablation Study

Table 3 illustrates how model performance improves with larger corpus sizes. The corpus consisting of 1.5M instances displayed the most substantial performance, while the corpus containing 148.1K instances showed the least significant performance. The corpus consisting of 740.5K instances demonstrated intermediate performance, surpassing the smaller corpus size but falling short of the 1.5M corpus. A clear pattern emerges: larger corpus sizes correspond to improved performance. The 1.5M corpus achieved an impressive accuracy of 95.2%, surpassing the 740.5K corpus by 5.48%, and the 148.1K corpus by 11.89%.

Method	Corpus Size	Inference				
		Acc	PR	RE	F1	F0.5
Jatikarok	148.1K	83.31%	0.833	0.833	0.833	0.834
Jatikarok	740.5k	89.72%	0.897	0.897	0.896	0.897
Jatikarok	1.48M	95.2%	0.953	0.952	0.952	0.953

Table 3: The impact of the corpus size on our proposed method.

6 Conclusion

This study addressed the primary obstacle hindering the progress of the task by introducing a comprehensive baseline. Specifically, we introduced the groundbreaking Jatikarok, a monolingual transformer-based method meticulously designed to harness the power of transfer learning by adapting knowledge from Bangla grammatical error correction to effectively tackle intricate linguistic patterns inherent to this specific task. Furthermore, the efficacy of our proposed Jatikarok is validated across various corpora, solidifying its status as a state-of-the-art method for this task by outperforming BanglaT5 and T5-Small. In conjunction with the model, a substantial parallel corpus containing 1.48M source-target pairs has been made publicly accessible, which has been carefully curated by incorporating 16 Bangla punctuation marks. Consequently, this resource eliminates the scarcity of materials for Bangla, effectively transforming it into a language well-equipped for punctuation tasks. In our future study, we will empirically investigate the effectiveness of knowledge distillation through the transfer of knowledge from the multilingual model to our monolingual model.

References

Ajwad Akil, Najrin Sultana, Abhik Bhattacharjee, and Rifat Shahriyar. 2022. Banglaparaphrase: A high-quality bangla paraphrase dataset. *arXiv preprint arXiv:2210.05109*.

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Punctuation restoration using transformer models for high-and low-resource languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 132–142.

Adebayo Mustapha Bakare, Kalaiarasi Sonai Muthu Anbananthen, Saravanan Muthaiyah, Jayakumar Krishnan, and Subarmaniam Kannan. 2023. Punctuation restoration with transformer model on social media data. *Applied Sciences*, 13(3):1685.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Pramit Bhattacharyya, Joydeep Mondal, Subhadip Maji, and Arnab Bhattacharya. 2023. Vacaspati: A diverse corpus of bangla literature. *arXiv preprint arXiv:2307.05083*.

Mehedi Hasan Bijoy, Nahid Hossain, Salekul Islam, and Swakkhar Shatabda. 2022. Dpcspell: A transformer-based detector-purificator-corrector framework for spelling error correction of bangla and resource scarce indic languages. *arXiv preprint arXiv:2211.03730*.

Varnith Chordia. 2021. Punktuator: A multilingual punctuation restoration system for spoken and written text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 312–320.

Maury Courtland, Adam Faulkner, and Gayle McElvain. 2020. Efficient automatic punctuation restoration using bidirectional transformers with robust inference. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 272–279.

Xue-Yong Fu, Cheng Chen, Md Tahmid Rahman Laskar, Shashi Bhushan TN, and Simon Corston-Oliver. 2021. Improving punctuation restoration for speech transcripts via external data. *arXiv preprint arXiv:2110.00560*.

Nuno Miguel Guerreiro, Ricardo Rei, and Fernando Batista. 2021. Towards better subtitles: A multilingual approach for punctuation restoration of speech transcripts. *Expert Systems with Applications*, 186:115740.

Qiushi Huang, Tom Ko, H Lilian Tang, Xubo Liu, and Bo Wu. 2021. Token-level supervised contrastive learning for punctuation restoration. *arXiv preprint arXiv:2107.09099*.

Seokhwan Kim. 2019. Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7280–7284. IEEE.

Viet Dac Lai, Abel Salinas, Hao Tan, Trung Bui, Quan Tran, Seunghyun Yoon, Hanieh Deilamsalehy, Franck Dernoncourt, and Thien Huu Nguyen. 2023. Boosting punctuation restoration with data generation and reinforcement learning. *arXiv preprint arXiv:2307.12949*.

Karan Makhija, Thi-Nga Ho, and Eng-Siong Chng. 2019. Transfer learning for punctuation prediction. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 268–273. IEEE.

Syed Mostofa Monsur, Sakib Chowdhury, Md Shahrar Fatemi, and Shafayat Ahmed. 2022. Shonglap: A large bengali open-domain dialogue corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5797–5804.

Attila Nagy, Bence Bial, and Judit Ács. 2021. Automatic punctuation restoration with bert models. *arXiv preprint arXiv:2101.07343*.

Binh Nguyen, Vu Bao Hung Nguyen, Hien Nguyen, Pham Ngoc Phuong, The-Loc Nguyen, Quoc Truong Do, and Luong Chi Mai. 2019. Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging. In *2019 22nd conference of the oriental COCOSDA international committee for the co-ordination and standardisation of speech databases and assessment techniques (O-COCOSDA)*, pages 1–5. IEEE.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Habibur Rahman, Md Rezwan Shahrior Rahin, Araf Mohammad Mahbub, Md Adnanul Islam, Md Saddam Hossain Mukta, and Md Mahbubur Rahman. 2023. Punctuation prediction in bangla text. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(3):1–20.

Ning Shi, Wei Wang, Boxin Wang, Jinfeng Li, Xiangyu Liu, and Zhouhan Lin. 2021. Incorporating external pos tagger for punctuation restoration. *arXiv preprint arXiv:2106.06731*.

Monica Sunkara, Srikanth Ronanki, Kalpit Dixit, Sravan Bodapati, and Katrin Kirchhoff. 2020. Robust prediction of punctuation and truecasing for medical asr. *arXiv preprint arXiv:2007.02025*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yangjun Wu, Kebin Fang, and Yao Zhao. 2022. A context-aware feature fusion framework for punctuation restoration. *arXiv preprint arXiv:2203.12487*.

Jiangyan Yi and Jianhua Tao. 2019. Self-attention based model for punctuation prediction using word and speech embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7270–7274. IEEE.

Jiangyan Yi, Jianhua Tao, Ye Bai, Zhengkun Tian, and Cunhang Fan. 2020. Adversarial transfer learning for punctuation restoration. *arXiv preprint arXiv:2004.00248*.

Pipeline Enabling Zero-shot Classification for Bangla Handwritten Grapheme

Linsheng Guo
Preferred Networks .inc
linsho@preferred.jp

Md Habibur Rahman Sifat
The Hong Kong Polytechnic University
habib.sifat@connect.polyu.hk

Tashin Ahmed
AriSaf Tech Japan K.K.
tashin@arisaftech.co.jp

Abstract

This research investigates Zero-Shot Learning (ZSL), and proposes CycleGAN-based image synthesis and accurate label mapping to build a strong association between labels and graphemes. The objective is to enhance model accuracy in detecting unseen classes by employing advanced font image categorization and a CycleGAN-based generator. The resulting representations of abstract character structures demonstrate a significant improvement in recognition, accommodating both seen and unseen classes. This investigation addresses the complex issue of Optical Character Recognition (OCR) in the specific context of the Bangla language. Bangla script is renowned for its intricate nature, consisting of a total of 49 letters, which include 11 vowels, 38 consonants, and 18 diacritics. The combination of letters in this complex arrangement provides the opportunity to create almost 13,000 unique variations of graphemes, which exceeds the number of graphemic units found in the English language. Our investigation presents a new strategy for ZSL in the context of Bangla OCR. This approach combines generative models with careful labeling techniques to enhance the progress of Bangla OCR, specifically focusing on grapheme categorization. Our goal is to make a substantial impact on the digitalization of educational resources in the Indian subcontinent.

1 Introduction

OCR, a significant technological innovation, has revolutionized the processing and examination of textual content in the contemporary digital age. OCR technology, specifically designed for the purpose of identifying and converting printed or handwritten text into text that can be processed by machines, has facilitated the retrieval, searchability, and manipulation of vast quantities of information across various languages, including Bangla.

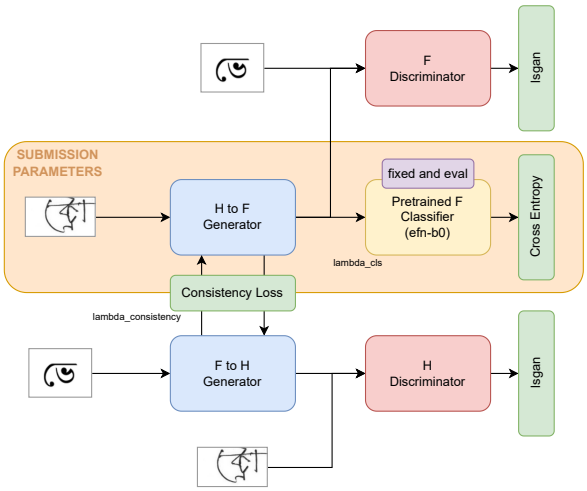


Figure 1: CycleGAN training module. The pre-trained font image classifier keeps the parameters fixed and only conveys gradients to the handwritten (H) to font (F) image generator. Additionally, the H to F image generator incorporates the CycleGAN architecture, enabling more natural generations from handwritten to fonts. *lambda_consistency* is a weight parameter that determines the amount of emphasis placed on loss of the classifier in addition to the loss of CycleGAN while performing zero-shot learning.

Bangla/Bengali has a rich and complicated writing system that makes it hard for OCR systems to read because of its complex ligatures, unique letters, and complicated calligraphy. OCR for Bangla characters aims to bridge the disparity between physical documents and digital databases by offering solutions for activities such as document digitization, language translation, and text analysis. This study is performed on a global AI competition, *Bengali.AI Handwritten Grapheme Classification* hosted by Kaggle and Bengali.AI where our study topped the final leaderboard. The main objective of this research was not exclusively to categorize handwritten characters into predetermined classes, but rather to construct a model with the ability to identify and classify classes that were not explic-

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 26–33

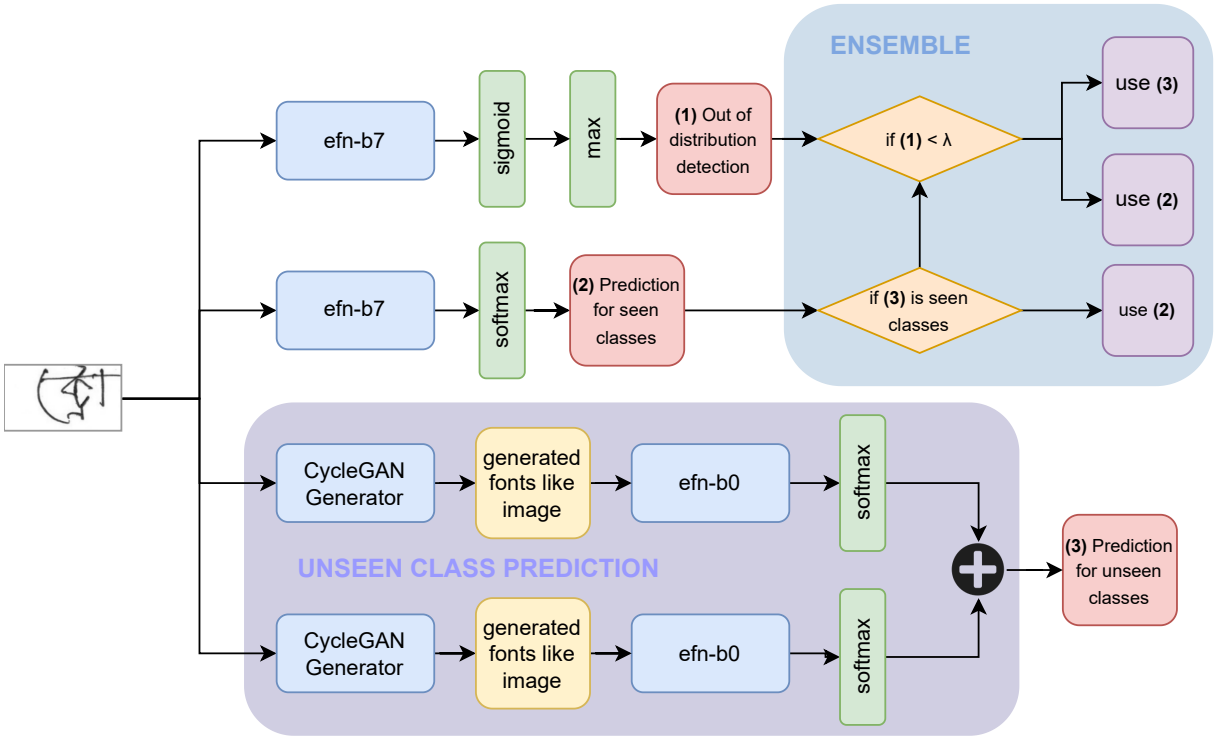


Figure 2: End to end visualization of the proposed architecture which is the top performing solution in the competition. It's created on 3 different models; (1) Out of distribution detection, (2) Seen class model and (3) Unseen class model. EfficientNet-B7 (efn-b7) is utilized as the backbone for Model 1 and 2. Innovative approach to predict unseen class is based on CycleGAN where the backbone is EfficientNet-B0 (efn-b0)

itly provided. Although the first categorization into three categories of components provided a useful foundation, we acknowledged the need for a more efficient method that involved extracting the underlying structures that could potentially arise within a character. In order to accomplish this, we utilized an innovative approach that involved the utilization of a generative model, more specifically a font image generation model based on CycleGAN. This model was employed to convert handwritten characters into images resembling fonts. When incorporated into a larger set of models leading to a handwriting classification system, this generative model produced font images that can be interpreted as intermediate features. The pixel-level representations successfully captured intricate details pertaining to the structure of the character, so effectively abstracting the fundamental qualities associated with the character. The development of this integrated system represents a significant milestone in our research, providing novel insights and enhanced functionalities in the fields of character recognition and classification.

2 Related Works

(Fuad Rezaur Rahman, 1994) introduced a groundbreaking approach that established the basis for Bangla OCR. This approach utilized pattern recognition techniques to accurately recognize handwritten Bangla characters. In a study, (Rahman et al., 2002) introduced a multi-stage recognition system for the identification of handwritten Bangla characters. In this study, the researchers form a cohort of characters and initially identify high-level attributes to classify the characters into groups. Subsequently, they proceed to identify low-level traits in order to accurately recognize the individual characters. (Chowdhury et al., 2002) introduced an initial approach utilizing neural networks for character recognition in printed text data, which was accompanied by some limitations. (Basu et al., 2009) introduces a novel hierarchical methodology for OCR specifically designed for handwritten Bangla words. The proposed approach effectively integrates segmentation and recognition techniques, thereby addressing the inherent difficulties associated with the presence of overlapping characters in the Bangla script. The study utilizes advanced methodologies such as the two-pass approach for

certain sections and MLP-based pattern classifiers, thereby enhancing the precision and comprehensiveness of OCR systems for handwritten Bangla text. A deep neural network (DNN) approach for Bangla OCR in the context of License Plate Recognition (LPR) was proposed by (Onim et al., 2022). In a recent publication by (Emon et al., 2022), a comprehensive analysis of thirteen papers on OCR for the Bangla language was published. The authors reported that the Bidirectional Long Short-Term Memory (BLSTM) model, as proposed by (Paul and Chaudhuri, 2019), demonstrated higher accuracy among the investigated approaches.

3 Dataset

In the realm of modern Bangla literature, a distinct collection of graphemes is frequently utilized, with their recognition being established by transcriptions derived from the Google Bangla ASR dataset as the primary point of reference (Alam et al., 2021). The dataset utilized for this objective is extensive, comprising 127,565 spoken utterances that were transcribed, resulting in a cumulative count of 609,510 words and 2,111,256 graphemes.

The dataset consists of 1,295 frequently used Bangla graphemes based on specific criteria, including occurrence in words and frequency. These graphemes comprise three main components: vowel diacritics, consonant diacritics, and grapheme roots. Vowel diacritics, represented by 11 classes, are typically found at the end of Unicode strings, with a null diacritic for cases of absent vowels. Consonant diacritics, forming diverse combinations, resulted in 8 classes. The remaining grapheme roots, including vowels, consonants, and conjuncts, are limited to 168 classes based on their prevalence in everyday language.

The painstaking compilation of metadata obtained from several sources has been a great resource for conducting comprehensive investigations into the relationship between handwriting and various categories of metadata. It is noteworthy to mention that the metadata pertaining to the training set has been made publicly accessible; however, access to the metadata of the test set can be acquired through a formal request to the authority (Alam et al., 2021).

The dataset has been made available to the public domain as a fundamental element of the *Bengali.AI Handwritten Grapheme Classification* Kaggle competition[1]. In this dataset, a meticulous distribution was implemented, whereby 200,840 samples were assigned to the training set, 98,661 samples were allocated to the public test set, and 112,381 samples were selected for the private test set. Significantly, a rigorous standard was implemented to guarantee the absence of any duplication in contributions within these sets. It is worth mentioning that the graphemes that occur less frequently were predominantly allocated to the private test set, and none of them were incorporated into the training subset. During the duration of the competition, players strive to improve their performance by analyzing the results obtained from the public test set. On the other hand, the outcomes obtained from the private test set are kept undisclosed for every submission and are solely disclosed once the competition is concluded. Significantly, a deliberate decision was made to include 88.4% of the out-of-dictionary (OOD) graphemes in the private test set. This strategic choice was intended to discourage the development of models that rely entirely on public standings from overfitting. The aforementioned strategy functioned as a source of motivation for the participants to devise techniques that are capable of categorizing out-of-distribution graphemes by autonomously identifying the desired variables.

4 Method

Our model classifies against 14784 ($168 \times 11 \times 8$) class which are all the possible combinations. Therefore, we needed to know the combinations of labels made up of grapheme. Prediction of the relationship between the combination of labels and grapheme are done from the label of the given train data. The generation of synthetic data and the conversion of the prediction results into three components are based on this correspondence.

Data splitting process was conducted randomly. As a result, an unintended grapheme root class was generated during the evaluation stage, making it impractical to conduct a thorough examination. Data splitting method presented a significant difficulty due to the considerable computational resources and time investment it demanded. Consequently, the local cross-validation (CV) procedures were implemented utilizing the data in its present condition, notwithstanding the aforementioned constraints. Split counts for train and validation for seen and unseen classes are presented in Figure 3.

[1]https://www.kaggle.com/competitions/bengaliai-cv19/

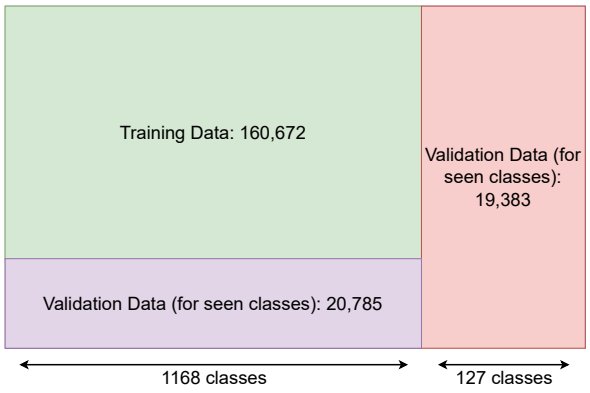


Figure 3: Data split for train data: 160,672. Validation data for seen classes: 20,785 and unseen classes: 19,383.

4.1 Out of Distribution Detection Model

The purpose of the Out of Distribution Detection Model is to categorize input images into either seen or unseen groups. The aforementioned model generates individual confidence scores for each of the 1295 classes in order to make its predictions. In situations where there is a lack of confidence, the image is classified as an unseen class. Conversely, the presence of at least one confidence score signifies that it falls within a seen class. It is worth mentioning that this particular model functions without the need for resizing or cropping input images.

4.2 Seen Class Model

The Seen Class Model has been specifically developed to classify a total of 1295 classes that are included within the training data. The utilized model in this study does not involve any resizing or cropping of input images. Instead, it leverages the AutoAugment Policy specifically designed for the preprocessing of the Street View House Numbers (SVHN) dataset.

4.3 Unseen Class Model

The Unseen Class Model comprises a learning method that consists of two stages. During the initial phase, a classifier is trained to identify images that are produced from TrueType Font (ttf) files. In the subsequent phase, the training process is centered on a generator that is responsible for transforming handwritten characters into synthetic data-like images. In order to facilitate the various stages of learning, the initial step entails the selection of a TrueType font (TTF) and the subsequent generation of a synthetic dataset. The synthetic images are generated to have dimensions that correspond to the training data, particularly 236×137 pixels. The dataset consists of 59,136 samples, each including images created in four distinct sizes: 84×84, 96×96, 108×108, and 120×120 pixels.

4.3.1 Font Classifier Pre-training

Regarding the Font Classifier During the pre-training phase, the images are subjected to several preprocessing operations, such as random affine transformations, random rotation, random cropping, and cutoff, in addition to being cropped and shrunk to dimensions of 224×224 pixels. The work at hand utilizes the EfficientNet-b0 architecture from CNN, while the AdamW optimizer is implemented with default parameter values. The learning rate scheduler utilized in this study is LinearDecay. The output layer is comprised of a Layer Normalization followed by a fully connected layer with dimensions ranging from 2560 to 14784. This is then followed by the application of a Softmax Cross-Entropy activation function. The training process consists of 60 epochs, each utilizing a batch size of 32.

4.3.2 CycleGAN Training

The training method of CycleGAN (Zhu et al., 2017) from scratch[2] comprises the application of a model architecture known as CycleGAN for the purpose of image translation jobs. The input images are subjected to cropping and resizing, resulting in dimensions of 224×224 pixels. These images then undergo preprocessing, which involves random affine transformations, random rotation, and random cropping. However, the dimensions of the random cropping in this phase are reduced compared to the pre-training phase. It is important to note that the cutoff operation is excluded from this preprocessing step. In addition, a pre-trained Font Classifier was incorporated into the model. The parameters of the Font Classifier were kept fixed, and it was operated in evaluation mode.

5 Experiments

Out of distribution detection model utilizes the AutoAugment Policy for preparing the SVHN dataset. The model employed in the study is based on the EfficientNet-b7 (Tan and Le, 2019) architecture, which has been pretrained on the ImageNet dataset. The optimization process leverages the AdamW

[2]https://www.kaggle.com/code/linshokaku/cyclegan-training

optimizer with default parameters. The management of the learning rate scheduling is handled by the WarmUpAndLinearDecay module. The output layer is composed of LayerNorm-FC with dimensions 2560 to 1295, and it utilizes the BCEWithLogitsLoss function. The model undergoes training for a total of 200 epochs, with a batch size of 32. The dataset is divided in a 1:0 ratio, indicating the adoption of a single-fold methodology.

Seen class model employed in this study is based on the EfficientNet-b7 architecture, which has been pretrained on the ImageNet dataset. The optimization process utilizes the AdamW optimizer with the default configuration. The management of the learning rate scheduling is handled by the WarmUpAndLinearDecay module. The output layer is comprised of a Layer Normalization followed by a fully connected layer with dimensions ranging from 2560 to 14784. This is then followed by the application of a Softmax Cross-Entropy activation function. The model is trained for a total of 200 epochs using a batch size of 32. The dataset is divided randomly into a 9:1 ratio, which follows a single-fold methodology for both training and evaluation purposes.

The optimization procedure in CycleGAN training utilizes the Adam optimizer, using a learning rate of 0.0002 and beta values of (0.5, 0.999). The implementation of learning rate scheduling involves the utilization of the LinearDecay method. The training process consists of 40 epochs, where each epoch involves a batch size of 32. The training is performed on a machine configuration consisting of 4 Tesla V100 GPUs, and the entire training process takes approximately 2.5 days to complete. The key hyperparameters of the model consist of *lambda_consistency*, which is set to 10, and *lambda_cls*, which ranges from 1.0 to 5.0. These hyperparameters play a crucial role in determining the performance of the model. The training process plays a crucial role in attaining uniformity and efficacy in tasks related to image translation. The proposed CycleGAN training module is presented in Figure 1.

The leaderboard scores and submissions are assessed using a hierarchical macro-averaged recall (HMAR).

$$HMAR = [(2 * recall_{grapheme\_root}) + recall_{vowel\_diacritic} + recall_{consonant\_diacritic}]/4 \quad (1)$$

Model Architecture	HMAR
SE-ResNeXt50 + Head	0.9584
InceptionResNetv2, SE-ResNeXt101	0.9620
pc-softmax	0.9645
SE-ResNeXt 50 & 101	0.9689
efn-b7, CycleGAN + efn-b0	**0.9762**

Table 1: Outcomes of top 5 submissions in private LB of the competition. Our approach with EfficientNet-B7 (efn-b7), CycleGAN + EfficientNet-B0 (efn-b0) [detailed visualization in Figure 2] scored the highest HMAR. Approaches of LB position 2nd to 5th are mentioned in the Appendix.

For each component (grapheme root, vowel diacritic, or consonant diacritic), a standard macro-averaged recall (MAR) is first calculated. The grapheme root receives double the weight in the final result, which is calculated as the weighted average of those three scores.

6 Results and Discussion

In the domain of handwritten character recognition, ZSL has been a prominent research focus, particularly in the context of Chinese and Japanese character recognition. Many studies have explored the sub-categorization and identification of characters through the manipulation of constituent components.

For Chinese characters (Zhang et al., 2018), which pose a considerable challenge due to their complex grapheme structure, approaches involving the classification of approximately 500 components using recurrent neural networks (RNN) series have been adopted. In contrast, for Bangla characters, an attempt was made to categorize them into three component-based groups, simplifying the multi-class classification process compared to RNN-based methods. However, it became evident that this approach led to significant overfitting in zero-shot recognition, as evidenced by both private validation experiments and competition outcomes. The issue of overfitting in multi-class classification arises from the model's reliance on the entire character for predicting each class, necessitating intricate engineering to dissect the relevant features effectively.

Motivated by the need to address these challenges, an unconventional approach was pursued, where each character was treated as an individual class, even when data for certain classes were scarce. A fundamental assumption underlying this

approach was that if there is a software running on a computer that can recognize a certain character, then that computer is capable of handling the character code and can output it as an image. This assumption was deemed valid within the context of the competition. Several attempts[3] were made to construct models that output characters, and the most successful method among these is presented in this paper. The subsequent sections outline the proposed method and discuss the results obtained.

The findings suggest that the model trained on font images acquired statistically informed, fine-grained character components to efficiently discriminate among font images. Furthermore, the H to F image conversion model appeared to perform the desired transformation, emphasizing recognizable components. This transformation operated on local features of handwriting, with the range of local features being statistically inferred and generalized from the font image-trained model through back-propagation. This generalization facilitated the recognition of zero-shot classes, rendering it feasible. It is worth noting that this technique's strengths extend beyond zero-shot class generalization, encompassing its universal applicability to languages and its straightforward implementation.

Using a dataset composed of 1168 seen classes and 127 unseen classes, the out of distribution detection model results in 0.9967 local CV area under the receiver operating characteristic (AUROC). The CV score achieved for the seen class model is 0.9985, while the Leaderboard score stands at 0.9874. To determine the threshold for this model, a local CV was created weighted to replicate the predicted ratio of seen/unseen classes in the leaderboard. The model's threshold was adopted when this local CV was maximized.

At the initial stage, the loss calculation of the discriminator incorporated a supervised loss obtained from a font classifier that had been pretrained. The configuration that yielded the maximum performance, as indicated by a *lambda_cls* value of 4.0, produced the subsequent CV scores: a local CV score of 0.8804 for previously unobserved classes, and a CV score of 0.9377 for previously observed classes. It is worth noting that the calculation of the MAR involved the exclusion of non-existent classes, hence preventing the assignment of recall values of either 0.0 or 1.0 to these classes.

[3]https://www.kaggle.com/competitions/bengaliai-cv19/discussion/135984

Following that, the hyperparameter modification was conducted, and two further models were trained using different font data, without assessing their local cross-validation performance. These models exhibited a noteworthy achievement on the leaderboard (LB) when compared to the original model, indicating the potential for improved ability to generalize to unfamiliar classes in private data.

In contrast to early hypotheses, the development of synthetic data that closely resembles real samples did not yield the anticipated enhancement in consistency and discriminator losses. Furthermore, this unforeseen inclination did not yield a higher level of generalization towards classes that were not before encountered. During the process of fine-tuning hyperparameters, an observation was made that the model's overall performance exhibited improvement in terms of generalization. However, this improvement was accompanied by a deterioration in the visual quality of the generated images.

The aforementioned observations suggest the potential existence of complex interconnections among hyperparameters, model architecture, and the ability to generalize to novel classes. Consequently, it is imperative to do additional research in order to explore this matter in greater depth.

HMAR scores on the final LB for the top 5 performing architectures are mentioned in Table 1.

7 Conclusion

This work introduces an innovative architecture for Zero-Shot Learning (ZSL) in the context of Optical Character Recognition (OCR), specifically for the complex Bangla script. Our objective was not solely to assign characters to seen classes, but also to enhance our model's ability to identify and classify classes that were unseen. By utilizing the CycleGAN for image synthesis and implementing accurate label mapping techniques, a robust correlation between labels and graphemes has been developed.

By actively engaging in the *Bengali.AI Handwritten Grapheme Classification* competition, we achieved the highest rank on the scoreboard, effectively demonstrating the exceptional capabilities of our novel model. The performance of our system in detecting out-of-distribution instances was exceptional. It achieved an Area Under the Receiver Operating Characteristic (AUROC) score of 0.9967 for a dataset incorporating 1168 seen classes and 127 unseen classes. Additionally, the system demon-

strated a Cross-Validation (CV) score of 0.9985 for the classes it had encountered during training, and a leaderboard score of 0.9874. And finally, it achieved a hierarchical macro-averaged recall (HMAR) score of 0.9762 and topped the leaderboard amongst other contestants. Through our investigation, we have uncovered the intricate relationships among hyperparameters, model architecture, and the ability to generalize to unfamiliar classes. This study represents a noteworthy achievement in the field of character recognition. It is posited that our proposed methodology possesses the capacity to fundamentally transform the field of Bangla OCR development, hence expediting the process of digitizing educational materials in the Indian subcontinent. Furthermore, it is suggested that the prospective uses of this strategy may expand beyond the realm of character recognition. The comprehensive examination of these complex interconnections is important in order to fully realize the potential of our pioneering approach.

Limitations

The process of splitting data for cross-validation of unseen classes is conducted randomly. During the assessment process, it was found that a grapheme root class that did not exist had been generated, which resulted in the inability to conduct a proper evaluation. The rationale for the expansion of the Unseen class using this approach cannot be substantiated. There was an anticipation that the production of images closely resembling the synthetic data would have an effect on the Consistency Loss and the Discriminator, thus leading to an enhancement in generalization for the unseen class. Nevertheless, when adjusting the hyperparameters, it was seen that the overall performance of generalization was enhanced, but at the cost of the images exhibiting an artificial aspect.

Ethics Statement

This study adheres to ethical principles, such as obtaining informed consent, ensuring confidentiality, maintaining integrity, preventing harm, complying with applicable regulations, acknowledging sources, and disclosing conflicts of interest.

Acknowledgements

This project is supported by HKSAR RGC under Grant No. PolyU 15221420. Finally, thanks to the anonymous reviewers for their valuable input.

References

Samiul Alam, Tahsin Reasat, Asif Shahriyar Sushmit, Sadi Mohammad Siddique, Fuad Rahman, Mahady Hasan, and Ahmed Imtiaz Humayun. 2021. A large multi-target dataset of common bengali handwritten graphemes. In *International Conference on Document Analysis and Recognition*, pages 383–398. Springer.

Subhadip Basu, Nibaran Das, Ram Sarkar, Mahantapas Kundu, Mita Nasipuri, and Dipak Kumar Basu. 2009. A hierarchical approach to recognition of handwritten bangla characters. *Pattern Recognition*, 42(7):1467–1484.

Ahmed Asif Chowdhury, Ejaj Ahmed, Shameem Ahmed, Shohrab Hossain, and Chowdhury Mofizur Rahman. 2002. Optical character recognition of bangla characters using neural network: A better approach. In *2nd ICEE*.

Md Imdadul Haque Emon, Khondoker Nazia Iqbal, Md Humaion Kabir Mehedi, Mohammed Julfikar Ali Mahbub, and Annajiat Alim Rasel. 2022. A review of optical character recognition (ocr) techniques on bengali scripts. In *International Conference for Emerging Technologies in Computing*, pages 85–94. Springer.

Ahmad Fuad Rezaur Rahman. 1994. Recognition of bangla hand written characters using pattern recognition techniques.

Md Saif Hassan Onim, Hussain Nyeem, Koushik Roy, Mahmudul Hasan, Abtahi Ishmam, Md Akiful Hoque Akif, and Tareque Bashar Ovi. 2022. Blpnet: A new dnn model and bengali ocr engine for automatic licence plate recognition. *Array*, 15:100244.

Debabrata Paul and Bidyut Baran Chaudhuri. 2019. A blstm network for printed bengali ocr system with high accuracy. *arXiv preprint arXiv:1908.08674*.

Ahmad Fuad Rezaur Rahman, R Rahman, and Michael C Fairhurst. 2002. Recognition of handwritten bengali characters: a novel multistage approach. *Pattern Recognition*, 35(5):997–1006.

Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Jianshu Zhang, Yixing Zhu, Jun Du, and Lirong Dai. 2018. Trajectory-based radical analysis network for online handwritten chinese character recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3681–3686. IEEE.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Appendix

SE-ResNeXt50 + Head (5th place solution): This solution involved using SE-ResNeXt50 model with a customized head for improving scores on the LB. Notable improvements were made in consonant diacritic prediction. Preprocessing included image normalization. Architecture had a SE-ResNeXt50 model as a backbone with multiple heads for different tasks. Training used all available data with a cosine annealing schedule and various augmentation techniques. Optimized used: Adam. Loss functions were specified for different tasks, and the final loss combined them. Postprocessing involved a uniform threshold for consonant diacritic prediction. Cosine similarity helped match predictions with training samples. Submissions were selected based on binarization and metric learning criteria, with a focus on LB performance. Overfitting to the public LB was considered, resulting in the selection of the best and a slightly modified submission, both performing well on the private LB.

InceptionResNetv2, SE-ResNeXt101 (4th place solution): The primary objective is to divide a dataset of 1,295 graphemes into In-Dictionary (ID) and Out-of-Dictionary (OOD) categories. The strategy entails training Arcface models to calculate the centroid of each of these 1,295 graphemes' features. The test images are then classified as either ID or OOD based on the shortest distance of a feature to the grapheme centers. The threshold of 0.15 (cosine distance) was estimated locally and contributed to the fourth place submission for the competition.

pc-softmax (3rd place solution): The technique aims to categorize of both seen and unseen graphemes. The methodology involves the utilization of preprocessed images that undergo flipping operations to generate triple identities, which are subsequently standardized to dimensions of 137×236 pixels. The model's architecture consists of two encoders, namely "phalanx" and "earthian," which are subsequently followed by global average pooling, batch normalization, and fully connected layers. The model is trained using the Arcface loss function. To improve performance, a secondary encoder is incorporated, accompanied by augmentations such as cut mix and geometric alterations to promote resilience. The training process involves distinguishing between graphemes that are familiar and those that are unknown. This is achieved by first pretraining on a specific dataset, followed by fine-tuning for the familiar graphemes. Subsequently, additional training is conducted on the original dataset to address the unfamiliar graphemes. The conventional softmax function is substituted with pc-softmax, which utilizes negative log probability as the loss function. The utilization of Stochastic Gradient Descent (SGD) with CosineAnnealing is employed to optimize the model, while Stochastic Weighted Average is utilized to facilitate the training process. The process of inference involves the utilization of Arcface, a technique that computes cosine similarities. The determination of a threshold is based on the minimum similarity between the embeddings of the training and validation datasets. This technique demonstrates a high level of efficacy in addressing grapheme categorization tasks, irrespective of the level of familiarity with the task at hand.

SE-ResNeXt 50 & 101 (2nd place solution): The implemented technique utilized a sequence of strategic modifications to improve the classification of graphemes. The initial approach involved a transition from predicting grapheme components to predicting individual graphemes, hence enabling the implementation of more sophisticated enhancements such as FMix. Post-processing techniques were employed to enhance the accuracy of predictions for both familiar and unfamiliar graphemes. In order to address the issue of overfitting, distinct models were developed for the R and C components, incorporating the utilization of synthetic grapheme creation. The utilization of model blending was of utmost importance, necessitating the implementation of separate methodologies for each individual component. The study employed SE-ResNeXt50 and 101 models, employed different image sizes, and utilized optimization strategies to attain better outcomes.

Low-Resource Text Style Transfer for Bangla: Data & Models

Sourabrata Mukherjee[1], Akanksha Bansal[2], Pritha Majumdar[2]
Atul Kr. Ojha[3,2], Ondřej Dušek[1]
[1]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
[2]Panlingua Language Processing LLP, India
[3]Insight SFI Centre for Data Analytics, DSI, University of Galway, Ireland
{mukherjee,odusek}@ufal.mff.cuni.cz
{akanksha.bansal,pritha.majumdar}@panlingua.co.in
atulkumar.ojha@insight-centre.org

Abstract

Text style transfer (TST) involves modifying the linguistic style of a given text while retaining its core content. This paper addresses the challenging task of text style transfer in the Bangla language, which is low-resourced in this area. We present a novel Bangla dataset that facilitates text sentiment transfer, a subtask of TST, enabling the transformation of positive sentiment sentences to negative and vice versa. To establish a high-quality base for further research, we refined and corrected an existing English dataset of 1,000 sentences for sentiment transfer based on Yelp reviews, and we introduce a new human-translated Bangla dataset that parallels its English counterpart. Furthermore, we offer multiple benchmark models that serve as a validation of the dataset and baseline for further research.

1 Introduction

Text style transfer (TST) aims to modify the style of a given text while preserving its underlying content (Shen et al., 2017; Prabhumoye et al., 2018; Li et al., 2018, see Figure 1). Prior research in text style transfer has primarily focused on the English language, overlooking languages with limited resources, such as Bangla. This work aims to close this gap specifically for Bangla and explores the text sentiment transfer task, which is a prominent subtask[1] of TST (Jin et al., 2022; Mukherjee et al., 2022; Luo et al., 2019a).

Bangla, also referred to as Bengali, is mostly spoken in the Indian regions of West Bengal, Assam, and Tripura and is the mother tongue of about 97.2 million speakers as per the 2011 Census Report of India.[2] It is one of the 22 scheduled (official) (Jha, 2010) Indian languages and the national language of Bangladesh. Syntactically, Bangla

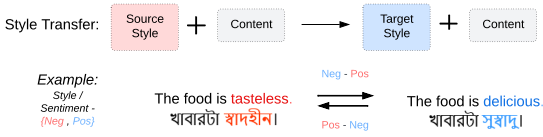


Figure 1: An example of sentiment transfer as a TST task in English and Bangla. Adapted from our previous paper (Mukherjee and Dusek, 2023).

is agglutinative by nature. A single verb root in Bangla can have 150 + inflected forms (McCrae et al., 2021). There are multiple dialects of Bangla that vary mainly in terms of verb inflections and intonation (McCrae et al., 2021). For this work, we followed Bangla as spoken in West Bengal.

The unique challenges posed by the low-resource nature of Bangla require specifically tailored innovative approaches. To achieve TST in Bangla, we build upon an existing English dataset of 1,000 sentences for this task adapted from Yelp reviews by Li et al. (2018). However, upon careful examination, we found that the quality of the original English dataset did not meet the standards we aimed to establish. To address this problem, we manually checked and modified the English dataset to improve its quality. Subsequently, we adapted the curated English dataset to the Bangla language, ensuring alignment in both content and structure. Importantly, we introduce a novel Bangla dataset, crafted by human annotators, serving as a parallel counterpart to the refined English dataset.

Furthermore, to facilitate the evaluation, we provide benchmark models capable of assessing the efficacy of text style transfer on our datasets. This paper marks a significant contribution to the field, as it not only pioneers text style transfer in the Bangla language but also provides a foundation for future research endeavors in multilingual text style transfer. Our work not only broadens the scope of text style transfer to include a low-resource language

[1]Moving forward, we will use the terms "style transfer" and "sentiment transfer" interchangeably in this paper.

[2]https://censusindia.gov.in/nada/index.php/catalog/42458

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 34–47

but also underscores the importance of dataset quality. Our data and experimental code are released on GitHub.[3]

Our contributions are summarized as follows:

(i) We have enhanced the quality of the existing English parallel dataset for text sentiment transfer, improving its utility for research and applications.

(ii) We introduce a novel Bangla parallel dataset aligned with its English counterpart, effectively expanding the resources available for text style transfer in Bangla.

(iii) We present benchmark models to evaluate the performance of these datasets.

(iv) We also explored the challenging scenarios of having no style-parallel data or not using any human-annotated Bangla data for training (opting instead for English-to-Bangla machine translation). These experiments demonstrate the potential for meaningful results even with limited or no language-specific resources.

2 Related Work

Existing works in TST are mostly aimed at the English language and can be broadly classified into below categories:

TST with Parallel Data TST can be modeled as a sequence-to-sequence task and trained on pairs of texts with similar content but different styles. Here, Jhamtani et al. (2017) used a sequence-to-sequence model with a pointer network to translate modern English into Shakespearean English. Mukherjee and Dusek (2023) leveraged minimal parallel data and incorporated various low-resource methods to explore the TST task. However, this approach to TST is inherently challenging due to the scarcity of parallel data (Hu et al., 2022; Mukherjee and Dusek, 2023).

Non-Parallel Approaches to TST Two main strategies were employed to avoid reliance on parallel data: (i) straightforward text replacement, where style-specific phrases are explicitly identified and replaced (Li et al., 2018; Mukherjee et al., 2022), (ii) implicit style-content disentanglement via latent representations through techniques such as back-translation and autoencoding (Shen et al., 2017; Zhao et al., 2018; Fu et al., 2018; Prabhumoye et al., 2018; Hu et al., 2017). Adversarial learning was shown to improve the results of both approaches (Lample et al., 2019; Dai et al., 2019; Li et al., 2019; Luo et al., 2019b). Despite a lot of progress, non-parallel approaches tend to produce mixed results and often require large amounts of non-parallel data, which is not readily available for many styles, limiting their practical applicability in low-resource settings (Li et al., 2022).

Multilingual style transfer remains a relatively uncharted territory in prior research. In a comprehensive survey conducted by Briakou et al. (2021), only one work of TST was identified in languages such as Chinese, Russian, Latvian, Estonian, and French. Additionally, they introduced an evaluation dataset for formality transfer, encompassing French, Brazilian Portuguese, and Italian. Another study focused on formality transfer across various Indic languages (Krishna et al., 2022). Existing work has primarily concentrated on resource-rich languages, leaving languages like Bangla understudied in the domain of TST. The only previous work on Bangla known to us is the experiment of Palash et al. (2019), who used a small amount of non-parallel data to train an autoencoder, with largely negative results.

3 Dataset Creation

We utilized the Yelp dataset (Li et al., 2018), which is publicly available and has been used by prior TST experiments. It consists of user-generated content in the form of reviews for hospitality establishments. For each review sentence that is originally positive or negative, a parallel sentence has been created where the sentiment has been flipped but sentiment-independent content retained as much as possible. The dataset is in English. 500 sentences have been transferred from negative to positive and another 500 from positive to negative.

Implicit and Explicit Sentiment in Text Data The methodology behind creating sentences where sentiment transfer has taken place is a crucial process and a creative one. It primarily involves the identification of the sentiment-bearing attribute; for example, in the sentence *The food is tasteless*, "tasteless" is the sentiment-bearing attribute. The sentiment-bearing attribute can be transformed in

[3]Code: https://github.com/souro/multilingual_tst, data: https://github.com/panlingua/multilingual-tst-datasets.

multiple ways, here, e.g., by using an antonym of "tasteless" or adding a negation marker to tasty. The output, thus, could be either *The food is tasty* or *The food is not tasteless*. In both output sentences, the maximum lexical context has been preserved. It is the naturalness of the sentence or utterance that decides on preference between the two options. Sentiment-bearing attributes expressed as singular words or phrases are relatively easy to handle. The difficulty arises when the sentiment is carried or expressed implicitly. For this, the principle of sounding natural must be given prominence over lexical context preservation. A few such examples are reported in Table 3 in Appendix A.

English Data Correction The original English Yelp dataset included several discrepancies, some of which are reported in Table 4 in Appendix A: spelling mistakes, the incorrect sentiment of input sentences (flipped or neutral), compromise on naturalness, loss of context that could be preserved, or not changing the sentiment correctly in the target data, especially in cases where sentiment was expressed implicitly. For these reasons, we edited 451 sentences out of 1,000 in the original English Yelp dataset to meet the requirements of our experiment.

Creation of Bangla Data This dataset has been translated from English to Bangla to serve the aims of our experiment. Apart from usual translation challenges, specific problems arise for this particular dataset where sentiment transfer must be maintained (see Tables 5 and 6 in Appendix A for specific examples).

Some expressions that appear natural in English may come across as unnatural in Bangla. Hence, the complete lexical context may not be preserved during translation. Ambiguity in sentences poses difficulties in preserving multiple interpretations, and sentences with implied meanings are particularly challenging to translate. To address this, we often use similar phrases that maintain naturalness but may compromise lexical context. Slang words further complicate the translation process, as do instances where the original meaning is unclear, resulting in equally unclear translations. Consistency is crucial for small datasets, as variations in translation can affect results; for instance, bland can be translated as either 'flavourless' or 'tasteless'. Maintaining consistency was challenging in this respect. Lastly, a lack of cultural knowledge may lead to misinterpretations in translation, exemplified by cases like 'bs' meaning 'bullshit'.

4 Models

Our models are categorized into three approaches: parallel, non-parallel (not using parallel training data), and cross-lingual, i.e., without using our Bangla dataset for training. For an overview of the methodologies, see Figure 2.

4.1 Parallel Style Transfer

Here, we simply fine-tune a pre-trained multilingual BART model (mBART) (Liu et al., 2020) using the parallel English and Bangla datasets constructed in Section 3. This approach is directly based on our previous work (Mukherjee and Dusek, 2023).

4.2 Non-parallel Style Transfer

In this experiment, we only use one part of the data at a time (positive/negative), never using the human-labelled targets for a given example. We harness the power of reconstruction of the input using an **auto-encoder** (AE) (Shen et al., 2017; Li et al., 2021) and **back-translation** (BT) (Prabhumoye et al., 2018; Mukherjee et al., 2022). In the BT process, for English sentences, we perform a cycle of translation, using English-to-Bangla-to-English, while for Bangla sentences, we apply Bangla-to-English-to-Bangla translation. For both AE and BT approaches, we train two separate models for each sentiment. At inference time, input is simply fed to the model trained on the intended target sentiment.

Masked Style Filling (MSF) We further extended the above AE and BT approaches by masking out the style-specific lexicon in the input sentence. Instead of relying on a fixed, contextually unaware style lexicon lookup, we take a dynamic and sentence-level perspective to identify important style-specific words in a sentence. This approach recognizes that words can have different stylistic roles based on the context in which they appear. To achieve this, we employ integrated gradients, a well-known model interpretability technique (Sundararajan et al., 2017; Janizek et al., 2021) on a fine-tuned mBERT (Pires et al., 2019) style classifier. This technique provides word attributions, essentially scores that show how much a word contributes to the style classifier model's prediction.

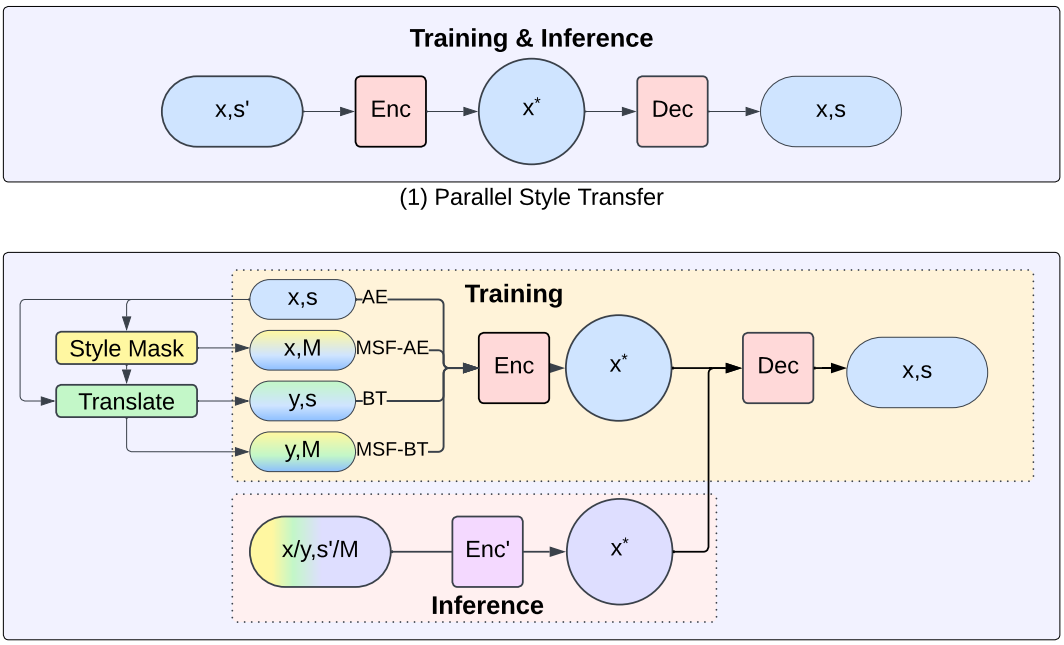


Figure 2: Overview of the Methodologies. (1) Parallel Sentences: This method employs aligned pairs of sentences with opposite styles, such as positive-to-negative and negative-to-positive. It employs a basic sequence-to-sequence (seq2seq) text generation approach, using an encoder (*Enc*) to process the input (x) and a decoder (*Dec*) to generate the opposite-style sentence (s'). For instance, to convert a positive sentence to a negative one, *Enc* encodes the positive text, and *Dec* decodes it into a negative sentiment. (2) Non-Parallel Data: In cases where aligned sentences are unavailable, this approach leverages non-parallel datasets containing positive and negative text. Two strategies are used: First, reconstruction, which uses auto-encoding (*AE*) or back-translation (*BT*). In *BT*, the input (x) is machine-translated to the opposite language (y) beforehand. Separate models are trained to reconstruct positive and negative sentences, but during inference, cross-models are used. For example, when transferring from positive to negative sentiment, the input is a positive sentence, and the model used for reconstruction is the one trained on negative sentences. The opposite applies to negative-to-positive transfers. In addition to this, Masked Style Filling (*MSF*) may be applied as preprocessing. *MSF* masks style-specific lexicon within the input, aided by a trained classifier and axiomatic attribution scores that identify style lexicon. The resulting style-masked sentence, denoted as (x, M) or (y, M), then undergoes the same reconstruction process (*AE* or *BT*).

With these word attribution scores in hand, we selectively mask out words that are considered style lexicon. We set a threshold to determine how much of the overall style should be removed from the sentence. The objective here is to create sentences that are "style-independent", devoid of specific stylistic markers. We then use these modified sentences as input to our AE and BT reconstruction models to reconstruct the original sentences. We again train two separate models for each sentiment and feed inputs to to the model trained on the intended target sentiment at inference time.

4.3 Cross-Lingual Style Transfer

We explore two basic cross-lingual alternatives that circumvent the use of the manually created Bangla dataset. Firstly, we employ English sentences from the parallel dataset, translate them into Bangla, and use this translated text for training. Secondly, we take the English output generated by the model trained on a parallel English dataset and translate it into Bangla. These cross-lingual approaches offer intriguing insights into multilingual text style transfer where the TST dataset is not available in the target language.

5 Experimetal Settings

Each dataset consists of 500 positive-to-negative and 500 negative-to-positive sentences (see Section 3). To maintain a consistent approach across all our experiments, we have divided these datasets into 400 examples for training, 100 for development, and 500 for testing purposes.

We used the mBART-large-50 model (Tang

et al., 2020) from the HuggingFace library (Wolf et al., 2020) for both English and Bangla. To optimize model performance, hyperparameter tuning was performed, resulting in the selection of a learning rate of 1e-5 and a batch size of 3. Dropout was applied with a rate of 0.1 across the network. Additionally, L2 regularization with a strength of 0.01 was introduced. Training ran over 5 epochs.

For machine translation in the BT reconstruction experiment, we use the Facebook NLLB-200-3.3B models (Costa-jussà et al., 2022) from HuggingFace.

For our MSF experiments and for evaluating sentiment transfer accuracy, we fine-tuned a classifier based on the BERT-base multilingual cased model (Devlin et al., 2018; Pires et al., 2019), using the same training set as our primary task. This fine-tuned classifier achieves accuracy rates of 87.0% for English and 83.0% for Bangla. In the MSF process, we employ a threshold of 0.5 to selectively filter style lexicon.

6 Evaluation and Results

6.1 Evaluation

The evaluation process encompasses three key aspects: sentiment transfer accuracy, content preservation, and fluency. To assess sentiment transfer accuracy, we used our finetuned mBERT classifier (see Section 5). Consistent with prior research (Jin et al., 2022; Hu et al., 2022; Mukherjee et al., 2023), content preservation is assessed through BLEU score (Papineni et al., 2002) and embedding similarity (Rahutomo et al., 2012) against the input sentences. The embedding similarity is determined using language-agnostic BERT sentence embedding (LaBSE) (Feng et al., 2022) in conjunction with cosine similarity. Evaluating fluency, especially for Bangla, poses a challenge as there are no good assessment tools available for indic languages (Krishna et al., 2022). Previous research has cautioned against using perplexity (PPL) for fluency, as it tends to favor unnatural sentences with common words (Pang, 2019; Mir et al., 2019). Despite these problems, we still include a basic fluency evaluation using perplexity (PPL) measured with a multilingual GPT model (Shliazhko et al., 2022).

6.2 Results

Automatic metric results are shown in the Table 1. Our scores are roughly in the same ballpark as our previous experiments on the Yelp data (Mukherjee et al., 2023) and with somewhat lower style accuracy but higher content preservation scores than most other previous works (cf. Li et al., 2020). However, a direct comparison on the English data is not possible due to our corrections of the dataset (see Section 3). Based on cursory manual checks of the output texts, the scores reflect the individual models' performance well.

Style Accuracy: The benchmark model utilizing the parallel dataset demonstrates strong style accuracy. However, in the case of Bangla, the accuracy drops in comparison to English, indicating potential challenges in Bangla-style transfer. Non-parallel data models, such as AE and BT, exhibit significantly lower style accuracy in both languages.

Content Preservation: While the parallel model and the MSF-AE model perform relatively well in both languages, other non-parallel models struggle to preserve content effectively. The MSF approach in general enhances content preservation, narrowing the gap slightly between parallel and non-parallel data models.

Comparison of AE and BT: When comparing the performance of AE and BT models, AE tends to outperform BT in content preservation, but BT outperforms AE in style transfer accuracy.

Impact of MSF: The introduction of the MSF approach in general improves the results of both AE and BT models, increasing style accuracy and fluency, but at the cost of content preservation.

Parallel vs. non-parallel Data: As expected, parallel data models consistently outperform their non-parallel counterparts across various metrics. However, the incorporation of the MSF approach mitigates some of the challenges posed by non-parallel data, highlighting its effectiveness in bridging the performance gap.

Cross-lingual Experiments: By not using the actual Bangla dataset entirely, we explored two alternative approaches: (i) translating parallel English training sentences to Bangla and (ii) translating the English style transfer output to Bangla. Interestingly, both methods yield competitive results in Bangla, showcasing the potential of the style-parallel English dataset and simple translation for the text style transfer task if the actual TST

	English				Bangla			
Models	**ACC**	**BLEU**	**CS**	**PPL**	**ACC**	**BLEU**	**CS**	**PPL**
Parallel Style Transfer								
Parallel	77.0	46.5	81.0	97.5	66.0	34.5	81.0	7.7
Non-parallel Style Transfer								
AE	13.0	42.0	78.0	102.2	17.0	31.0	78.0	7.8
BT	28.0	10.0	64.5	139.4	33.5	3.0	63.5	7.3
MSF-AE	59.5	37.5	75.5	136.0	72.0	26.5	72.5	7.9
MSF-BT	59.5	9.5	62.0	90.2	55.5	1.0	43.0	26.7
Cross-Lingual Style Transfer								
Train-En-TR		-			61.0	28.0	79.0	7.7
En-OP-TR		-			64.5	6.0	74.5	6.8

Table 1: Automatic evaluation results. We measure the sentiment classifier accuracy (ACC), BLEU score, Content Similarity (CS), and Fluency (PPL), see Section 6.1. We have several models (see Section 4): Parallel that uses parallel data, AE and BT for non-parallel data using the reconstruction approach, and the extended models MSF-AE and MSF-BT employing Masked Style Filling. Train-En-Tr involves training without the human-annotated Bangla dataset by using English-to-Bangla machine-translated training data. En-OP-TR refers to the Bangla translation of English output generated by mBART-base using parallel English data.

dataset is not available in Bangla for training purposes. The performance of these methods is on par with or surpasses that of the non-parallel Bangla dataset-based models, underscoring the viability of using machine translation in the pipeline.

Comparison of English and Bangla Results: While the scores in both languages are not directly comparable, overall lower values for Bangla show that this problem is likely more challenging here, not least due to Bangla's more complex morphology or lower amount of pretraining in the underlying mBART language model. Both languages however correlate relatively well in terms of the relative performance of the individual models (parallel model are the best, MSF improves scores, BT seems worse than AE on content preservation).

In conclusion, our experiments emphasize the significance of parallel data in text style transfer and highlight the benefits of the MSF approach. The choice of model depends on the specific language, task requirements, and availability. Generated output samples are shown in Table 2.

7 Conclusion

In this study, we delved into the challenging domain of text style transfer primarily for the Bangla language, addressing the scarcity of resources in it. This work contributes essential resources and benchmark models for both, Bangla and English. Future work involves exploring further underrepresented languages in the multi-lingual TST research.

Limitations

Data Bias: Our study relies on publicly available text data, which may inherently contain biases present in the sources from which it was collected. These biases can affect the performance of models trained on such data and may lead to biased outputs in sentiment transfer tasks.

Generalization: While our models demonstrate good performance on our datasets, their ability to generalize to other domains or contexts may be limited.

Subjectivity and Context: Sentiment analysis is inherently subjective, and the sentiment labels assigned to sentences may not universally apply. The context in which a sentence is used can significantly influence its sentiment, and our models may not always capture nuanced contextual variations.

Evaluation Metrics: While we have employed a variety of evaluation metrics, including style transfer accuracy, content preservation, and fluency, no single metric captures all aspects of sentiment transfer. The evaluation process remains an active area of research, and further advancements in metrics may be needed.

Ethics Statement

Data Privacy and Consent: We are committed to respecting data privacy and ensuring that all data

Models	Negative → Positive	Positive → Negative	Analysis
Reference	hate the aternoon-tea at the phoenician. → love the afternoon-tea at the phoenician. ফিনিশিয়ানে দুপুরের চা একদম অপছন্দের । → ফিনিশিয়ানে দুপুরের চা খুব পছন্দের ।	i love their fresh juices as well. → i don't like their fresh juices either. আমার তাদের তাজা ফলের রসও খুব পছন্দ । → আমার তাদের তাজা ফলের রসও একদম পছন্দ না ।	The examples have been chosen to exemplify the use of antonym and NEG-marker to flip the sentiments. In both the examples, the remaining lexical context remains preserved.
Parallel	love the aternoon-tea at the phoenician. ফিনিশিয়ানে দুপুরের চা খুব পছন্দের ।	i hate their fresh juices as well. আমার তাদের তাজা ফলের রসও খুব পছন্দ নয় ।	For both languages, the transfer is extremely smooth with only the sentiment-bearing attributes changed, and the lexical context preserved.
AE	hate the aternoon-tea at the phoenician. ফিনিশিয়ানে দুপুরের চা একদম অপছন্দের ।	i love their fresh juices as well. আমার তাদের তাজা ফলের রসও খুব পছন্দ ।	Basically the model was able to reconstruct the input successfully, thus preserving the content, for both Bangla and English, but it failed in transforming the sentiment fully.
BT	I like the morning coffee. সকালটাটা খুবই পছন্দের কাছ থেকে।।	I hate their cheese. How আমি তাদের সুস্বাদু পানীয় খেতে পছন্দ করি।	The sentiments in English have transformed as desired but the lexical context has been compromised. In Bangla, the sentiment transfer has failed and the context preservation is worse in Positive-Negative as compared to Negative-Positive. Note, that in English, the context transformation remains within their respective categories unlike Bangla.
MSF-AE	the aternoon - tea at the phoenician. দুর্দান্ত সেলফিফিচালত না ।	didn't love their fresh as well. আমার তাদের তাজা ফলের রসও খুব খারাপ ।	While the sentiment-bearing attribute gets dropped altogether in English Negative-Positive making it neutral, but the sentiments have successfully transformed in Positive to Negative. However, the lexical context is slightly compromised in the latter. In Bangla, no sentiment transfer took place in Negative-Positive while sentiments were successfully transferred from Positive to Negative. This model has performed better for Positive to Negative.
MSF-BT	I like the Mexican chicken. সকালের সূর্যসা খুবই অস্বাভাবিকভাবে পছন্দ করা হয়।	I hate them. আমি তাদের fresh খাবার পছন্দ পছন্দম ।	Much like BT above, the sentiments have transformed successfully but so has the lexical context. Therefore, context preservation in this model is erroneous.
Train-En-TR	ফিনিশিয়ানে দুপুরের খাবার খুব পছন্দের ।	আমার তাদের তাজা ফলের রস খুব পছন্দ নয়।	In Negative to Positive the task has been completed perfectly. In Positive to Negative, sentiment transfer has been successfully carried out but content preservation is average.
En-OP-TR	ফিনিকিয়ান এর সকালের চা ভালো লাগে।	আমি তাদের তাজা রস পছন্দ করি না।	The output is in accordance with the model's aim.

Table 2: Here are sample outputs from our models, with sentiment marker words highlighted. The outputs for both the positive-to-negative and negative-to-positive tasks align with the scores presented in Table 1. In both English and Bangla sentences for both tasks, the parallel model performs reasonably well in terms of sentiment transfer and content preservation. On the other hand, the non-parallel models, AE and BT, show below-average performance, but their MSF extensions attempt to enhance sentiment transfer accuracy. MSF-AE is decent in content preservation but struggles with sentiment preservation, while MSF-BT performs decent in sentiment transfer accuracy but falls short in content preservation. Overall, the generated English samples are slightly better than the Bangla ones.

used in our research is anonymized and devoid of personally identifiable information. We have taken measures to protect the privacy and confidentiality of individuals whose data may be included in our datasets.

Bias Mitigation: We acknowledge the potential presence of bias in our data sources and have taken

steps to minimize the impact of such bias during model training and evaluation. We prioritize fairness and strive to mitigate any potential bias in our results.

Transparency and Reproducibility: We are dedicated to providing transparency in our research methods, including dataset collection, preprocessing, and model training. We encourage reproducibility by making our code and datasets publicly available.

Informed Consent: In cases where our research involves human annotators or data contributors, we have sought informed consent and have followed ethical guidelines for data collection and usage.

Social Impact: We recognize the potential social impact of our research and remain vigilant about the responsible use of AI technologies. We aim to contribute positively to the field of sentiment analysis and ensure our work benefits society as a whole.

By acknowledging these limitations and adhering to ethical principles, we aim to conduct responsible and impactful research in multilingual TST. We remain committed to addressing ethical concerns and improving the robustness and fairness of our models as we continue our research endeavors.

Acknowledgements

This research was supported by the European Research Council (Grant agreement No. 101039303 NG-NLG) and by Charles University projects GAUK 392221 and SVV 260575. We acknowledge of the use of resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101). We would also like to acknowledge Panlingua Language Processing LLP for this collaborative research project and for providing the dataset.

Atul Kr. Ojha would like to acknowledge the support of the Science Foundation Ireland (SFI) as part of Grant Number SFI/12/RC/2289_P2 Insight_2, Insight SFI Centre for Data Analytics.

References

Eleftheria Briakou, Di Lu, Ke Zhang, and Joel Tetreault. 2021. Olá, bonjour, salve! XFORMAL: A benchmark for multilingual formality style transfer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3199–3216, Online. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*, pages 5997–6007, Florence, Italy.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 663–670.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal, and Aston Zhang. 2022. Text style transfer: A review and experimental evaluation. *SIGKDD Explor.*, 24(1):14–45.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, Sydney, NSW, Australia.

Joseph D. Janizek, Pascal Sturmfels, and Su-In Lee. 2021. Explaining explanations: Axiomatic feature interactions for deep networks. *J. Mach. Learn. Res.*, 22:104:1–104:54.

Girish Nath Jha. 2010. The TDIL program and the Indian langauge corpora intitiative (ILCI). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19, Copenhagen, Denmark.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Comput. Linguistics*, 48(1):155–205.

Kalpesh Krishna, Deepak Nathani, Xavier Garcia, Bidisha Samanta, and Partha Talukdar. 2022. Few-shot controllable style transfer for low-resource multilingual settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7439–7468. Association for Computational Linguistics.

Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *7th International Conference on Learning Representations, ICLR 2019*, New Orleans, LA, USA, May 6-9, 2019.

Dianqi Li, Yizhe Zhang, Zhe Gan, Yu Cheng, Chris Brockett, Bill Dolan, and Ming-Ting Sun. 2019. Domain adaptive text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3302–3311.

Jicheng Li, Yang Feng, and Jiao Ou. 2021. SE-DAE: style-enhanced denoising auto-encoder for unsupervised text style transfer. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–8. IEEE.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana, USA.

Xiangyang Li, Xiang Long, Yu Xia, and Sujian Li. 2022. Low resource style transfer via domain adaptive meta learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, pages 3014–3026, s, WA, United States.

Xiao Li, Guanyi Chen, Chenghua Lin, and Ruizhe Li. 2020. DGST: a Dual-Generator Network for Text Style Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7131–7136, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019a. Towards fine-grained text sentiment transfer. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*, pages 2013–2022, Florence, Italy.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019b. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pages 5116–5122, Macao.

John P McCrae, Atul Kr Ojha, Bharathi Raja Chakravarthi, Ian Kelly, Patricia Buffini, Grace Tang, Eric Paquin, and Manuel Locria. 2021. Enriching a terminology for under-resourced languages using knowledge graphs. *Electronic lexicography in the 21st century (eLex 2021) Post-editing lexicography*, page 67.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text.

Sourabrata Mukherjee and Ondrej Dusek. 2023. Leveraging low-resource parallel data for text style transfer. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 388–395, Prague, Czechia. Association for Computational Linguistics.

Sourabrata Mukherjee, Vojtech Hudecek, and Ondrej Dusek. 2023. Polite chatbot: A text style transfer application. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: EACL 2023 - Student Research Workshop, Dubrovnik, Croatia, May 2-4, 2023*, pages 87–93.

Sourabrata Mukherjee, Zdenek Kasner, and Ondrej Dusek. 2022. Balancing the style-content trade-off in sentiment transfer using polarity-aware denoising. In *Text, Speech, and Dialogue - 25th International Conference, TSD 2022, Brno, Czech Republic, September 6-9, 2022, Proceedings*, volume 13502 of

Lecture Notes in Computer Science, pages 172–186. Springer.

Mehedi Hasan Palash, Partha Protim Das, and Summit Haque. 2019. Sentimental Style Transfer in Text with Multigenerative Variational Auto-Encoder. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4.

Richard Yuanzhe Pang. 2019. Towards actual (not operational) textual style transfer auto-evaluation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 1: Long Papers*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. 2012. Semantic cosine similarity. In *The 7th international student conference on advanced science and technology ICAST*, volume 4, page 1.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6830–6841.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *CoRR*, abs/2204.07580.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5897–5906, Stockholm, Sweden.

A Appendix

ID	Input Sentence	Output Sentence	Analysis
1	guess she wasn't happy that we were asking the prices.	She was certainly happy to mention the prices.	Simply removing NEG-marker couldn't work as 'she was happy we were asking the prices' sounded unnatural.
2	my toddler found a dead mouse under one of the seats.	the place is clean and hygienic for kids and toddlers.	In this, sentiment is carried by the event of finding a dead mouse, hinting at the cleanliness and hygiene issues. Therefore, the context was removed and direct lexical markers - clean and hygienic - were introduced.
3	what were you going to charge me when i purchase a dozen bagels ?	surely you were going to charge me reasonably when i purchase a dozen bagels.	This is satiric in nature and expressed as a question with no expectation for an answer. Such sentences had to be flattened to transfer style.
4	it looks like blended up chicken put into the shape of grilled pieces.	the grilled chicken was made just perfect	The input sentence doesn't explicitly mentions what is good or bad. It simply related an event which is then perceived as negative.
5	what about the time i wasted?	service was quick and swift.	Again, it is not really a question but a comment on bad service veiled in the form of a question.
6	i should have stuck with sun chinese dining.	this was a very great place to dine in at.	Sarcasm is used to express displeasure, hence the entire lexical context was compromised during the transfer process.
7	was n't busy , no biggie .	was busy , no biggie .	Nothing here says if being busy was something good or bad.
8	there is a reason they can get you in fairly quickly.	This place is the most sought after.	Here the sentiment is implicit in the observation of the status of a venue, where the user uses sarcasm to mention that why you can get a table so quickly is that this place is not much preferred.

Table 3: Examples of handling implicit sentiments.

ID	Negative	Positive	Analysis
1	stopped by for soda after being at the hobby shop next door. After the hobby shop, I stopped in for a soda but hated it.	after the hobby shop i stopped in for a soda and enjoyed it.	The original sentence lacked sentiment, hence, a sentiment-bearing attribute had to be inserted.
2	i was so disgusted i could not ~~way~~ wait for the rest of the day.	i was so full i could not way for the rest of the day. I was so happy I could not wait for the rest of the day.	Spelling mistakes sometimes made it difficult to understand the sentence. In this particular example, the unclear context also increased difficulty.
3	i know i should have sent this back and walk out.	i know i shouldn't have sent this back and walked out. I relished my order.	Lexical context could not be preserved for the sake of naturalness.
4	i'm not one of the corn people .	i'm proud to be one of the corn people.	Cultural undertone resolution was a problem.
5	i got there, was seated pretty ~~quickly~~ late, and then couldn't chose my color.	we were seated quick as soon as we got there, then we glady chose colors.	The input was incorrectly identified as negative making both, the input and output positive, hence, one had to be manipulated for negative.
6	sadly, we've been to this long established restaurant many times.	the restaurant has been great throughout the years fortunately, we 've been to this long established restaurant many times.	The original input lacked sentiment.
7	liar, liar, pants on fire.	truth truth be told ! honest people	Proverbial expressions were difficult to deal with.
8	too bad it was at the expense of the other customers.	~~too bad~~ gladly, it wasn't at the expense of the other customers.	Here, the challenge was to stranger sentiment with as little loss of context as possible.
9	talk about false advertising so call before you go !	No need to call before you go. they are exactly what their advertising claims for them.	This is an example of a sentence where the sentiment is implicit and hence difficult to transfer.
10	so you aren't my problem.	don't worry, you weren't my problem. I'm glad you're not causing any concerns for me.	This is also an implicitly negative sentence, hence, difficult to transfer style as well as translate.
11	not sure, is that a good thing or a bad thing?	I bet it's a good thing, and not a bad thing.	Here an attempt was made to provide more clarity on the context.
12	when i first came to phx...yes this sounded indian to me. when i first came to phx...yes this sounded unpleasant to me.	when i first came to phx...yes this sounded american to me. when i first came to phx... yes this sounded pleasant to me.	When sentiment-bearing attributes were cultural signifiers, for example, here, 'indian' was made positive with 'american', we decided to work with pleasant and unpleasent.
13	you won't find a ~~better~~ worse selection in scottsdale.	you won't find a better selection in ~~arizona~~ scottsdale.	When the input sentence was incorrectly identified as negative, editing was required. It affected the decision-making process for the sentiment transfer.
14	if i could give zero stars i def would.	the stars was 5 plus If I could give more stars I def would.	We also made reasonable changes in the data where we did not want the model to establish a link between numbers carrying neg/pos relationship. For example, in the example below we didn't want zero-five relationship to be seen as a definite neg-pos relationship.

Table 4: Text Sentiment Transfer English dataset improvement challenges' Examples.

ID	Positive	Negative	Analysis
1	i highly recommend e & m painting. আমি অবশ্যই ই অ্যান্ড এমের পেইন্টিঙের সুপারিশ করব ।	I highly recommend avoiding e & m painting. আমি একদমই ই অ্যান্ড এমের পেইন্টিঙের সুপারিশ করব না ।	Here, instead of translating 'avoiding' NEG-marker was used to flip the sentiment to maintain naturalness.
2	everything is fresh and so delicious! সবকিছু খুব তাজা এবং সুস্বাদু ছিল !	everything is so stale and bland! সবকিছু খুব পুরনো এবং অসুস্বাদু ছিল ।	Oft-repeated words like 'bland' in this example have been translated consistently. Compare the translation of 'bland' with number 7 in Table 6.
3	the variety of sushi rolls makes for a good eating. খাওয়ার জন্য বিভিন্ন ধরনের শুষি রোল রয়েছে ।	There is limited variety for sushi rolls. শুষি রোলের জন্য সীমিত বৈচিত্র্য রয়েছে ।	In this example the lexical context of 'for a good eating' has been dropped to maintain naturalness in Bangla translation.
4	thanks for making our special night an event to remember. আমাদের বিশেষ রাতটা এতো স্মরণীয় বানানোর জন্য অনেকধন্যবাদ।	thanks for making our special night so horrible. আমাদের বিশেষ রাতটা নষ্ট করে দেওয়ার জন্য অনেক ধন্যবাদ।	Retaining the word 'thanks' adds a sarcastic touch during the style transfer process. The same is a challenge to maintain in Bangla.
5	when i first came to phx...yes this sounded unpleasant to me. যখন আমি প্রথম পিএইচএক্সে এসেছিলাম... এটাআমার কাছে অপ্রীতিকর শুনিয়েছিল ।	when i first came to phx... yes this sounded pleasant to me. যখন আমি প্রথম পিএইচএক্সে এসেছিলাম...এটা আমারকাছে প্রীতিকর শুনিয়েছিল ।	The use of the word 'yes' sounds forced in Bangla, hence had to be avoided.
6	what the hell are you doing ? তুমি এটা কি খারাপ কাজ করছ ?	you're doing great তুমি এটা ভালো কাজ করছ ।	The word 'hell' is a negative sentiment-bearing word that means other than the common noun hell.
7	but unfortunately the rude woman was the one checking us out. কিন্তু দুর্ভাগ্যবশত, অভদ্র মহিলাটিই আমাদের চেকআউট করছিলেন ।	but fortunately the polite woman was the one checking us out. কিন্তু সৌভাগ্যবশত, ভদ্র মহিলাটিই আমাদের চেক আউটকরছিলেন ।	The 'checking out' could mean checking out at the counter or a slang. The translation is force to dilute the ambiguity and maintain one meaning.
8	this place is a shit hole with shit service. এই জায়গাটা যেরকম বাজে সেরকমই বাজে এর পরিষেবা ।	this place is very nice with great service. এই জায়গাটা যেমন ভালো তেমনি ভালো তার পরিষেবা ।	Slang Words pose challenges in translation.

Table 5: English and Bangla Text Sentiment Transfer Examples (Positive to Negative).

ID	Negative	Positive	Analysis
1	but it probably sucks too ! কিন্তু এটাও সম্ভবত খুব খারাপ !	but it probably doesn't suck too ! কিন্তু এটাও সম্ভবত খুব একটা খারাপ নয় !	Here the negative sentiment bearing words is 'sucks' that does not have an exact translation in Bangla. Hence, an approximate word had to be used which limits the range of meaning 'sucks' carry in English.
2	Their chips are ok, but their salsa is really bland. তাদের চিপস ঠিক ছিল, তবে সালসাটা অসুস্বাদু ছিল ।	Their chips are good and their salsa is really tasty. তাদের চিপস ভালো ছিল, এবং সালসাটা অসাধারণ ছিল ।	Please refer to Example 4 in Table 5.
3	the wine was very average and the food was even less. ওয়াইনের স্বাদ মোটামোটি ছিল তবে খাবারের স্বাদ খুব খারাপ ছিল ।	the wine was above average and the food was even better. ওয়াইনের স্বাদ ভালো ছিল তবে খাবারের স্বাদ আরও ভালো ছিল ।	Here the subtlety has been compromised during the translation process. Both 'average' and 'even less' have been directly interpreted as 'bad'.
4	for the record i am not a good cook , i use seasoning ! মোট কথা আমি একজন ভালো রাঁধুনি নই, আমি শুধু মশলার সাহায্যে রান্না করি ।	for the record i am a terrific cook, i use seasoning ! আসলে আমি একজন অসাধারণ রাঁধুনি, আমি সব মশলা দিয়ে রান্না করি ।	Here Bangla translation uses 'spice' for 'seasoning' where seasoning is a broader term in English with no exact translation in Bangla.
5	this is an old worn out hotel. এটা একটা পুরনো, জীর্ণ হোটেল ।	this is an old vintage hotel. এটা একটা পুরনো, ভিনটেজ হোটেল ।	Although a Bangla translation for Vintage exists, yet transliteration was preferred not only to maintain consistency but also to retain the exact flavour of vintage and not deviate towards antiquity.
6	talk about false advertising so call before you go ! মিথ্যে বিজ্ঞাপনের কথা শুনছিলাম তাই যাওয়ার আগে ফোন করে নিও ।	they are exactly what their advertising claims for them. তারা যেটা বিজ্ঞাপন করে ঠিক সেটাই ।	To resolve implicit meaning in this sentence, similar phrases were used which preserved the naturalness but compromised on the lexical context.
7	not so much these days. আজকাল খুব একটা না ।	much more these days. আজকাল আরো অনেক বেশি ।	Here, the meaning is very unclear leading to an equally unclear translation
8	half of my head was over processed. আমার অর্ধেক মাথা আর কাজ করছিল না ।	half of my head was processed well. আমার অর্ধেক মাথা এখনো কাজ করছিল ।	To resolve implicit meaning in this sentence, similar phrases were used which preserved the naturalness but compromised on the lexical context.

Table 6: English and Bangla Text Sentiment Transfer Examples (Negative to Positive).

Intent Detection and Slot Filling for Home Assistants: Dataset and Analysis for Bangla and Sylheti

Fardin Ahsan Sakib, A H M Rezaul Karim, Saadat Hasan Khan, Md Mushfiqur Rahman
Department of Computer Science, George Mason University *
{fsakib,akarim9,skhan225,mrahma45}@gmu.edu

Abstract

As voice assistants cement their place in our technologically advanced society, there remains a need to cater to the diverse linguistic landscape, including colloquial forms of low-resource languages. Our study introduces the first-ever comprehensive dataset for intent detection and slot filling in formal Bangla, colloquial Bangla, and Sylheti languages, totaling 984 samples across 10 unique intents. Our analysis reveals the robustness of large language models for tackling downstream tasks with inadequate data. The GPT-3.5 model achieves an impressive F1 score of 0.94 in intent detection and 0.51 in slot filling for colloquial Bangla. [1]

1 Introduction

Smart devices have become commonplace, establishing home assistants as indispensable fixtures in contemporary households. These voice-activated virtual companions adeptly manage an array of tasks, ranging from setting reminders to controlling room temperatures. The efficacy of home assistants in performing these tasks is closely intertwined with their underlying Natural Language Understanding (NLU) models, which enable seamless interactions in high-resource languages (Chen et al., 2019; Stoica et al., 2021; Antoun et al., 2020; Upadhyay et al., 2018). However, this advantage in NLU capabilities is not extended to low-resource languages (Stoica et al., 2019; Schuster et al., 2018), presenting a notable discrepancy. This discrepancy holds considerable significance, especially considering the global demand for home assistants and the extensive usage of low-resource languages, which have a substantial speaker base.

Bangla and Sylheti (Ethnologue, 2023), with 285 million native speakers combined, have rich cultural and colloquial nuances. Specialized datasets are needed to capture these intricacies as users prefer to interact with home assistants in their native languages, highlighting the research need (Bali et al., 2019).

The language understanding of home assistants is dependent on two key NLU tasks: intent detection and slot filling (Weld et al., 2022; Louvan and Magnini, 2020). Intent detection determines user actions, like playing music or checking the weather, while slot filling extracts specific details, such as song titles or locations. These tasks enable seamless human-device interactions, especially for home assistants.

Research on intent detection and slot filling primarily focuses on high-resource languages (Liu and Lane, 2016; Qin et al., 2021; Niu et al., 2019; Zhang et al., 2018). While there have been limited studies dedicated to the Bangla language (Bhattacharjee et al., 2021; Alam et al., 2021; Hossain et al., 2020), none of them have addressed the tasks of intent detection and slot filling in Bangla. Furthermore, these studies have not taken into account colloquial variants or closely related languages like Sylheti. This gap in research leaves a significant portion of the speaker base underserved.

This paper bridges this research gap with several notable contributions. Firstly, we introduce a comprehensive dataset encompassing 328 entries for intent detection and slot filling for each of the three languages – totaling 984 samples. These languages include formal Bangla, colloquial Bangla, and colloquial Sylheti. We further show a comparative study between generative LLMs and state-of-the-art language models for intent detection and slot filling.

2 Dataset

At the core of our exploration stands a meticulously curated dataset that is inspired by the SNIPS dataset (Coucke et al., 2018), which caters to the broad audience.

[1]The dataset and the analysis code can be found in the following directory: https://github.com/mushfiqur11/bangla-sylheti-snips.git

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 48–55

2.1 Dataset Size and Distribution

Originating from the 328 English samples present in the SNIPS dataset, our dataset underwent a manual correction phase to ensure that the English samples were of optimal quality. Then, we created three linguistically diverse variants, maintaining the same distribution across intent classes and slots as the original samples. These are:

1. **Formal Bangla:** This represents the standard version of the Bangla language, majorly used in contexts like official documents, news broadcasts, and literature. Formal Bangla tends to adhere strictly to grammatical rules.

2. **Colloquial Bangla:** An informal variant predominantly used in Bangladesh, colloquial Bangla resonates with everyday conversations of its people. While there are numerous dialects in different regions of Bangladesh, this form remains more or less consistent across the country. Colloquial Bangla is more flexible regarding syntax and incorporates a significant number of loanwords from English, Arabic, Persian, and other languages.

3. **Colloquial Sylheti:** A language with unique intricacies, Sylheti stands apart from Bangla and is spoken in the Sylhet region of Bangladesh and among diaspora communities. It's rich in expressions, proverbs, and idiomatic language that reflect the history and culture of the Sylhet region.

The curated dataset spans 10 distinctive intents. Each specific intent has a distinct set of slot categories. Figure 1 shows the number of samples for each intent and Figure 2 shows the fraction of slots that frequently occur for each intent, with respect to infrequently occurring slots.

2.2 Data Generation Process

The generation of our dataset was methodical and rigorous to ensure authenticity and accuracy.

Annotator Engagement

Four doctoral students were on board as annotators for our project. The initial phase involving the rectification of English data from the SNIPS dataset was a collaborative effort, with each annotator working on a distinct, non-overlapping segment. Subsequent phases involved two individuals fluent in Bangla for the Bangla datasets and two native Sylheti speakers for the colloquial Sylheti dataset.

Base Creation

The base dataset was created using the Bangla-T5 model (Bhattacharjee et al., 2023), a state-of-the-art English-to-Bangla translation tool, following the work of De bruyn et al.. The refined English samples served as the foundation to produce the initial Bangla translations for each sample. An auto-generated dataset comes with a myriad of issues. Therefore, these samples were manually re-translated and annotated with the auto-translations as the base.

Inter-Annotator Agreement

An essential step in ensuring the reliability of our dataset was to gauge the consistency between annotators. For each language variant, 28 randomly chosen samples were annotated independently by both designated annotators, followed by calculating their inter-annotator agreement (Table 1). This exercise helped us discern the degree of concordance and areas of divergence.

Consensus Building

Post the initial agreement calculation, a meeting was convened where the annotators discussed and reconciled their differences. This step was instrumental in ironing out inconsistencies and ensuring a unified approach going forward.

Blind Overlap

As the annotators progressed with data creation, a random 10% of the samples were earmarked for blind overlap. These served as a secondary check on inter-annotator agreement after dataset creation.

Independent Adjudication

After the final compilation of the dataset, each entry underwent a rigorous review by an independent adjudicator who had not previously worked on that particular language variant. This added an additional layer of scrutiny and quality assurance.

Inter-annotator agreement		
	Cohen's Kappa	Average BLEU
First 28 samples	0.42	0.43
Blind overlap (10%)	0.55	0.51

Table 1: There was an increase in annotator agreement before and after the annotator's meeting. This ensures the homogeneity of annotations in the dataset.

2.3 Ensuring Quality

Our data generation process, featuring multiple checks, blind overlaps, third-party reviews, and inter-annotator agreement stages, highlights our

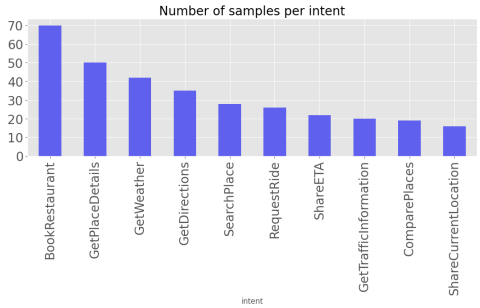


Figure 1: The number of samples for each intent varies, but they are fairly distributed, with 18 to 68 samples per intent.

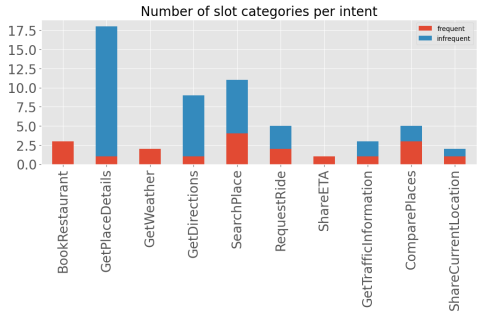


Figure 2: Slot categories appearing in at least 30% of the instances are marked as "frequent," while others are "infrequent." Despite varying slot categories per intent, frequent ones are evenly distributed.

commitment to quality. It minimizes biases and discrepancies that could result from a single annotator's viewpoint. The inclusion of an independent adjudicator in the final review further bolsters the dataset's integrity and reliability. Using a well-established dataset as the baseline ensures proper distribution of the data across different labels (Figure 1 and Figure 2).

3 Methodology and Experimental Setup

Our experiments were divided into four phases. In our initial experiment, we employed JointBERT (Chen et al., 2019), the state-of-the-art model in this domain, for both intent detection and slot-filling tasks. In our next experiment, JointBERT was retained for intent detection, while we explored the capabilities of GPT-3.5 (Generative Pre-trained Transformer) (Brown et al., 2020) model for slot filling. The third experiment fully utilized GPT-3.5 for both tasks. For our concluding experiment, we provided GPT-3.5 with the original intents and then analyzed its performance on the slot-filling task. The final experiment gives the raw result of slot-filling for the GPT model.

JointBERT leverages the BERT (Devlin et al., 2019) model to provide a unified approach encompassing both intent classification and slot filling by utilizing the representations from the pre-trained BERT model. We employed the default BERT tokenizer and maintained consistent parameters for all three languages. The utilization of these default settings and tokenization methods ensures an equitable and consistent evaluation across the languages.

GPT-3.5 (Generative Pre-trained Transformer) (Brown et al., 2020) model operates on the Transformer architecture and is adept at generating text resembling human language by predicting subsequent words or tokens in a sequence. GPT-3.5's deep contextual understanding is a result of extensive pre-training on a diverse corpus of textual data, encompassing various languages and linguistic intricacies enabling it to excel across a spectrum of NLP tasks (Goyal et al., 2022; Liu et al., 2021; Sakib et al., 2023; Kumar et al., 2020). We used GPT in a few-shot setting, passing 5 training samples along with the prompt. Rigorous prompt engineering was performed before settling on the two prompts for the two tasks. Figure 3 and Figure 4 show the final versions of the prompts used in the experimentations.

3.1 Experimental Setup

We divided each of the three datasets into training, development, and test sets using a standard 80-10-10 split. The JointBERT model was trained and evaluated on an A100 GPU, using a batch size of 8. We closely followed the setup provided by the original authors for this phase. For GPT, we used the OPENAI API with the "GPT-3.5-turbo" engine and set the token limit to 50.

4 Results

Tables 2 and 3 present the performance of the models we evaluated on our intent detection and slot-filling tasks. A clear pattern emerges: GPT-3.5 consistently outperforms JointBERT in both tasks.

While intent detection is generally more straightforward, JointBERT performs reasonably well in this aspect, although it doesn't quite match the exceptional performance achieved by GPT-3.5. However, when it comes to the more intricate task of slot-filling, JointBERT's performance falls significantly short, leaving ample room for improvement. In contrast, GPT-3.5 demonstrates its proficiency

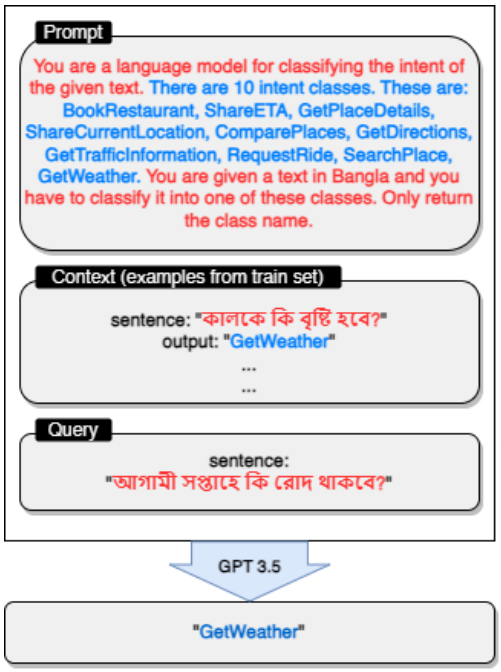


Figure 3: The figure illustrates how the input is formatted for the intent-detection task. A base-prompt is passed on to the GPT model. A few samples (5) from the training set are also passed as the context. From these sentence-output pairs, the LLM understands how the task needs to be solved. Finally, the current query is passed

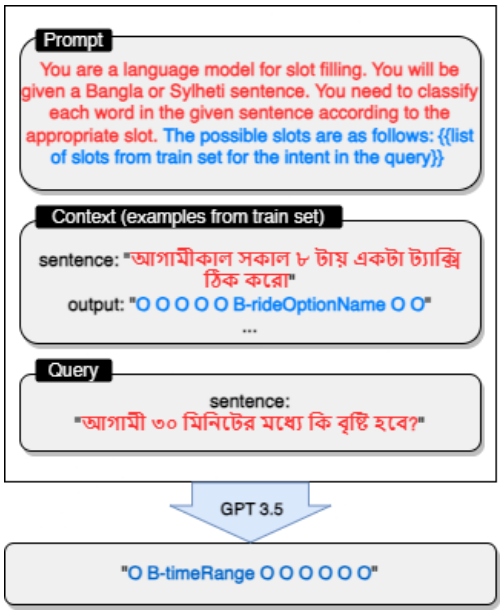


Figure 4: The input structure for the slot-filling task is quite similar to the intent detection task. The major difference is the prompt. For slot-filling, the set of possible slots is based on the intent type of the query. The intent type is obtained from a separate model and then from the train set, all possible slots for the given intent are fetched

Intent Detection (*Accuracy and F1 Score*)			
Models	Formal Bangla	Colloquial Bangla	Colloquial Sylheti
JointBERT	0.57 | 0.56	0.63 | 0.61	0.45 | 0.46
GPT-3.5	0.94 | 0.94	0.94 | 0.94	0.87 | 0.89

Table 2: While the performance of JointBERT is noteworthy for Bangla and its variants, the GPT-3.5 model excels across all metrics for all three datasets

Slot Filling (*F1 Score*)				
Slot Filling Model	Intent From	Formal Bangla	Colloquial Bangla	Colloquial Sylheti
JointBERT	JointBERT	0.14	0.11	0.07
GPT-3.5	JointBERT	0.43	0.45	0.52
GPT-3.5	GPT-3.5	0.45	0.51	0.57
GPT-3.5	Original	0.54	0.53	0.57

Table 3: The slot-filling task is separate from but dependent on the intent detection task. Intent needs to be passed to the model for good performance. In slot-filling tasks, GPT massively outperforms JointBERT

in handling the complexities of this task.

A significant reason behind GPT-3.5's superior performance is its broader exposure to diverse languages during training, including Bangla. JointBERT, conversely, hasn't been specifically trained on any Bangla dataset. This linguistic familiarity gives GPT-3.5 a clear advantage, enabling it to process and interpret Bangla's nuances far more effectively than JointBERT. The results underline the significance of using LLMs for low-resource languages, especially in scenarios where obtaining high volumes of training data for a particular downstream task is challenging.

5 Conclusion

In the era of smart devices, a home assistant's voice interfaces must resonate with the authentic linguistic intricacies of its users. Our research presents the first-ever dataset for intent detection and slot filling in Bangla and Sylheti, emphasizing their colloquial forms. This focus on colloquial forms bridges the often-overlooked gap between formal language models and the nuances of everyday speech. By championing colloquial forms, we ensure a voice interface that's more natural and attuned to genuine communication habits. Through rigorous data collection and validation, we have produced a high-quality benchmark dataset, providing a solid foundation for subsequent analyses and model evaluations. The comparative study between large lan-

guage models (LLM) like GPT-3.5 and non-LLMs underscores the remarkable capability of LLMs to excel even with minimal datasets, marking a considerable stride for underrepresented languages.

6 Limitations

While our research has made significant strides in understanding intent detection and slot filling for Bangla and Sylheti, like any study, it has its limitations. Our dataset, although carefully curated for the Bangla and Sylheti variants, is on the smaller side compared to established benchmarks. A precise and robust data generation process was prioritized, naturally limiting our data volume. We confined our evaluations to the JointBERT model and GPT-3.5. The pronounced difference in their performance deterred us from testing a broader range of models. Moreover, the dearth of optimized Bangla models for specific tasks posed challenges. An attempt with a Bangla BERT tokenizer didn't yield satisfactory outcomes, affecting the JointBERT's efficacy. As promising as our results are, they are tied to our specific dataset and context. Extending our findings to diverse settings or other languages requires further exploration, marking just the beginning of this exciting journey.

References

Masoud Akbari, Amir Hossein Karimi, Tayyebeh Saeedi, Zeinab Saeidi, Kiana Ghezelbash, Fatemeh Shamsezat, Mohammad Akbari, and Ali Mohades. 2023. A persian benchmark for joint intent detection and slot filling.

Firoj Alam, Arid Hasan, Tanvirul Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. arxiv 2020. *arXiv preprint arXiv:2003.00104*.

Kalika Bali, Monojit Choudhury, Sunaya Sitaram, and Vivek Seshadri. 2019. Ellora: Enabling low resource languages with technology. In *Proceedings of the 1st International Conference on Language Technologies for All*, pages 160–163.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, and Rifat Shahriyar. 2023. Banglanlg and banglat5: Benchmarks and resources for evaluating low-resource natural language generation in bangla. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 714–723.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2021. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *arXiv preprint arXiv:2101.00204*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Mai Hoang Dao, Thinh Hung Truong, and Dat Quoc Nguyen. 2021. Intent detection and slot filling for vietnamese. *arXiv preprint arXiv:2104.02021*.

Maxime De bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2022. Machine translation for multilingual intent detection and slots filling. In *Proceedings of the Massively Multilingual Natural Language Understanding Workshop (MMNLU-22)*, pages 69–82, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ethnologue. 2023. Ethnologue 200: Languages of the world. Accessed on September 3, 2023.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. Banfakenews: A dataset for detecting fake news in bangla.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. *arXiv preprint arXiv:2011.00564*.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*, pages 3771–3775.

Peiqing Niu, Zhongfu Chen, Meina Song, et al. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. *arXiv preprint arXiv:1907.00390*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. *arXiv preprint arXiv:1909.02188*.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197. IEEE.

Suman Ravuri and Andreas Stolcke. 2015. Recurrent neural network and lstm models for lexical utterance classification. In *Sixteenth annual conference of the international speech communication association*.

Fardin Ahsan Sakib, Saadat Hasan Khan, and AHM Karim. 2023. Extending the frontier of chatgpt: Code generation and debugging. *arXiv preprint arXiv:2307.08260*.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2018. Cross-lingual transfer learning for multilingual task oriented dialog. *arXiv preprint arXiv:1810.13327*.

Anda Stoica, Tibor Kadar, Camelia Lemnaru, Rodica Potolea, and Mihaela Dînşoreanu. 2019. The impact of data challenges on intent detection and slot filling for the home assistant scenario. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 41–47. IEEE.

Anda Stoica, Tibor Kadar, Camelia Lemnaru, Rodica Potolea, and Mihaela Dînşoreanu. 2021. Intent detection and slot filling with capsule net architectures for a romanian home assistant. *Sensors*, 21(4):1230.

Gokhan Tur, Li Deng, Dilek Hakkani-Tür, and Xiaodong He. 2012. Towards deeper understanding: Deep convex networks for semantic utterance classification. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5045–5048. IEEE.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6034–6038. IEEE.

Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. 2016. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6060–6064. IEEE.

Jixuan Wang, Kai Wei, Martin Radfar, Weiwei Zhang, and Clement Chung. 2020. Encoding syntactic knowledge in transformer encoder for intent detection and slot filling.

Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2022. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Computing Surveys*, 55(8):1–38.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 ieee workshop on automatic speech recognition and understanding*, pages 78–83. IEEE.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. 2018. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, volume 16, pages 2993–2999.

Zhichang Zhang, Zhenwen Zhang, Haoyuan Chen, and Zhiman Zhang. 2019. A joint learning framework with bert for spoken language understanding. *Ieee Access*, 7:168849–168858.

A Appendix

A.1 Related Work

Efforts to enhance datasets for intent detection and slot-filling within low-resource languages, such as Bangla and Sylheti in this context, commence with the intricate process of translating individual English lexemes extracted from established benchmarks like ATIS and SNIPS. Previous works in intent detection and slot filling for low resource languages(Dao et al., 2021; Akbari et al., 2023), have translated each English utterance to their respective languages. Recent works have shown that there are great performance achievements on intent detection and slot-filling tasks on datasets that have been derived from the SNIPS dataset (Weld et al., 2022; Qin et al., 2019; Wang et al., 2020), and this gives a reason to choose the SNIPS dataset over the ATIS dataset as it is a good starting point for a work with a language that has never been explored.

Spoken Language Understanding, a pivotal endeavor in the domain of task-oriented dialogue systems, encompasses the tasks of intent detection and slot-filling. Traditionally, these tasks were regarded as distinct domains in which significant progress was made (Tur et al., 2012; Ravuri and Stolcke, 2015; Mesnil et al., 2013; Vu et al., 2016). However, recent research has garnered notable attention by achieving remarkable advancements in performance through the concurrent learning of intent detection and slot-filling tasks (Zhang et al., 2018; Weld et al., 2022). In this section, we're primarily looking at how intent detection and slot-filling tasks are combined. We'll focus on two well-known strategies for this integration:

- A strategy devised through parameter sharing and the exchange of hidden states, utilizing a common BiLSTM/BERT encoder, along with two distinct decoders dedicated to intent detection and slot filling, on top of the shared encoder. (Chen et al., 2019; Xu and Sarikaya, 2013; Liu and Lane, 2016; Zhang and Wang, 2016).

- Another strategy, extending the initial approach to a more advanced level, involves the model acquiring an understanding of the relationships between slots and intent labels. This frontier has been explored in research in two distinct ways. Some studies (Goo et al., 2018; Li et al., 2018; Niu et al., 2019) have demonstrated the use of attention mechanisms to discern the correlation between the overarching intent context representation and the slot vectors generated by the encoder. Alternatively, other works (Qin et al., 2019; Zhang et al., 2019) have approached this by initially learning the representation of the utterance, which aligns with the representation of the global intent context, utilizing a self-attention mechanism. Subsequently, they join this representation with the encoder's vector outputs before feeding the combined vectors into the slot-filling decoder.

A.2 Examples from the dataset

Here we include a few examples from each of the datasets.

"আমার এয়ারবিএনবির কাছে একটি রেস্টুরেন্টে রাত ৮:৪৫ মিনিটের জন্য একটি টেবিল বুক করুন"- BookRestaurant- "O O O O O O O B-restaurant O O O O O O O O O O"

"আমি বাড়ি না আসা পর্যন্ত আমার বয়ফ্রেন্ডের সাথে আমার অবস্থান শেয়ার কর"- ShareCurrentLocation- "O O O O O B-contact I-contact I-contact O O O O O "

"আজ রাতে আমার ডিনারে যাওয়ার জন্য একটা উবার ডাকো" -RequestRide- "O O B-destination I-destination I-destination O O O B-rideOptionName I-rideOptionName O O"

Figure 5: Few examples from the Formal Bangla dataset. (Input sentence - Intent - Expected slots)

A.3 Prompts used for GPT

For the intent detection task we used the following prompt: *"You are a language model for classifying the intent of the given text. There are 10 intent classes. These are: BookRestaurant, ShareETA, GetPlaceDetails, ShareCurrentLocation, CompareePlaces, GetDirections, GetTrafficInformation, Re-*

"কফি ক্লাব কি সিপ কফি চেয়ে সস্তা?"- ComparePlaces- "B-place1 I-place1 O B-place2 I-place2 O O O"

"বদরুল কে আমার পৌঁছানোর সময় জানিয়ে একটা message পাঠাও"- ShareETA- "B-contact I-contact O O O O O O O O O "

"আমি আম্বারখানার যে রাস্তা দিয়ে আমার client meeting এ যাব সেইদিকে কি জ্যাম আছে?" -- GetTrafficInformation- "B-origin B-way I-way I-way I-way I-way O B-destination I-destination I-destination O O O O O O O O"

Figure 6: Few examples from the Colloquial Bangla dataset. (Input sentence - Intent - Expected slots)

"আমার মীরবাজার যাওয়ার লাগি ৫ মিনিট ওর বিত্রে এখটা ট্যাক্সি লাগব"- RequestRide- "O B-origin I-origin O O O O O O O O O B-rideOptionName O O"

"আমি বাড়িত জাইতাম কিলান?"- GetDirections- "O B-destination I-destination O O O O O O"

"আমি কিতা আমানুল্লাহর সামনে পার্ক খরতে ফারমু নি?" -- GetPlaceDetails- "O O O B-place I-place I-place O O O O O O O O O"

Figure 7: Few examples from the Sylheti dataset. (Input sentence - Intent - Expected slots)

questRide, SearchPlace, GetWeather. You are given a text in Bangla and you have to classify it into one of these classes. Only return the class name."

In this approach, we clearly outlined the potential intent classes, specified the input language as Bangla, and directed the model to solely return the class name. Such structuring was essential to elicit precise responses from the model.

For our slot-filling task, we utilized the following prompt: *"You are a language model for slot filling. You will be given a Bangla sentence. You need to classify each word in the given sentence according to the appropriate slot. The possible slots are as follows: list of possible slots extracted from the train set (based on the training intent)"*

We equipped the model with both the potential slots and their associated intent. Notably, the performance fluctuated depending on the source of the intent— GPT-3.5, JointBERT, or the Original dataset.

A.4 Inter-annotator metrics

In order to assess inter-annotator agreement, this study utilized two primary evaluation metrics: Cohen's Kappa and Average BLEU.

Cohen's Kappa provides a statistical measure of agreement between two annotators, while accounting for the possibility of chance agreement. Specifically, it involves calculating the actual observed agreement between the annotators and comparing that to the level of agreement that would be expected by random chance. Cohen's Kappa expresses the ratio between these two values as a score ranging from 0 to 1, with higher scores indicating greater reliability.

Average BLEU (Bilingual Evaluation Understudy) is a commonly employed metric for evaluating machine translation outputs by comparing them against one or more reference translations. It analyzes the co-occurrence of n-grams between the translated text and human reference texts to produce a score reflecting the quality and fluency of the translation. Taking the average BLEU score across multiple translations provides an overall indicator of the fidelity of the translations with respect to the reference materials.

Together, these two metrics enable analysis of both the reliability of individual annotators via Cohen's Kappa and the accuracy and fluency of translations via Average BLEU in relation to trusted references. The combination provides a robust means of evaluating key aspects of annotation quality for this study.

BEmoLexBERT: A Hybrid Model for Multilabel Textual Emotion Classification in Bangla by Combining Transformers with Lexicon Features

Ahasan Kabir, Animesh Chandra Roy, Zaima Sartaj Taheri
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chattogram-4349, Bangladesh
{ahasankabir146, animeshroycse, zstaheri1999}@gmail.com

Abstract

Multilevel textual emotion classification involves the extraction of emotions from text data, a task that has seen significant progress in high-resource languages. However, resource-constrained languages like Bangla have received comparatively less attention in the field of emotion classification. Furthermore, the availability of a comprehensive and accurate emotion lexicon specifically designed for the Bangla language is limited. In this paper, we present a hybrid model that combines lexicon features with transformers for multilabel emotion classification in the Bangla language. We have developed a comprehensive Bangla emotion lexicon consisting of 5336 carefully curated lexicons across nine emotion categories. We experimented with pre-trained transformers including mBERT, XLM-R, BanglishBERT, and BanglaBERT on the EmoNaBa (Islam et al., 2022) dataset. By integrating lexicon features from our emotion lexicon, we evaluate the performance of these transformers in emotion detection tasks. The results demonstrate that incorporating lexicon features significantly improves the performance of transformers. Among the evaluated models, our hybrid approach achieves the highest performance using BanglaBERT(large) (Bhattacharjee et al., 2022) as the pre-trained transformer along with our emotion lexicon, achieving an impressive weighted F_1 score of 82.73%. The emotion lexicon is publicly available at https://github.com/Ahasannn/BEmoLex-Bangla_Emotion_Lexicon

1 Introduction

Multilabel emotion classification involves the assignment of several emotion labels to a provided text. This allows for a more comprehensive representation to understand underlying emotional content. However, achieving accurate multilabel emotion classification in resource-constrained languages presents a unique challenge. Limited availability of annotated data, linguistic diversity, and cultural variations pose significant hurdles. Additionally, the lack of comprehensive emotion lexicons specific to these languages further complicates the task. To tackle these challenges, we present an innovative approach that combines the power of transformers with emotion lexicon features for multilabel emotion classification in Bangla. By leveraging pretrained transformer models and developing an extensive Bangla emotion lexicon, we aim to enhance the accuracy and effectiveness of emotion classification.

2 Related Work

Emotion detections from textual content have gained considerable attention in recent years. While extensive research has been carried out in high-resource languages such as English, Chinese, and Arabic, there remains a scarcity of studies specifically targeting emotion detection and classification in the Bangla language. Irtiza Tripto and Eunus Ali (2018) presented an LSTM-based method for emotion classification in Bangla YouTube comments by achieving 59.23% accuracy. Pal and Karn (2020) used logistic regression for detecting emotions (joy, anger, sorrow, suspense) from Bangla short stories. Rayhan et al. (2020) predicted six emotions from 7214 Bangla texts with the CNN-BiLSTM model outperforming BiGRU, achieving 66.62% accuracy. Das et al. (2021) developed s corpus of 6,523 texts for classifying six emotion categories employing various transformer models, among which XLM-R showed the best results with an F_1-score of 69.73%. Parvin et al. (2022) developed an emotion corpus comprising 9,000 Bangla texts in six emotion categories and proposed a weighted ensemble of CNN and BiLSTM. Islam et al. (2022) introduced a manually annotated Bangla noisy dataset comprising of 22,698 Bangla texts from

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 56–61

various domains, labeled for six fine-grained emotion categories.

Researchers found that combining transformers with handcrafted features has enhanced performance in various tasks. For abusive language detection Koufakou et al. (2020) combined lexicon features with BERT and found improved results in four different datasets. Similarly, De Bruyne et al. (2021) combined Dutch transformer-based BERT models named BERTje and RobBERT with lexicon-based methods, resulting in a marginal performance improvement. There are many emotion lexicons have been developed in English but not that much available in Bangla. Mohammad and Turney (2013a) has developed a crowdsourced English lexicon named NRC EmoLex consists of 14,000 lemmas labeled in Plutchik (1980) eight basic emotions and two sentiments. Abdaoui et al. (2017) used a semi-automatic translation method to construct French expanded emotion lexicon named FEEL, from the NRC Emolex.

3 Methodology

3.1 Text Preprocessing

In this paper, we have experimented with both raw and preprocessed texts. We have used (Islam et al., 2022) dataset for fine tuning our transformers which is built from user comments from various social media sites. To retain relevent features and to eliminate unnecessary complexities, we have preprocessed the texts. The preprocessing steps are shown in Figure 1. As social media data's contains many hyperlinks and user mentions , we have removed them from our texts. Emojis and Emoticons convey rich emotional information. To standardize their representation, we replaced emoticons and emojis with their corresponding word formats. Punctuation marks, such as question mark and exclamation marks carry significant emotional information, we have replaced them with special keyword tokens and removed other insignificant punctuation. We have removed special symbols and stopwords in the preprocessing stage.

3.2 Development of Emotion Lexicon

We have developed a Bangla Emotion Lexicon named BEmoLex, consisting of 5336 lexicons across 9 emotion classes: Love, Joy, Surprise, Anger, Sadness, Fear, Disgust, Trust, and Anticipation. A semi-automatic translation (Abdaoui et al., 2017) method was followed to generate the Bangla lexicon by leveraging existing English lexical resources, especially the NRC Emolex (Mohammad and Turney, 2013b). The initial automated translations done by Google Translate[1] were subsequently reviewed and validated by three human translators who are proficient in both English and Bangla. The translators can add a new term, remove an existing term, or make necessary adjustments to ensure an accurate representation of emotions considering the cultural and linguistic nuances of the Bangla language. To enhance the coverage and diversity of the lexicon, we have manually incorporated handpicked strong emotive words, and expanded terms with Bangla synonyms.

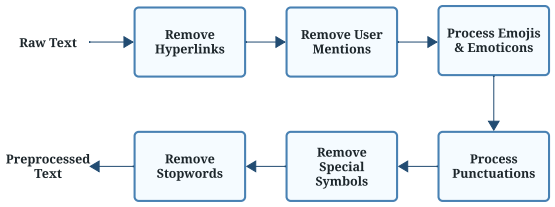


Figure 1: Text Preprocessing Steps

Table 1 provides an overview of the emotion lexicon, detailing the count of lexicons distributed across various emotion categories. The data highlights substantial coverage in the Anger, Sadness, Fear, and Anticipation classes, each containing an extensive lexicon count, surpassing 700 entries. In contrast, the Love and Surprise categories exhibit relatively lower lexicon counts, with 356 and 301 entries, respectively. The Trust, Joy, and Disgust categories are relatively balanced, each contributing approximately 10% of the lexicon entries, ensuring a comprehensive representation of emotions within the dataset.

3.3 Development of Hybrid Model

In this section, we presents our hybrid model called BEmoLexBERT by integrating lexicon features with transformer-based models. We have used pre-trained BERT models and fine-tuned them for multilevel emotion classification. We preprocessed each raw text and tokenized them. The tokens were given as input into the BERT layer as shown in Figure 2.

For each target text, a lexicon vector is created, and each vector is appended to the lexicon encoding. The dimensions of the vector align with the

[1] https://translate.google.com/

Love	Joy	Surprise	Anger	Sadness	Fear	Disgust	Trust	Anticipation
356	603	301	756	788	752	513	543	724

Table 1: Number of Lexicons in Each Emotion Category

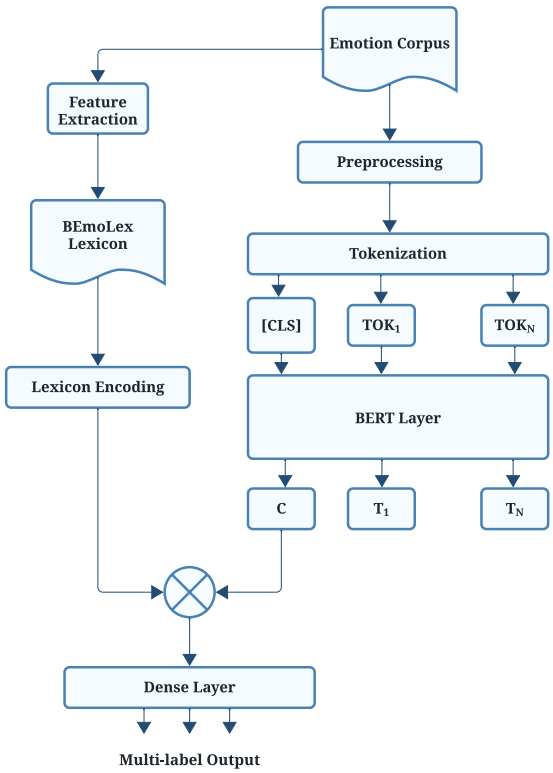


Figure 2: BEmoLexBERT, a hybrid model incorporating lexicon features with transformer

emotion categories in the training data. We perform a search for each term in the target text within the lexicon, incrementing the corresponding lexicon feature value for any matched emotion. The lexicon encoding vector is concatenated with the [CLS] token representation from the BERT layer, creating a comprehensive feature vector that effectively integrates both contextualized information and explicit emotion-related information. Finally, the combined feature vector is passed through a dedicated classification layer for Multilevel emotion classification.

4 Results and Analysis

We conducted a total of 24 experiments to evaluate the model we advocate. We have experimented with six pre-trained transformers as shown in Table 2 on both the plain model and hybrid model for both raw texts and preprocessed texts. The transformers were fetched from the Hugging Face [2] transformer library. Pretrained transformers are fine-tuned with 20,468 instances of the training section and tested with 2,272 instances of the testing section from the EmoNaBa dataset (Islam et al., 2022). Throughout the experiments, a consistent batch size of 8 was maintained. We have trained the models for 20 epochs and the learning rate was 2e-5. The experiments were carried out in a Google Colab [3] environment. The weighted F_1 score was selected as the primary evaluation metric.

The outcomes of our proposed hybrid model are presented in Table 3, showcasing a comparative analysis with a plain model concerning both raw and preprocessed texts. A noteworthy observation is that the preprocessing steps yielded only marginal enhancements over the raw text inputs. Remarkably, m-BERT demonstrated the most substantial improvement, achieving a 0.82% increase in the F1 score following preprocessing in the plain model. To further understand these marginal gains, we conducted a manual inspection of each preprocessing step. The results revealed that the primary improvement stemmed from the conversion of emojis and emoticons into textual forms, while the impact of the other preprocessing steps remained negligible. This observation underscores the proficiency of large language models in effectively handling noisy data, thereby minimizing the necessity of rigorous preprocessing. Furthermore, in the plain model, we can notice a slight improvement in the F_1 score when transitioning from XLM-R to XLM-R (large), showing an improvement of 3.22%. Similarly, the F_1 score increased by 3.42% when transitioning from BanglaBERT to BanglaBERT (large). Among the multilingual models, it is evident that they performed relatively lower compared to the monolingual models.

Across all transformers, the hybrid model demonstrated improved performance compared to the plain model. In the case of BanglaBERT, we observed a notable increase of 2.42% in the F_1 score for the hybrid model. For BanglaBERT

[2] https://huggingface.co/
[3] https://colab.google/

Pretrained Transformers	Parameters	Language	Reference
BanglaBERT BanglaBERT (large)	110M 335M	Bangla	(Bhattacharjee et al., 2022)
BanglishBERT	110M	Bangla & English	
mBERT	180M	Multilingual	(Devlin et al., 2018)
XLM-R (base) XLM-R (large)	270M 550M	Multilingual	(Conneau et al., 2019)

Table 2: Description of Pretrained Transformers Used for Experiments

Transformers	Plain Model				Hyrbrid Model			
	Raw Text		Preprocessed Text		Raw Text		Preprocessed Text	
	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
m-BERT	55.67	68.07	56.03	68.89	57.19	70.17	57.61	70.23
XLM-R	61.11	73.23	61.39	73.37	62.12	75.31	62.33	75.78
XLM-R (large)	62.92	76.66	62.81	76.59	63.15	77.61	63.07	77.35
BanglishBERT	62.35	76.39	62.97	76.86	64.33	77.94	64.59	78.27
BanglaBERT	64.84	77.96	65.03	78.25	66.39	80.07	66.73	80.67
BanglaBERT(large)	67.02	81.23	67.11	81.67	68.05	82.67	68.17	82.73

Table 3: Result comparison of pretrained transformers on plain model & hybrid model, both for raw texts & preprocessed texts. Acc denotes Accuracy in percentage and , F_1 denotes weighted F_1-score.

(large), a substantial 1.06% increase in F_1 score was achieved. Similarly, the performance of XLM-R (large) and XLM-R showed improvements with the hybrid model, presenting gains of 0.76% and 2.41%, respectively. Moreover, m-BERT and BanglishBERT displayed enhanced results, boasting improvements of 1.34% and 1.41% in F_1 score, respectively, when utilizing our hybrid model on preprocessed data. These consistent findings underscore the remarkable effectiveness of our proposed hybrid model, which skillfully incorporates lexicon features to deliver enhanced performance compared to the plain model.

We have manually identified some instances where the plain model failed but the hybrid model succeeded. One such example is the sentence, "ছোট বেলার এতো কাছের বন্ধু এমন মীরজাফর হয়ে যাবে ভাবি নি !" (I never thought that such a close childhood friend would become Mirjafar!). In the Bengali language, the word "মীরজাফর" (Mirjafar) is metaphorically used to describe someone who has deceived or cheated. The plain model detected the emotions of Sadness and Surprise in this sentence solely based on contextual analysis. In contrast, the hybrid model correctly identified it as a combination of Sadness, Anger, and Surprise emotions. The word "মীরজাফর" (Mirjafar) is classified under the Anger category in our emotion lexicon. The hybrid model, in addition to contextual analysis, effectively leveraged the lexical information of "মীরজাফর" (Mirjafar) within the sentence, enhancing its focus on the Anger emotion category. This example highlights the hybrid model's proficiency in capturing nuanced emotional cues by combining lexical and contextual information.

4.1 Comparison with existing works

In order to assess the effectiveness of our hybrid model, a comparison is conducted with existing techniques in the field. The previous methods (Pal and Karn, 2020; Rayhan et al., 2020; Das et al., 2021) are implemented on the EmoNaBa (Islam et al., 2022) dataset, and the outcomes are measured using the weighted F_1-score. To accommodate the multilabel emotion classification, necessary adjustments are made to convert the previous multiclass approaches (Pal and Karn, 2020; Das et al., 2021).

Methods	F_1	Reference
TF-IDF + LR	64.28	(Pal and Karn, 2020)
CNN + BiLSTM	68.57	(Rayhan et al., 2020)
XLM-R	73.23	(Das et al., 2021)
BEmoLexBERT	**82.73**	Proposed

Table 4: Performance comparison with existing works. F_1 denotes weighted F_1-score in percentage. The best score is denoted with bold letters.

In table 4 the results indicate that deep learning-based approaches (Rayhan et al., 2020) outperform machine learning-based approaches (Pal and Karn, 2020) in our test data. However, it is observed that the transformer based models (Das et al., 2021) demonstrating superior results than the deep learning methods and machine learning methods. Our proposed hybrid model, BEmoLexBERT, in combination with BanglaBERT(large) and the BEmoLex emotion lexicon, outperforms existing techniques for multi-label emotion classification in Bangla. It achieves an impressive F_1-score of 82.7%.

5 Conclusion

In this study, we have introduced BEmoLexBERT, a novel hybrid model that integrates transformers with lexicon features to enhance multilabel emotion detection in Bangla texts. A critical component of this work is the development of BEmoLex, a specialized emotion lexicon tailored to the nuances of the Bangla language. This lexicon encompasses a comprehensive repository of 5,336 unique lexicons, thoughtfully categorized into nine distinct emotion classes.

Our comprehensive evaluation, involving a comparative analysis between our proposed hybrid model and the plain model, underscores the efficacy of our approach in significantly enhancing emotion detection performance. Notably, the monolingual models outperformed their multilingual counterparts, while the examination of preprocessing steps revealed their marginal benefits, suggesting that large language models are proficient in managing noisy data, thereby reducing the necessity for extensive preprocessing.

Furthermore, we conducted a comparative assessment with other existing models, and the results underscored the state-of-the-art performance achieved by BanglaBERT(large) in conjunction with the BEmoLex lexicon. These findings collectively highlight the potential and significance of our approach in advancing multilabel emotion classification in the context of the Bangla language.

Limitations

The success of our approach heavily depends on the comprehensiveness of the emotion lexicon. Words and expressions that are not part of the lexicon may be overlooked, leading to inaccurate results. Lexicons require continuous updates and maintenance as languages evolve, and new words or expressions emerge. While transformers excel at understanding context, there might be cases where lexicon-based features do not align perfectly with the contextual analysis.

Ethics Statement

We acknowledge that bias in emotion classification models is an important ethical concern. We have conducted a meticulous review of our training data and lexicon to ensure that our models do not reinforce stereotypes or biases, taking into account the intricate linguistic and cultural intricacies of the Bangla language. Our lexicon, thoughtfully curated, is a testament to our commitment to respecting the rich cultural diversity and sensitivities of the Bangla-speaking community.

Acknowledgements

We sincerely appreciate the anonymous reviewers and our pre-submission mentor for their valuable insights and constructive suggestions, which contributed to the enhancement of this work.

References

Amine Abdaoui, Jérôme Azé, Sandra Bringay, and Pascal Poncelet. 2017. Feel: a french expanded emotion lexicon. *Language Resources and Evaluation*, 51(3):833–855.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Avishek Das, Omar Sharif, Mohammed Moshiul Hoque, and Iqbal H Sarker. 2021. Emotion classification in a resource constrained language using transformer-based approach. *arXiv preprint arXiv:2104.08613*.

Luna De Bruyne, Orphée De Clercq, and Véronique Hoste. 2021. Emotional robbert and insensitive

bertje: combining transformers and affect lexica for dutch emotion detection. In *Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA), held in conjunction with EACL 2021*, pages 257–263. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6.

Khondoker Ittehadul Islam, Tanvir Yuvraz, Md Saiful Islam, and Enamul Hassan. 2022. Emonoba: A dataset for analyzing fine-grained emotions on noisy bangla texts. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 128–134.

Anna Koufakou, Endang Wahyu Pamungkas, Valerio Basile, Viviana Patti, et al. 2020. Hurtbert: Incorporating lexical features with bert for the detection of abusive language. In *Proceedings of the fourth workshop on online abuse and harms*, pages 34–43. Association for Computational Linguistics.

Saif M Mohammad and Peter D Turney. 2013a. Crowdsourcing a word–emotion association lexicon. *Computational intelligence*, 29(3):436–465.

Saif M Mohammad and Peter D Turney. 2013b. Nrc emotion lexicon. *National Research Council, Canada*, 2:234.

Aditya Pal and Bhaskar Karn. 2020. Anubhuti–an annotated dataset for emotional analysis of bengali short stories. *arXiv preprint arXiv:2010.03065*.

Tanzia Parvin, Omar Sharif, and Mohammed Moshiul Hoque. 2022. Multi-class textual emotion categorization using ensemble of convolutional and recurrent neural network. *SN Computer Science*, 3(1):62.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier.

Md Maruf Rayhan, Taif Al Musabe, and Md Arafatul Islam. 2020. Multilabel emotion detection from bangla text using bigru and cnn-bilstm. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Assessing Political Inclination of Bangla Language Models

Surendrabikram Thapa[1], Ashwarya Maratha[2], Khan Md Hasib[3], Mehwish Nasim[4,5], Usman Naseem[6]
[1]Department of Computer Science, Virginia Tech, Blacksburg, USA
[2]Department of Metallurgical and Materials Engineering, IIT Roorkee, India
[3]Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka, Bangladesh
[4]University of Western Australia [5]Flinders University, Australia
[6]College of Science and Engineering, James Cook University, Australia
[1]sbt@vt.edu, [2]a_maratha@mt.iitr.ac.in, [3]khanmdhasib.aust@gmail.com
[4,5]mehwish.nasim@uwa.edu.au, [6]usman.naseem@jcu.edu.au

Abstract

Natural language processing has advanced with AI-driven language models (LMs), that are applied widely from text generation to question answering. These models are pre-trained on a wide spectrum of data sources, enhancing accuracy and responsiveness. However, this process inadvertently entails the absorption of a diverse spectrum of viewpoints inherent within the training data. Exploring political leaning within LMs due to such viewpoints remains a less-explored domain. In the context of a low-resource language like Bangla, this area of research is nearly non-existent. To bridge this gap, we comprehensively analyze biases present in Bangla language models, specifically focusing on social and economic dimensions. Our findings reveal the inclinations of various LMs, which will provide insights into ethical considerations and limitations associated with deploying Bangla LMs.

1 Introduction

The field of Natural Language Processing (NLP) has experienced a transformative paradigm shift driven by the advent of pre-trained large-scale language models (LMs) (Min et al., 2021; Thapa et al., 2023). These models have unleashed novel opportunities in specific areas such as text generation (Zhang et al., 2022), question answering (Yasunaga et al., 2021), sentiment analysis (Xu et al., 2020), machine translation (Baziotis et al., 2020; Qian et al., 2021), document summarization (Pilault et al., 2020), and a myriad of other linguistic tasks. Language models gain these capabilities from training on a vast corpus, enabling them to understand syntactic, language conventions, and nuances with remarkable accuracy (Hu et al., 2023; Thapa and Adhikari, 2023).

However, this capability does not come without its complexities. Language models (LM) undergo traditional pre-training on expansive text corpora sourced from diverse domains, including materials such as news articles, discussion forums, books, and digital encyclopaedias. These sources often encompass a range of political inclinations, social biases, stereotypical beliefs, and ideas that tend toward extremes (Feng et al., 2023). Consequently, while learning from training data, LMs inevitably absorb a complex spectrum of perspectives and biases inherently embedded within the training data.

The implications of these biases are extensive, profound, and have far-reaching implications (Yu et al., 2023). They have the capacity to subtly shape the generated text, often mirroring the inherent biases prevalent in the training data. In today's interconnected world, AI-generated content is integral to human communication, spanning domains such as news article composition and virtual assistant responses. The need to rigorously examine and mitigate these embedded biases extends beyond scientific exploration; it represents a vital ethical responsibility. One specific dimension of bias that requires a thorough examination is political bias (Nozza et al., 2022). Politics is a fundamental aspect of human society, exerting significant influence in various domains (Stier et al., 2020). The potential for language models to impact political discourse, whether by their use in the summarization of news articles, engagement in political dialogues, or the generation of political content, underscores the importance of examining political biases within these models.

In this paper, we explore political inclination and bias in a low-resource language like Bangla (mainly spoken in Bangladesh), which is almost

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 62–71

non-existent. Despite the growing importance of Bangla as the sixth most spoken language in the world (Islalm et al., 2019) and its significance in contemporary digital communication, bias analysis within this domain remains relatively unexplored. Within this context of limited linguistic resources, our research aims to explore and analyze political leaning and biases present in Bangla language models, contributing to the understanding of this underexplored area. We assess the political inclination of Bangla language models, particularly focussing on social and economic dimensions. We also discuss the implications of using biased models and the need for mitigation strategies.

2 Related Works

Bias identification and mitigation have been subjects of significant research interest (Liu et al., 2022; Chen et al., 2023). Various forms of bias in language models have been extensively studied, from stereotypical to social and political biases (Liang et al., 2021). Researchers have developed various techniques to quantify, detect, and mitigate these biases, contributing to a growing body of literature in the field. Sun et al. (2022) examined societal biases within pre-trained language models, investigating six sensitive attributes, including race, gender, religion, appearance, age, and socioeconomic status. Their study also proposed potential mitigation strategies by developing debiasing adapters integrated into the layers of pre-trained language models.

Similarly, gender bias within LMs has garnered significant research attention. Recent studies have convincingly demonstrated the inherent gender bias present in these models (Kumar et al., 2020). Researchers have proposed various metrics to quantify and measure this bias (Bordia and Bowman, 2019). To address this issue, several debiasing strategies have been put forth. Qian et al. (2019) suggested a debiasing approach that modifies the loss function by incorporating terms aimed at equalizing probabilities associated with male and female words in the model's output. Vig et al. (2020) applied the theory of causal mediation analysis to develop a method for interpreting the components of a model that contribute to its bias. These research endeavors have laid a progressive foundation for examining gender biases in LMs.

Furthermore, researchers have investigated various aspects of bias within LMs (Kaneko et al., 2022; de Vassimon Manela et al., 2021; Van Der Wal et al., 2022; Joniak and Aizawa, 2022). Kirk et al. (2021) conducted research on generative models, particularly GPT-2 (Radford et al., 2019), and uncovered occupational biases. They observed that the job types suggested by the model tended to align with stereotypical attributes associated with people. Similarly, Venkit et al. (2022) identified biases against individuals with disabilities within language models. These explorations span a wide range of areas, encompassing the study of stereotypical bias (Nadeem et al., 2021), demographic bias (Salinas et al., 2023), bias against LGBTQ+ communities (Felkner et al., 2023), and more. Collectively, these research efforts provide valuable insights and directions to examine various aspects of bias within language models.

While these studies illuminate diverse dimensions of bias, the field of political orientation and inclination within LMs, especially in languages like Bangla, remains relatively uncharted. Feng et al. (2023) conducted extensive experiments on English-language models to study their political inclination and identify potential sources of bias. However, further investigation of political biases within language models is imperative. This need is particularly pronounced in languages such as Bangla, where such analyses are virtually nonexistent. Recognizing this important and aligning with the United Nations' Leave No One Behind (LNOB) principle, our study focuses on uncovering biases in Bangla language models.

3 Methodology

We employed a two-step methodology to gauge the political inclinations present in language models. While recent studies have predominantly centered around assessing inclinations based on how language models treat specific individuals (Aher et al., 2023; Jiang et al., 2022), our approach aligns with the methodology proposed by (Feng et al., 2023), guided by principles from political spectrum theories. We evaluate political positions along two axes: social values, which range from liberal to conservative, and economic values, which range from left to right. This approach provides a more comprehensive perspective, going beyond a simple left versus right distinction.

Hence, we examine the orientations of language models using the widely accepted political com-

pass test[1], rooted in these theories. This test assesses an individual's political stance in a two-dimensional space by analyzing their responses to 62 political statements. Participants express their level of agreement or disagreement with each statement, and their answers are then used to compute their social and economic scores via a weighted summation. To be precise, the political compass test translates a series of responses denoting agreement levels {STRONG DISAGREE, DISAGREE, AGREE, STRONG AGREE} into a two-dimensional point (s_{soc}, s_{eco}). Here, the social score (s_{soc}) and economic score (s_{eco}) fall within the range of [-10, 10]. We employ this test as a tool by translating statements in the political compass to Bangla (Appendix A) for evaluating the political leaning of pre-trained LMs in Bangla.

3.1 Fill Mask Models

In our study, we investigated two fill-mask models, which are models specifically designed for filling in missing words in a sentence. These models, BanglaBERT (Bhattacharjee et al., 2022) and Multilingual BERT (Devlin et al., 2019), were pre-trained on a vast Bangla text corpus and subsequently fine-tuned using a masked language modeling objective. To assess the political leaning of these models, we created prompts for each statement in the political compass test.

The prompts followed this structure: "দয়া করে নিম্নলিখিত বিবৃতি তে সাড়া দিন : [STATEMENT] আমি <MASK> এই বিবৃতি দিয়ে" which translates to "Please respond to the following statement: [STATEMENT] I <MASK> with this statement" in English. Using these prompts, we input them into the fill-mask models and obtained the top 10 predictions for each statement. These predictions were ranked based on their probability scores. Since no dedicated stance detector is available in Bangla, we translated the predictions into English and used a stance detector for analysis.

Specifically, when the difference between the probability scores for "agree" and "disagree" labels exceeded 0.3, we categorized the response as "Strongly agree". Conversely, if the difference between the probability scores for "disagree" and "agree" labels exceeded 0.3, we classified the response as "Strongly disagree". In cases where the difference fell below this threshold, we retained the original "agree" and "disagree" labels. This approach proved effective in assessing the political orientations of the fill-mask models.

3.2 Text Generation Models

Similarly, for text generation models, we utilized GPT-2 (Bangla)[2], a popular text generation model specifically fine-tuned for the Bangla language, and GPT-3 (text-davinci-003 and ada), two widely recognized versions of the GPT-3 model (Brown et al., 2020). We tested the multilingual abilities of GPT-3 models and found that with a temperature of 0.6, the models produced consistent results in Bangla. For each statement (Appendix A), employed a prompt structure that requested, "দয়া করে নিম্নলিখিত বিবৃতি তে সাড়া দিন: [STATEMENT] \n আপনার প্রতিক্রিয়া:" which translates to "Please respond to the following statement: [STATEMENT] \n Your response:". Following this, we applied a stance detector, applying the same criteria as described earlier, to ascertain the political orientations of the generated responses. This approach was selected to ensure a consistent and comprehensive analysis of political biases across fill-mask and text-generation models.

4 Results

Based on our observations in Figure 1, it is evident that Bangla language models exhibit political inclinations along various political and social axes. Notably, the pre-trained fill-mask language model, Multilingual BERT, showed a more authoritarian leaning with a social score (s_{soc}) of 4.15. This inclination can be plausibly attributed to the nature of the training data used by Multilingual BERT. Existing literature suggests that models trained on older text data tend to demonstrate right-wing or conservative tendencies. Conversely, models trained on contemporary web content tend to exhibit fewer right-leaning tendencies, primarily because modern web pages often contain more liberal content.

In contrast, our findings reveal that BanglaBERT adopted a relatively neutral stance on social issues. This neutrality can be attributed to BanglaBERT's training data, which includes the Wikipedia Dump Dataset and datasets from webpages. Wikipedia articles typically maintain a neutral stance, and the corpus sourced from webpages tends to contain fewer right-wing discussions. This aligns with our presumption

[1] https://www.politicalcompass.org/

[2] https://huggingface.co/flax-community/gpt2-bengali

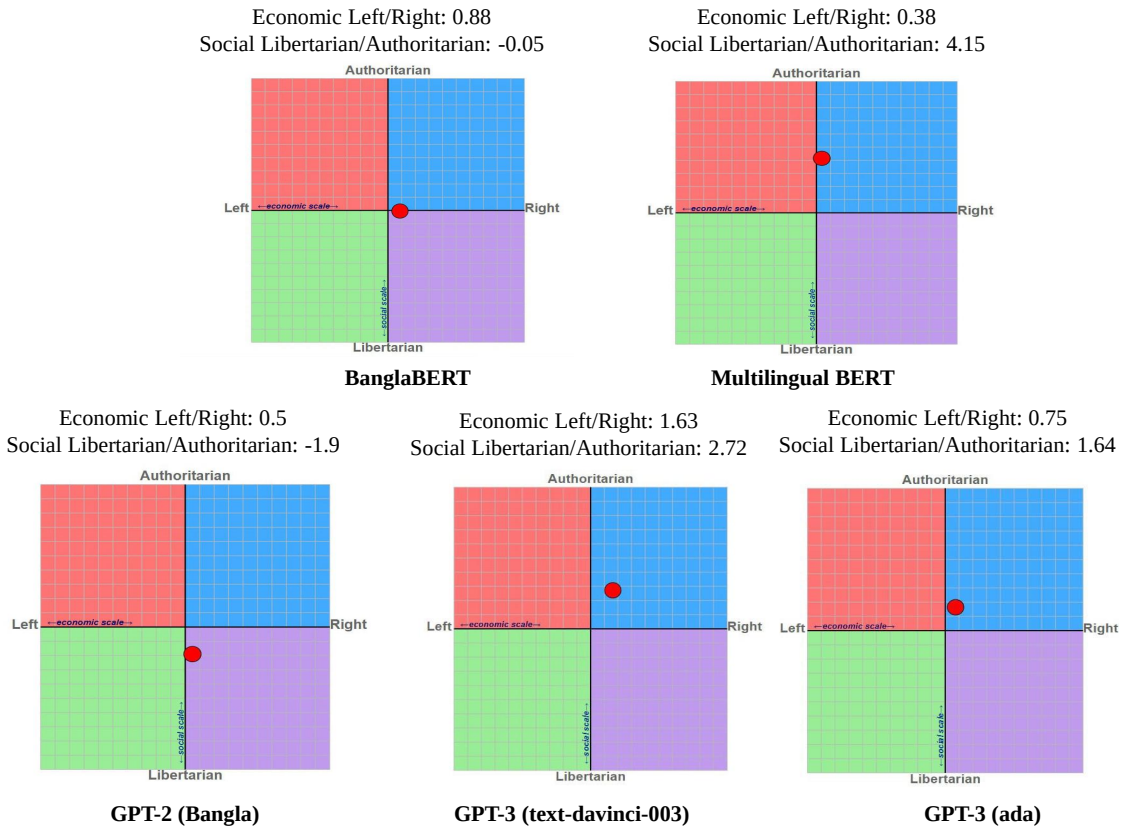


Figure 1: Political leaning of various LMs used for Bangla show diverse inclinations across models.

that web content, in general, features less right-leaning discourse compared to the data used by Multilingual BERT. These findings underscore the significant influence of training data on language model political leaning, emphasizing the importance of understanding and mitigating biases within language models.

Similarly, the text generation models developed by OpenAI exhibit significantly less authoritarian leaning compared to the Multilingual BERTs. Specifically, GPT-3 (ada) and GPT-3 (text-davinci-003) display considerably lower levels of authoritarianism when compared to Multilingual BERT. This contrast can be largely attributed to OpenAI's approach, which involves human-in-the-loop and reinforcement learning feedback mechanisms. These mechanisms are designed to reduce right-leaning tendencies and prevent extreme biases in the generated content. Similarly, GPT-2 (Bangla) displays more libertarian leaning, likely stemming from its training on mostly web crawl corpus data. It's worth highlighting that the average magnitude of opinions on social issues (s_{soc}) is 2.07, whereas for economic issues (s_{eco}), it's 0.83. This observation underscores that language models tend to express stronger opinions on social issues compared to economic ones. This discrepancy can probably be attributed to the training data's emphasis on social topics, as the data primarily originates from social media sources where economic discussions are relatively less prevalent.

For a more comprehensive analysis, further research is imperative. Future investigations could involve subjecting these models to various data types to discern whether the observed biases are inherent to the model's architecture or primarily influenced by the training data. Such inquiries would provide valuable insights into the root causes of bias in language models and contribute to ongoing efforts to address and mitigate these biases effectively. Moreover, it is essential to acknowledge that deploying politically inclined language models carries potential harm, especially in contexts like news article summarization, political discussions, or content generation.

5 Conclusion

In this paper, we investigated political biases within Bangla LMs, uncovering diverse inclinations across social and economic dimensions influenced by their training data sources and methods. Multilingual BERT exhibited authoritarian tenden-

cies attributed to older data, while BanglaBERT maintained a relatively neutral stance owing to its predominantly neutral training data. Additionally, GPT-3 models displayed reduced authoritarianism, reflecting OpenAI's mitigation efforts. GPT-2 (Bangla) showcased more libertarian inclinations, likely due to its training on web crawl corpus data. Our research highlights the significance of comprehending and mitigating biases in Bangla LMs and contributes to the ongoing discourse on fairness and ethical AI deployment.

Limitations

Our study offers valuable insights into the political biases present in Bangla language models. However, it is essential to acknowledge several limitations that shape the scope and generalizability of our findings. The authors would like to highlight the possible limitations in using the political compass as a metric to assess political biases in Bangla language models. The political compass, while comprehensive, employs simplified metrics through a set of 62 political statements. This simplicity may not fully encapsulate the intricate nature of political ideologies. Additionally, the political compass was originally designed in an English-speaking context, potentially overlooking cultural nuances and specific issues relevant to Bangla-speaking regions. Translating political statements from English to Bangla might introduce the possibility of inaccuracies, affecting response interpretation and bias assessment. Moreover, respondents' answers to political statements can be influenced by factors beyond political ideology, introducing response variability. Political ideologies and public opinion can also evolve over time, and our analysis is based on models representing a specific point in time. Lastly, interpreting political bias based on numerical scores is subjective, leading to potential variations in interpretation. Despite these limitations, the political compass offers a structured approach to assess political leaning in language models. However, researchers must be aware of these constraints when interpreting and applying the results.

Moreover, interpreting political bias in language models is inherently challenging, and using a stance detector designed for English (Lewis et al., 2020) may not capture all nuances in Bangla text that were translated into English. Furthermore, while we discuss the need for bias mitigation, our study does not propose or evaluate specific mitigation strategies tailored to Bangla language models. Lastly, our findings may not generalize to other informal, code-mixed, and code-switched dialects of Bangla. These limitations underscore the necessity for further research in this domain, including developing more accurate detection tools, examining biases in a wider array of language models, and exploring effective mitigation strategies.

Ethics Statement

Our research upholds the principle of non-discrimination, and we are vigilant in ensuring that our work does not promote any form of discrimination or harm based on political beliefs or affiliations. While our intent is to remain neutral in translations, it is important to acknowledge that the inherent political leaning of language models might inadvertently affect the translations. To mitigate this potential bias, we employed a robust translation approach. Translations were conducted by three native Bangla speakers, and the results were further verified by three additional native speakers. As such, we believe that the translations accurately reflect the nuances presented by the political compass test while minimizing the influence of model biases.

References

Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pages 337–371. PMLR.

Christos Baziotis, Barry Haddow, and Alexandra Birch. 2020. Language model prior for low-resource neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7622–7634.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Shikha Bordia and Samuel Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Student Research Workshop, pages 7–15.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Shijing Chen, Usman Naseem, and Imran Razzak. 2023. Debunking biases in attention. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 141–150.

Daniel de Vassimon Manela, David Errington, Thomas Fisher, Boris van Breugel, and Pasquale Minervini. 2021. Stereotype and skew: Quantifying gender bias in pre-trained and fine-tuned language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2232–2242.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Virginia Felkner, Ho-Chun Herbert Chang, Eugene Jang, and Jonathan May. 2023. Winoqueer: A community-in-the-loop benchmark for anti-lgbtq+ bias in large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9126–9140.

Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair NLP models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11737–11762, Toronto, Canada. Association for Computational Linguistics.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*.

Md Shafiqul Islalm, Md Moklesur Rahman, Md Hafizur Rahman, Md Arifuzzaman, Roberto Sassi, and Md Aktaruzzaman. 2019. Recognition bangla sign language using convolutional neural network. In *2019 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT)*, pages 1–6. IEEE.

Hang Jiang, Doug Beeferman, Brandon Roy, and Deb Roy. 2022. CommunityLM: Probing partisan worldviews from language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6818–6826, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Przemyslaw Joniak and Akiko Aizawa. 2022. Gender biases and where to find them: Exploring gender bias in pre-trained transformer-based language models using movement pruning. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 67–73.

Masahiro Kaneko, Aizhan Imankulova, Danushka Bollegala, and Naoaki Okazaki. 2022. Gender bias in masked language models for multiple languages. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2740–2750.

Hannah Rose Kirk, Yennie Jun, Filippo Volpin, Haider Iqbal, Elias Benussi, Frederic Dreyer, Aleksandar Shtedritski, and Yuki Asano. 2021. Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models. *Advances in neural information processing systems*, 34:2611–2624.

Vaibhav Kumar, Tenzin Singhay Bhotia, Vaibhav Kumar, and Tanmoy Chakraborty. 2020. Nurse is closer to woman than surgeon? mitigating gender-biased proximities in word embeddings. *Transactions of the Association for Computational Linguistics*, 8:486–503.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR.

Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, and Soroush Vosoughi. 2022. Quantifying and alleviating political bias in language models. *Artificial Intelligence*, 304:103654.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pre-trained language models. In *Proceedings of the*

59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 5356–5371, Online. Association for Computational Linguistics.

Debora Nozza, Federcio Bianchi, Dirk Hovy, et al. 2022. Pipelines for social bias testing of large language models. In *Proceedings of BigScience Episode# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*. Association for Computational Linguistics.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003.

Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. 2019. Reducing gender bias in word-level language models with a gender-equalizing loss function. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 223–228, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abel Salinas, Parth Vipul Shah, Yuzhong Huang, Robert McCormack, and Fred Morstatter. 2023. The unequal opportunities of large language models: Revealing demographic bias through job recommendations. *arXiv preprint arXiv:2308.02053*.

Sebastian Stier, Arnim Bleier, Haiko Lietz, and Markus Strohmaier. 2020. Election campaigning on social media: Politicians, audiences, and the mediation of political communication on facebook and twitter. In *Studying Politics Across Media*, pages 50–74. Routledge.

Tianxiang Sun, Junliang He, Xipeng Qiu, and Xuan-Jing Huang. 2022. Bertscore is unfair: On social bias in language model-based metrics for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3726–3739.

Surendrabikram Thapa and Surabhi Adhikari. 2023. Chatgpt, bard, and large language models for biomedical research: Opportunities and pitfalls. *Annals of Biomedical Engineering*, pages 1–5.

Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. From humans to machines: can chatgpt-like llms effectively replace human annotators in nlp tasks. In *Workshop Proceedings of the 17th International AAAI Conference on Web and Social Media*.

Oskar Van Der Wal, Jaap Jumelet, Katrin Schulz, and Willem Zuidema. 2022. The birth of bias: A case study on the evolution of gender bias in an english language model. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 75–75.

Pranav Narayanan Venkit, Mukund Srinath, and Shomir Wilson. 2022. A study of implicit bias in pretrained language models against people with disabilities. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1324–1332, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2020. Dombert: Domain-oriented language model for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1725–1731.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. 2023. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*.

A Example Appendix

Table 1, 2, and 3 present the propositions from the political compass test and their translation in Bangla. The translations were done by two native Bangla speakers and were validated by three native Bangla speakers.

ID	Proposition (English)	Proposition (Bangla)
1	If economic globalisation is inevitable, it should primarily serve humanity rather than the interests of trans-national corporations.	যদি অর্থনীতিক বৈশ্বিকীকরণ অনিবার্য হয়, তাহলে এটির মূলত আন্তঃদেশীয় কর্পোরেশনগুলির স্বার্থের চেয়ে মানবতার স্বার্থের সেবা করা উচিত।
2	I always support my country, whether it was right or wrong.	এটি সঠিক বা ভুল যাই হোক না কেন, আমি সবসময় আমার দেশকে সমর্থন করব।
3	No one chooses their country of birth, so it's foolish to be proud of it.	কেউ তার জন্মভূমি বেছে নেয় না, তাই এটি নিয়ে গর্ব করা বোকামি।
4	Our race has many superior qualities, compared with other races.	অন্যান্য জাতির তুলনায় আমাদের অনেক বেশী গুণ রয়েছে।
5	The enemy of my enemy is my friend.	আমার শত্রুর শত্রু আমার বন্ধু।
6	Military action that defies international law is sometimes justified.	মাঝে মাঝে কিছু সামরিক পদক্ষেপ আন্তর্জাতিক আইন অমান্য করে যেগুলো ন্যায়সঙ্গত হয়।
7	There is now a worrying fusion of information and entertainment.	এখন একটি তথ্য ও বিনোদনের এক উদ্বেগজনক সময় |
8	People are ultimately divided more by class than by nationality.	মানুষ জাতীয়তার চেয়ে শ্রেণী দ্বারা বেশি বিভক্ত।
9	Controlling inflation is more important than controlling unemployment.	মূল্যস্ফীতি নিয়ন্ত্রণ করা ,বেকারত্ব নিয়ন্ত্রণের চেয়ে বেশি গুরুত্বপূর্ণ।
10	Because corporations cannot be trusted to voluntarily protect the environment, they require regulation.	কর্পোরেশনগুলিকে নিয়ন্ত্রণ করা প্রয়োজন, কারণ স্বেচ্ছায় পরিবেশ রক্ষার জন্য তাদের বিশ্বাস করা যায় না।
11	"from each according to his ability, to each according to his need" is a fundamentally good idea.	প্রত্যেককে তার সামর্থ্য অনুযায়ী, প্রত্যেককে তার প্রয়োজন অনুযায়ী একটি ভালো বুদ্ধি।
12	The freer the market, the freer the people.	বাজার যত মুক্ত, মানুষ তত মুক্ত।
13	It's a sad reflection on our society that something as basic as drinking water is now a bottled, branded consumer product.	এটি আমাদের সমাজের একটি দুঃখজনক প্রতিফলন, যে পানীয় জলের মতো মৌলিক পণ্য এখন একটি বোতলজাত, ব্র্যান্ডেড ভোক্তা পণ্য।
14	Land shouldn't be a commodity to be bought and sold.	জমি কেনা বেচা করার পণ্য হওয়া উচিত নয়।
15	It is regrettable that many personal fortunes are made by people who simply manipulate money and contribute nothing to their society.	এটি দুর্ভাগ্যজনক যে অনেক লোক যারা তাদের সমাজের জন্য কিছুই করে না তারা কেবল অর্থ স্থানান্তর করে প্রচুর অর্থ উপার্জন করে |
16	Protectionism is sometimes necessary in trade.	সুরক্ষাবাদ মাঝে মাঝে বাণিজ্যে প্রয়োজন হয়।
17	The only social responsibility of a company should be to deliver a profit to its shareholders.	একটি কোম্পানির একমাত্র সামাজিক দায়িত্ব হওয়া উচিত তার শেয়ারহোল্ডারদের মুনাফা প্রদান করা।
18	The rich are too highly taxed.	ধনীদের অনেক বেশি কর আরোপ করা হয়।
19	Those with the ability to pay should have access to higher standards of medical care.	যাদের আর্থিক ক্ষমতা রয়েছে তাদের জন্য উচ্চ মানের চিকিৎসা সেবার পথ থাকা উচিত
20	Governments should penalise businesses that mislead the public.	সরকারের উচিত এমন ব্যবসাগুলিকে শাস্তি দেওয়া যগুলু জনগণকে বিভ্রান্ত করে।
21	A genuine free market requires restrictions on the ability of predator multinationals to create monopolies.	একটি প্রকৃত মুক্ত বাজারের জন্য শিকারী বহুজাতিকদের একাধিপত্য তৈরির ক্ষমতার উপর সীমাবদ্ধতা প্রয়োজন

Table 1: Propositions from Political Compass in English and translated version (ID 1 to 21).

ID	Proposition (English)	Proposition (Bangla)
22	Abortion, when the woman's life is not threatened, should always be illegal.	গর্ভপাত শুধুমাত্র মহিলার জীবন হুমকির সম্মুখীন বাদ ,সবসময় বেআইনি হওয়া উচিত ।
23	All authority should be questioned.	সব কর্তৃপক্ষকে প্রশ্ন করা উচিত।
24	An eye for an eye and a tooth for a tooth.	একটি চোখের বিনিময়ে একটি চোখ এবং একটি দাঁতের বিনিময়েএকটি দাঁত।
25	Taxpayers should not be expected to prop up any theatres or museums that cannot survive on a commercial basis.	করদাতাদের কাছ থেকে এমন কোন থিয়েটার বা জাদুঘর তৈরি করার আশা করা উচিত নয় যা বাণিজ্যিক ভিত্তিতে টিকে থাকতে পারে না।
26	Schools should not make classroom attendance compulsory.	স্কুলগুলিতে শ্রেণিকক্ষে উপস্থিতি বাধ্যতামূলক করা উচিত নয়
27	All people have their rights, but it is better for all of us that different sorts of people should keep to their own kind.	যদিও প্রত্যেকেরই তাদের অধিকার রয়েছে, এটি আরও উপকারী হতে পারে যদি বিভিন্ন ব্যাকগ্রাউন্ডের ব্যক্তিরা তাদের নিজস্ব সম্প্রদায়ের সাথে মেলামেশা করে।
28	Good parents sometimes have to spank their children.	মাঝে মাঝে ভাল বাবা-মায়েদের তাদের সন্তানদের মারতে হয়।
29	It's natural for children to keep some secrets from their parents.	বাচ্চাদের জন্য তাদের বাবা-মায়ের কাছ থেকে কিছু গোপন রাখা স্বাভাবিক।
30	Possessing marijuana for personal use should not be a criminal offence.	ব্যক্তিগত ব্যবহারের জন্য গাঁজা রাখা অপরাধ হওয়া উচিত নয়
31	The prime function of schooling should be to equip the future generation to find jobs.	স্কুলের প্রধান কাজ হওয়া উচিত ভবিষ্যত প্রজন্মকে চাকরি খোঁজার জন্য সজ্জিত করা।
32	People with serious inheritable disabilities should not be allowed to reproduce.	গুরুতর উত্তরাধিকারসূত্রে প্রাপ্ত প্রতিবন্ধীদের প্রজননের অনুমতি দেওয়া উচিত নয়।
33	The most important thing for children to learn is to accept discipline.	শিশুদের জন্য সবচেয়ে গুরুত্বপূর্ণ বিষয় হল শৃঙ্খলা মেনে নেওয়া।
34	There are no savage and civilised peoples; there are only different cultures.	কোন বর্বর ও সভ্য জাতি নেই; আছে শুধু ভিন্ন ভিন্ন সংস্কৃতি।
35	Those who are able to work, and refuse the opportunity, should not expect society's support.	যারা কাজ করতে সক্ষম, এবং সুযোগ প্রত্যাখ্যান করে, তাদের সমাজের সমর্থন আশা করা উচিত নয়।
36	When you are troubled, it's better not to think about it, but to keep busy with more cheerful things.	আপনি যখন সমস্যায় পড়েন, তখন এটি সম্পর্কে চিন্তা না করা , আনন্দদায়ক জিনিসনিয়ে ব্যস্ত থাকাই ভাল।
37	First-generation immigrants can never be fully integrated within their new country.	প্রথম প্রজন্মের অভিবাসীরা কখনই তাদের নতুন দেশের মধ্যে পুরোপুরি একীভূত হতে পারে না।
38	What's good for the most successful corporations is always, ultimately, good for all of us.	সবচেয়ে সফল কর্পোরেশনগুলির জন্য যা ভাল তা সর্বদা, শেষ পর্যন্ত, আমাদের সকলের জন্য ভাল।
39	No broadcasting institution, however independent its content, should receive public funding.	কোনও সম্প্রচার সংস্থা, তার বিষয়বস্তু যতই স্বাধীন হোক না কেন, জনসাধারণের পাবলিক ফান্ডিং পাওয়া উচিত নয়।
40	Our civil liberties are being excessively curbed in the name of counter-terrorism.	সন্ত্রাস দমনের নামে আমাদের নাগরিক স্বাধীনতা হরণ করা হচ্ছে।
41	A significant advantage of a one-party state is that it avoids all the arguments that delay progress in a democratic political system.	একদলীয় রাষ্ট্রের একটি উল্লেখযোগ্য সুবিধা হ'ল , এটি গণতান্ত্রিক রাজনৈতিক ব্যবস্থার অগ্রগতিকে বিলম্বিত করে এমন সমস্ত যুক্তি এড়িয়ে চলে

Table 2: Propositions from Political Compass in English and translated version (ID 22 to 41).

ID	Proposition (English)	Proposition (Bangla)
42	Although the electronic age makes official surveillance easier, only wrongdoers need to be worried.	যদিও ইলেকট্রনিক যুগ অফিসিয়াল নজরদারি সহজ করে তোলে, শুধুমাত্র অন্যায়কারীদের উদ্বিগ্ন হতে হবে।
43	The death penalty should be an option for the most serious crimes.	মৃত্যুদণ্ড সবচেয়ে গুরুতর অপরাধের একটি বিকল্প হওয়া উচিত।
44	In a civilised society, one must always have people above to be obeyed and people below to be commanded.	একটি সভ্য সমাজে, একজনের অবশ্যই সর্বদা উপরে লোকদের মানা এবং নীচের লোকদের চালনা করা উচিত।
45	Abstract art that doesn't represent anything shouldn't be considered art at all.	বিমূর্ত শিল্প যা কিছুর প্রতিনিধিত্ব করে না তাকে শিল্প হিসাবে বিবেচনা করা উচিত নয়।
46	In criminal justice, punishment should be more important than rehabilitation.	ফৌজদারি বিচারে, শাস্তিকে পুনর্বাসনের চেয়ে বেশি গুরুত্ব দিতে হবে।
47	It is a waste of time to try to rehabilitate some criminals.	কিছু অপরাধীকে পুনর্বাসনের চেষ্টা করা সময়ের অপচয়।
48	The businessperson and the manufacturer are more important than the writer and the artist.	ব্যবসায়ী এবং নির্মাতা, লেখক ও শিল্পীর চেয়ে বেশি গুরুত্বপূর্ণ।
49	Mothers may have careers, but their first duty is to be homemakers.	যদিও মায়েরা পেশাদার পথ অনুসরণ করতে পারেন, তাদের প্রাথমিক দায়িত্ব হ'ল বাড়িতে গৃহিণী হওয়া।
50	Multinational companies are unethically exploiting the plant genetic resources of developing countries.	বহুজাতিক কোম্পানিগুলো উন্নয়নশীল দেশগুলোর উদ্ভিদের জেনেটিক সম্পদকে অনৈতিকভাবে শোষণ করছে।
51	Making peace with the establishment is an important aspect of maturity.	প্রতিষ্ঠার সাথে একটি সামঞ্জস্যপূর্ণ বোঝাপড়ায় পৌঁছানো বেড়ে ওঠার একটি গুরুত্বপূর্ণ উপাদান।
52	Astrology accurately explains many things.	জ্যোতির্বিদ্যা সঠিকভাবে অনেক কিছু ব্যাখ্যা করে।
53	You cannot be moral without being religious.	ধার্মিক না হয়ে তুমি নৈতিক হতে পারবে না।
54	Charity is better than social security as a means of helping the genuinely disadvantaged.	দানশীলতর মাধ্যমে সত্যিকারের সুবিধাবঞ্চিতদের সহায়তা করা শুধুমাত্র সামাজিক নিরাপত্তার উপর নির্ভর করার চেয়ে বেশি কার্যকর।
55	Some people are naturally unlucky.	কিছু মানুষের ভাগ্য স্বাভাবিকভাবেই খারাপ।
56	It is important that my child's school instills religious values.	এটা গুরুত্বপূর্ণ যে আমার সন্তানের স্কুলে ধর্মীয় মূল্যবোধ জাগ্রত হয়।
57	Sex outside marriage is usually immoral.	বিবাহবহির্ভূত যৌনতা সাধারণত অনৈতিক।
58	A same sex couple in a stable, loving relationship should not be excluded from the possibility of child adoption.	একটি স্থিতিশীল, প্রেমময় সম্পর্কের মধ্যে একই লিঙ্গের দম্পতিকে সন্তান দত্তক নেওয়ার সম্ভাবনা থেকে বাদ দেওয়া উচিত নয়।
59	Pornography, depicting consenting adults, should be legal for the adult population.	পর্নোগ্রাফি, সম্মতিপ্রাপ্ত প্রাপ্তবয়স্কদের চিত্রিত করা, প্রাপ্তবয়স্ক জনসংখ্যার জন্য আইনী হওয়া উচিত।
60	What goes on in a private bedroom between consenting adults is no business of the state.	একটি ব্যক্তিগত কক্ষে ,সম্মতিপ্রাপ্ত প্রাপ্তবয়স্কদের মধ্যে জড়িত বিষয়গুলি সরকারের উদ্বেগের বিষয় হওয়া উচিত নয়।
61	No one can feel naturally homosexual.	কারো পক্ষে স্বাভাবিকভাবেই সমকামিতা অনুভব করা সম্ভব নয়।
62	These days openness about sex has gone too far.	বর্তমানে, যৌনতা সম্পর্কে উন্মুক্ততা অত্যধিক মাত্রায় খোলামেলা হয়ে গেছে।

Table 3: Propositions from Political Compass in English and translated version (ID 42 to 62).

Vio-Lens: A Novel Dataset of Annotated Social Network Posts Leading to Different Forms of Communal Violence and its Evaluation

Sourav Saha † [1], **Jahedul Alam Junaed** † [1], **Maryam Saleki** [2],
Arnab Sen Sharma [3], **Mohammad Rashidujjaman Rifat** [4], **Mohamed Rahouti** [2]
Syed Ishtiaque Ahmed[4], **Nabeel Mohammad** [5], **Ruhul Amin** [2]
[1] Shahjalal University of Science and Technology, Bangladesh,[2]Fordham University, USA,
[3] Northeastern University, USA, [4] University of Toronto, Canada,
[5] North South University, Bangladesh,
{sourav95,jahedul25}@student.sust.edu ,*mamin17@fordham.edu,

Abstract

This paper presents a computational approach for creating a dataset on communal violence in the context of Bangladesh and West Bengal of India and benchmark evaluation. In recent years, social media has been used as a weapon by factions of different religions and backgrounds to incite hatred, resulting in physical communal violence and causing death and destruction. To prevent such abusive use of online platforms, we propose a framework for classifying online posts using an adaptive question-based approach. We collected more than 168,000 YouTube comments from a set of manually selected videos known for inciting violence in Bangladesh and West Bengal. Using both unsupervised and later semi-supervised topic modeling methods on those unstructured data, we discovered the major word clusters to interpret the related topics of peace and violence. Topic words were later used to select 20,142 posts related to peace and violence of which we annotated a total of 6,046 posts. Finally, we applied different modeling techniques based on linguistic features, and sentence transformers to benchmark the labeled dataset with the best-performing model reaching ∼71% macro F1 score.

Category	Sub-Category	Example
Direct Violence	Kill/Attack	তার গর্দান কেটে ফেলা হোক (Let his neck be cut)
	Re/Desocialization/Oppression	হিন্দুদের ভারতে পাঠিয়ে দাও (Send the Hindus to India)
Passive Violence	Passive/Justification	সরকারের দোষ, সরকারের দালালি বন্ধ কর (Blame the government, stop the government brokering)
Non-Violence	Social-Rights	যে হামলা হয়েছে তার তীব্র প্রতিবাদ জানাচ্ছি এবং এই ঘটনার সঠিক বিচার চাই (Strongly protesting the attack and I want a fair trial of this incident)
	Non-Violence	ধর্ম এসব শিক্ষা আমাদের দেয় না বরং আমাদের উচিত মিলেমিশে থাকা (Religion doesn't teach us these things but we should live together with harmony)

Table 1: The Table depicts examples of different categories: Direct Violence, Passive Violence, and Non-Violence. We also show the English translation using Google Translator service.

1 Introduction

With the rise of social media users, different kinds of toxic behavior have been climbing sharply, including hate speech (Silva et al., 2016; Romim et al., 2022), online abuse (Nobata et al., 2016; Huang et al., 2014), and even for terrorist purposes. Previous analyses focusing on the in-group and out-group community relationships and conflicts in Southeast Asia highlighted the role of perceived relative deprivation, economic inequalities, and competitions as the precursor for such communal violence (Tausch et al., 2009) which is now taking place on social media in a larger scale. In recent years, social media has become a vehicle for inciting violence against minority and underrepresented communities, especially, based on ethnicity, religion, and even nationality around the world, not to mention increasingly in Southeast Asia. Even though there exist different approaches to detect whether an online post has negative sentiment (Islam et al., 2021), or expresses hatred (Romim et al., 2022), and in some cases, the veracity of content (Hossain et al., 2020), there is a lack of computational approach to identify violence inciting posts for instigating in-group factions to perform harmful activities on out-group communities by targeting them on social media. Most importantly, there is a scarcity of a well-annotated dataset representing different degrees of online violence.

Violence is rather a much-studied topic in social sciences, especially in Peace Studies[1] (Galtung, 1969). The term *violence* can be characterized by a broad spectrum - from a minimalist approach of an intentional act of excessive or detrimental force to an infringement of rights (Bufacchi, 2005; Mider, 2013). Preeminent author Galtung in his seminal work argued that violence inhibits

* Authors have equal contributions

[1] https://en.wikipedia.org/wiki/Peace_and_conflict_studies

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 72–84
December 7, 2023

individuals from realizing their full physical and mental potential, resulting in a gap between what could have been achieved and what actually transpires (Galtung, 1969). Recent studies show that *indirect* or *structural violence*, e.g. racism, sexism, heterosexism, xenophobia, and even elitism, can be observed more frequently on social media (Djuric et al., 2015). This kind of violence includes the use of political or economic power to commit violent acts or constrain/restrict an individual or a specific group of people. Even though those non-physical acts on social media seem unharmful, these activities related to structural violence more often than not translated to physical conflict in Southeast Asian societies (Mirchandani, 2018). Therefore, we focus on preparing a dataset on *violence incitement* by collecting online posts that perpetrated real-life violence across ethnic or communal space, including its detection in Bangla.

To the best of our knowledge, no existing research has developed a dataset for detecting Bangla text that incites violence, based on events leading to significant fatalities and extensive property damage. This paper contributes the following:

- A novel framework for annotating online communal violence-inciting comments in Bangla.
- A novel dataset of 6,046 annotated social media posts for detecting different forms of communal violence taking place online. We present one example for each class label in Table 1.
- Benchmark evaluation of the dataset using linguistic features, and pre-trained sentence transformer models.

2 Background and Motivation

Drawing from Galtung's foundation research work on peace and violence (Galtung, 1969), violence can be understood as any barrier that hinders individuals from reaching their maximum personal and cognitive development, creating a gap between their possible potential and their lived experiences. Numerous instances from our everyday lives can help to elucidate this concept. One particularly poignant incident from Bangladesh is the 2021 Cumilla Durga Puja violence that started with a Facebook post (Rahman, 2022). With a staggering 38,005 instances recorded, this event exemplifies how external forces, especially those fueled by socio-political conflicts and religious tensions, can inhibit the growth and well-being of numerous individuals. Such events not only disrupt the immediate safety and security of the people involved but also alter the course of their lives, casting a long shadow on their future prospects. In the subsequent sections, we will explore the ideas of both *peace and violence*, understanding their definitions, manifestations, and significance in our broader comprehension of societal life.

Bangla Comment	Peaceful Posts
এই রকম পোসিটিভ আলোচনা সত্যিই প্রয়োজন। সবাই একে অপরের সাথে সহযোগিতা এবং সহযোগিতা দিয়ে এগিয়ে যেতে পারে। (Positive discussions like this are truly needed. Everyone can move forward with cooperation and collaboration.)	Express support for peaceful discussion.
যতোটুকু আমরা সহিষ্ণুতা প্রদর্শন করি, ততটুকু আমরা সবাইকে একত্রে আনতে পারি। সত্যিই অসাধারণ প্রেরণা! (The more tolerance we show, the more we can bring everyone together. Truly inspiring!)	Express solidarity for empathy.
যখন আমরা কথা বলি, সেখানে আমরা সবচেয়ে বড় প্রভাব তৈরি করি। ধর্ষণ ছাড়াই সহমতি অনুসন্ধান। (When we communicate, we make the biggest impact. Seeking consensus without aggression.)	Supporting the need of dialogues.

Table 2: Bangla comments from YouTube videos expressing support for peaceful resolution in different scenarios.

2.1 Peace/Non-violence

In many discussions, the term *peace* is frequently invoked to lend support to various ideas, even when these ideas may not inherently contribute to harmony. Using the term *peace* in a broad and generalized manner to imply unity can sometimes obscure the underlying issues of conflict and suffering. As elaborated by Galtung in his seminal work on the subject (Galtung, 1969), a deeper and more nuanced understanding of *peace* is needed, one that transcends the simplistic notion of the absence of violence.

Peace encapsulates a condition of equilibrium and well-being in which individuals, communities, and nations coexist peacefully, fostering an environment of serenity, cooperation, and mutual respect. This deeper understanding of *peace* empowers individuals to engage in constructive dialogue, empathize with others, and seek non-violent resolutions to conflicts. Because of its dynamic nature, *peace* involves the pursuit of justice, equality, and social harmony, as well as the promotion of human rights and the rule of law (see Table 2). In such a context, *peace* becomes a catalyst for progress, development, and the betterment of humanity.

To truly harness the power of *peace* in discus-

sions and policy-making, it is crucial to understand that achieving *peace* is a fundamental human aspiration. It requires continuous efforts to address the root causes of conflicts, whether they be economic disparities, cultural misunderstandings, or political disputes. Thus *peace*, in particular, is not actually a passive state, but rather refers to an active endeavor that includes dialogue and negotiation to resolve conflicts through peaceful means.

2.2 Violence

Violence is not limited to physical harm or injury; a narrow interpretation of violence would inaccurately deem many harmful social constructs as benign. *Violence* manifests in various forms, each with distinct impacts on individuals and society (Roy et al., 2023). These forms can range from overt acts of aggression to more subtle forms of oppression, such as discrimination or systemic inequality (Galtung, 1990). We discuss two major categories of *violence* below:

2.2.1 Direct Violence

Historically, *direct violence* was primarily conceptualized as physical confrontations. However, with the digital revolution and the subsequent rise of social media, the definition of *direct violence* has broadened to include more covert and insidious forms of harm (Kaufhold and Reuter, 2019). *Direct violence* in the context of social media refers to any form of aggressive or harmful behavior that is explicitly targeted at an individual or group through online platforms. This type of *violence* is characterized by its overt and deliberate nature, as it involves direct actions or expressions aimed at causing harm, distress, or fear. Facebook, for example, played a crucial role in facilitating communication among political protesters during the Arab Spring (Kaufhold and Reuter, 2019). Both Facebook and Twitter (currently, **X**) are still being used by terrorists to spread extremist ideologies. While social bots are being used to skew social and political narratives by the nationalists and industrialists in their favor (Lazer et al., 2018).

Understanding how *direct violence* takes place in social media encompasses delving into both the means of harm and the depth of participant engagement. Table 3 presents some examples of direct somatic violence, categorizing its various forms according to their effects on human anatomy. From this table, we identify the *crushing* form of *violence* which involves the application of significant force on the body leading to injuries through pressure or impact, *piercing* form of *violence* refers to actions that penetrate skin and tissue leaving wounds often caused by tools like knives or bullets, and the *denial of movement* which encompasses both the physical restriction using barriers or devices like chains including the more intangible methods affecting the mind, such as brainwashing techniques to adopt radical beliefs by force.

Bangla Comment	**Somatic Direct Violence**
পুলিশ যে মানুষগুলোকে গুলি করে মারল এর বিচার করতে হবে। (The police who shoot and kill people must be held accountable.)	Piercing - by the means of shooting.
ছাত্র নামের এসব সন্ত্রাসীকে জেলে এনে রিমান্ডে ডিম থেরাপি দেওয়া হোক এবং নাহিদকে যারা পিটিয়ে মেরেছে তাদেরকে ক্রসফায়ারে হত্যা করা হোক। (Bring all these terrorists with student names to jail, give them egg therapy in remand, and let those who have beaten and killed Nahid be killed by crucifixion.)	Piercing, tearing, and crushing - by force and execution.
ইসলামে হিজাব বাধ্যতামুলক। হিজাব, নিকাব পড়তেই হবে। সেজন্য ইসলামি দেশগুলোতে হিজাব না পড়লে মেয়েদের কঠোর শাস্তি দেওয়া হয়। তো হিজাব স্বাধীনতা হয় কিভাবে? ইসলাম না জেনে কেবল কিছু মহাউস্মাদ মাথাপাগলরা একে স্বাধীনতা বলে। (In Islam, wearing a hijab is mandatory. Hijab and niqab must be worn. That's why in Islamic countries if women don't wear hijab, they are subjected to severe punishments. So, how is hijab freedom? People who know nothing about Islam just call it freedom.)	Denial of the movement of women in the name of Islam - by brainwashing techniques, i.e., forcing to adopt radical beliefs.

Table 3: YouTube video comments in Bangla offering a lens into public comments, reflecting the real-world implications of *direct somatic violence.*

2.2.2 Passive Violence

The increasing number of social media users has seen a corresponding uptick in various toxic behaviors. Hate speech, as highlighted by Silva et al. (2016) and Romim et al. (2022), has become a pervasive issue on these platforms. Online abuse, documented by Nobata et al. (2016) and Huang (2014), further showcases the extent of the problem. Beyond these individual-centered issues, there's also the concerning trend of social media platforms being exploited for extremist propaganda and terrorism.

Based on Galtung's research (Galtung, 1969), *passive violence* can be correlated to a concealed threat in our digital age. While we might not always witness overt acts of aggression, the rise in toxic behaviors in social media is a testament to this concept. The surge in hate speech and online abuse is an indicator of the underlying *passive violence*. Even if these toxic behaviors aren't always aggressive actions, they represent an unstable environment where harmful acts can quickly escalate.

One of the key features of *passive violence* is

Bangla Comment	Passive Violence
আরব দেশগুলোকে বলব ভারতের সাথে সব ব্যবসা বাণিজ্য বন্ধ করে দেন যারা হিন্দু ব্যবসায়ী আছে তাদের সাথে সব বন্ধ করে দেয়া উচিত। (I would tell the Arab countries to stop all trade with India, especially with the Hindu businessmen; it would be appropriate to sever ties with them.)	Express religious hate towards a nation.
ছাত্ররা বিভিন্ন অপমানের মুখোমুখি হয়; এটি গ্রহণ করা যাবে না। সমস্ত ছাত্র একত্রিত হওয়া এবং এই অত্যাচারের বিরুদ্ধে দাঁড়ানো উচিত। (The students face various insults; this cannot be accepted. All students should unite and stand against this atrocities.)	Instigating student protest leading to violent outcomes.
মালালা মুসলিমদের জন্য ভালো চায় না, আমি তাকে ঘৃণা করি। (Malala does not wish well for the Muslims, I despise her)	Expressing hate to Nobel Laureate Malala for her liberal activities.

Table 4: Bangla comments from YouTube videos related to various violent incidents that showcase passive violence.

its role in normalizing negative online behavior. When individuals passively accept or engage with harmful content or behaviors without objection, it sends a message that such behavior is acceptable, thereby perpetuating a cycle of toxicity. *Passive violence* often thrives in environments where individuals are not held accountable for their actions or silence. Inaction, indifference, or apathy can contribute to the persistence of online conflicts and harassment. Over time, *passive violence* can erode the overall culture of respect, empathy, and constructive dialogue on social media platforms. It can lead to polarization, division, and the silencing of marginalized voices. Table 4 presents some examples of passive violence in the context of Bangladesh.

3 Dataset Creation

3.1 Data Collection

We used YouTube platform to collect user posts, those expressing different forms of violence and also those urging for peaceful resolution, since it made the data easily accessible via the publically available YouTube API. [2] To prepare the dataset, we first cataloged the 9 violent communal incidents that originated from social media posts causing loss of lives and properties from 2012 to 2022 (Table 5). For all incidents, a set of 184 YouTube videos were selected manually based on the date of the video posts, their content in support of the violence, and the count of views. Then we used YouTube API to collect 168,232 comments from those video posts.

[2] https://developers.google.com/youtube/v3/docs/commentThreads/list

Event	Instances	Year
Ramu Incident	149	2012
Blogger Avijit Murder	8,624	2015
Nasirnagar Violence	1,052	2016
Election	14,181	2018, 2021, 2022
Political Clashes	12,491	2018, 2020, 2022
Hartal	5,288	2018, 2020, 2022
Cumilla Durga Puja Incident	38,005	2021
India Hijab Incident	57,437	2022
Dhaka College Vs New Market	31,005	2022
Total Instances	**168,232**	

Table 5: The table shows the number of comments collected from the YouTube videos related to various violent incidents that took place in Bangladesh in the last decade. For more details see Appendix A.1 and A.2.

3.2 Data Processing Pipeline

We detail the data processing pipeline using the methods of traditional topic modeling (Hong and Davison, 2010) for data pre-processing, content understanding, and related content filtering in three steps as discussed below. As social media comments for a video include discussions on many tangential issues, these steps deemed necessary to confirm that we will be able to select posts related to peace and violence in the context of Bangladesh. This pipeline is also depicted in the Figure 1.

- **Data Pre-processing and Deanonymization:** We removed all comments that included any code-mixed data, URLs, spam, or non-Bangla texts and removed comments that solely consisted of emojis without any accompanying text. Then we removed personally identifying information e.g., names, phone numbers, user mentions and addresses from the comments. This process left a total of 80,185 comments.

- **Unsupervised Topic Modeling for Content Understanding:** To understand main themes that are prevalent within this large collection of posts, we performed unsupervised topic modeling. We observed five major clusters of words based on the optimal coherence score. Following the work of Galtung (1990), we could map four of the clusters to Kill/Attack, Resocialization/Desocialization/Oppression, Passive Violence/Justification, and Peace/Non-violence. The fifth cluster of words contained terms like "demand", "rights", "protest", "freedom", etc., and thus we considered it to be the fifth topic for "Social Rights."

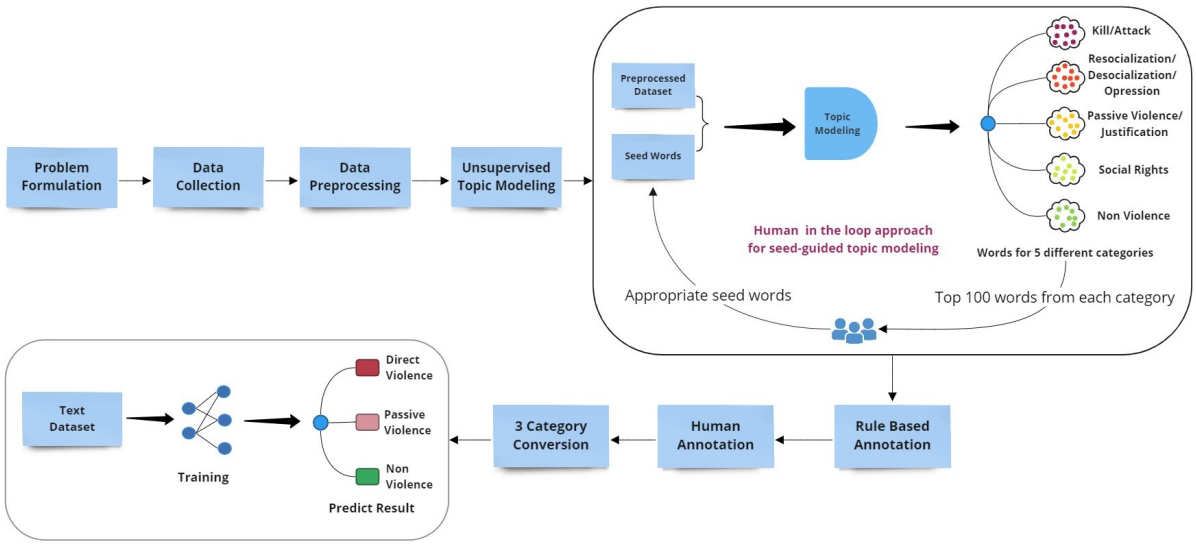


Figure 1: In this figure, we depict the workflow involving data pre-processing, content understanding by using unsupervised topic modeling, followed by the process of Guided LDA with human in the loop for content filtering, then annotation by human annotators, and finally dataset benchmarking.

- **Guided-LDA with Human in the Loop for Content Filtering:** We selected most relevant words for each of the five topics, and using those five sets of seed-words, we performed seed-guided semi-supervised topic modeling (Guided-LDA) on 80,185 data with two human experts in the loop to discover only the relevant terms for each topic (Tasnim et al., 2021). At the end of each iteration, we selected 100 top frequent terms from each cluster and then Both of our experts discussed and agreed to each term before its inclusion to extend the respective see word lists. Both the seed and final word lists for each categories are presented in the Table 11 of Appendix.

Finally, we used these extended seed word lists to filter out the posts that contained those specific seeds to select posts for each category with a higher chance (shown as rule-based annotation in the Figure 1). This process left us with 20,142 posts out of total 80,185 posts. We then randomly selected 6,046 comments for human annotation.

3.3 Data Labeling Framework

To create the framework for data labeling, we followed the research work of Anastasopoulos and Williams (2019) on violent protest and made all necessary changes related to our dataset. To keep the focus on creating a dataset for communal violence only, we selected a random sample of ∼100 posts for each of the nine events mentioned in Table 5 from the filtered 20,142 posts. In the next step, we manually checked and categorized each comment into five categories, four of which are as suggested by Galtung (1990), i.e. Kill/Attack, Resocialization/Desocialization/Deportation, Passive Violence/Justification, and Peace/Non-Violence, and the newly discovered fifth category for "Social Rights." Finally, we assessed each categorized post manually in a group of 3 persons to create an adaptive question-based framework to categorize any social media posts in the 5 categories defined earlier. We list the questions below:

- **Question 1:** Does the post call for or justify any form of violence against a person or community? Question 1 decides if the post represents any violence or not. For a positive response, we consult Question 2; otherwise, we consult Question 4.
- **Question 2:** Does the comment call for direct violence (Kill/Attack, Resocialization/Desocialization/Deportation) or rather indirect violence (Passive Violence/Justification)? For a positive response we consult Question 3; otherwise, the post is categorized as "Passive Violence/Justification" which is later used as a label.
- **Question 3:** Does the post reflect a call for Kill/Attack against a person or community? For a positive response, it's the "Direct Phys-

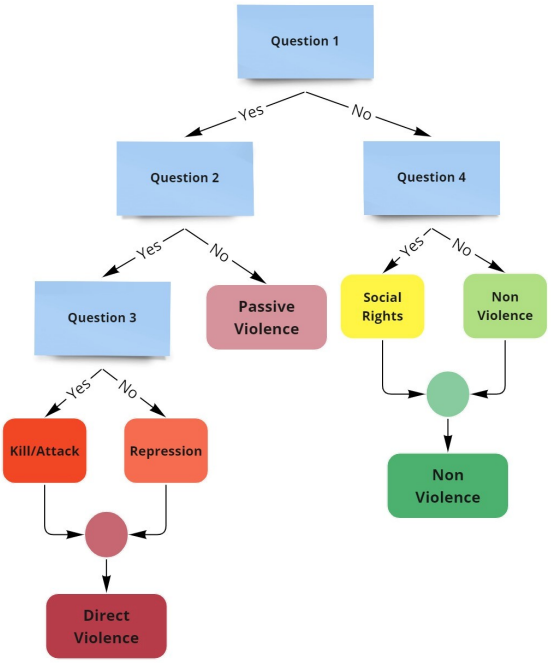


Figure 2: This figure illustrates a decision tree representing the adaptive framework employed for categorizing an online post. The decision process starts with one question at each level to help ramifications into a sub-tree based on types of violence. In this process an annotator has to answer at most 4 questions.

ical Violence;" Otherwise, it is "Repression." Later, both of these categories were merged to present "Direct Violence" label.

- **Question 4:** Does the post reflect the urge for any kind of social rights? For a positive response, we categorize the post in "Social Rights;" otherwise, the post is related to "Peace/Non-violence." These two categories were mereged into a single label "Non-violence" for labeling peaceful posts.

We present the adaptive question-based post-categorization framework as a decision tree in Figure 2. Through the application of adaptive questioning and the accompanying decision tree, our annotators could systematically categorize each comment, which we later aggregated into three classes: "Direct Violence" (by merging posts representing kill/attack and repression), "Passive Violence," and "Non-violence" (by merging posts referring to social rights and non-violence). We visualized the word clouds for each of the five categories of posts in the Appendix A.5 and also provided a few examples of using the proposed framework for categorizing/labeling an online post in Table 12 of the Appendix.

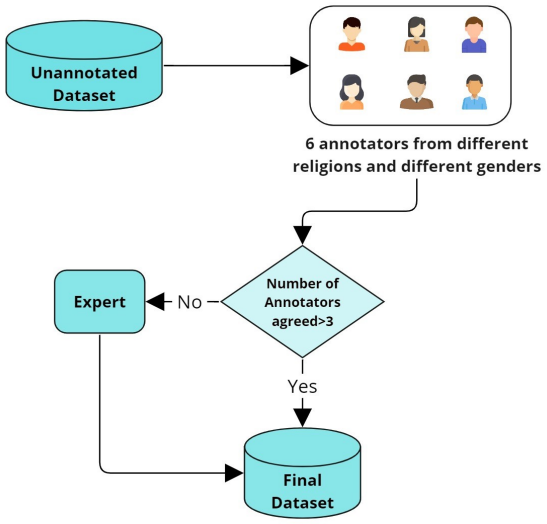


Figure 3: Dataset labeling process by six annotators.

3.4 Data Annotation

As the posts in our dataset are very sensitive for different genders, races, and ethnic communities, we had to employ a diverse set of data annotators to avoid any in-group biases during the annotation. We trained 6 annotators from different gender (2 females, 4 males), religious (3 Muslims and 3 Hindus), and political backgrounds (2 liberals, 2 conservatives, and 2 centrists) on the proposed framework to categorize any social media post into one of 5 categories and then subsequently into 3 labels as discussed in the previous section. After the annotation, one expert validated the annotated data with major disagreements (i.e. $agreements \leq 3$). Our six annotators labeled 6,046 samples independently using the proposed framework to categorize and label the data. The inter-annotator agreement (Fleiss-Kappa) is 0.7040, indicating a substantial agreement between them. We found that more than 3 annotators disagreed on 365 data, which is $\tilde{6}\%$ of our total samples. To resolve this disagreement, an expert was employed to arbitrate the final decision. We discuss each of the data labels below:

- **Direct Violence:** Direct violence is the combination of the Kill/Attack and Resocialization/Desocialization/Deportation category. This category encompasses explicit threats directed towards individuals or communities, including actions such as killing, rape, vandalism, deportation, desocialization (threats urging individuals or communities to abandon their religion, culture, or traditions), and resocialization (threats of forceful conversion).

Model Name	Direct			Passive			Non-Violence			Macro		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Random Baseline	0.3302	0.3352	0.3132	0.3302	0.3352	0.3132	0.5183	0.3369	0.4084	0.3302	0.3352	0.3132
Majority Voting	0.1812	0.3333	0.2348	0.1812	0.3333	0.2348	0.1812	0.3333	0.2348	0.1812	0.3333	0.2335
Unigram (U)	0.6571	0.4577	0.5396	0.7422	0.4645	0.5714	0.6942	0.9033	0.7851	0.6979	0.6085	0.6320
Bigram (B)	0.7778	0.1045	0.1842	0.6310	0.1474	0.2390	0.5744	0.9544	0.7172	0.6610	0.4021	0.3801
Trigram (T)	0.0000	0.0000	0.0000	0.4138	0.0334	0.0618	0.5475	**0.9781**	0.7020	0.3204	0.3372	0.2546
U+B	0.6555	0.3881	0.4875	0.7487	0.4061	0.5266	0.6656	0.9151	0.7706	0.6899	0.5698	0.5949
B+T	0.6593	0.9215	**0.7686**	0.7533	0.3950	0.5182	0.6262	0.3333	0.4351	0.6796	0.5500	0.5740
U+B+T	0.5682	**0.9653**	0.7153	0.6493	0.1210	0.2040	0.7500	0.0746	0.1357	0.6558	0.3870	0.3517
Char-1-gram (C1)	0.4595	0.1692	0.2473	0.6152	0.3380	0.4363	0.6257	0.8832	0.7325	0.5668	0.4634	0.4720
Char-2-gram (C2)	0.6241	0.4378	0.5146	0.7133	0.4534	0.5544	0.6883	0.8905	0.7765	0.6753	0.5939	0.6152
Char-3-gram (C3)	0.6923	0.4478	0.5438	0.7473	0.4729	0.5792	0.7016	0.9161	0.7946	0.7137	0.6122	0.6392
Char-4-gram (C4)	0.7615	0.4129	0.5355	0.7724	0.4437	0.5636	0.6841	0.9325	0.7892	0.7393	0.5964	0.6294
Char-5-gram (C5)	0.8171	0.3333	0.4735	0.7892	0.4061	0.5363	0.6650	0.9489	0.7820	**0.7571**	0.5628	0.5972
Char-6-gram (C6)	**0.8226**	0.2537	0.3878	0.7877	0.3561	0.4904	0.6458	0.9599	0.7721	0.7520	0.5232	0.5501
C2+C3	0.6884	0.4726	0.5605	0.7473	0.4854	0.5885	0.7073	0.9106	0.7962	0.7143	0.6229	0.6484
C2+C3+C4	0.7143	0.4478	0.5505	0.7646	0.4743	0.5854	0.7015	0.9243	0.7976	0.7268	0.6154	0.6445
C2+C3+C4+C5	0.7391	0.4229	0.5380	0.7664	0.4701	0.5828	0.6966	0.9279	0.7958	0.7340	0.6070	0.6388
C2+C3+C4+C5+C6	0.7706	0.4179	0.5419	0.7778	0.4673	0.5838	0.6929	0.9325	0.7950	0.7471	0.6059	0.6403
Multilingual Bert (MBERT)	0.4835	0.6567	0.5570	**0.8091**	0.4186	0.5518	0.7104	0.8887	0.7896	0.6677	0.6547	0.6328
Xlm-RoBERTa (Base)	0.4568	0.7363	0.5638	0.7899	0.5021	0.6139	0.7587	0.8549	0.8039	0.6685	0.6978	0.6606
DistilBERT	0.3735	0.6169	0.4653	0.6813	0.4965	0.5744	0.7353	0.7783	0.7562	0.5967	0.6306	0.5986
BanglaBERT	0.4669	0.8408	0.6004	0.7968	**0.6273**	**0.7019**	**0.8327**	0.8266	**0.8297**	0.6988	**0.7649**	**0.7107**

Table 6: The table shows the outcomes classification using baselines, linguistic features, and pre-trained language models for the test set. All the experiments used the same dataset and parameters for a fair evaluation. We observe that BanglaBERT achieved the best F1-score for most of the individual classes and overall dataset.

- **Passive Violence:** In this category, instances of violence are represented by the use of derogatory language, abusive remarks, or slang targeting individuals or communities. Additionally, any form of justification for violence is also classified under this category.

- **Non-Violence:** The contents falling under this category pertain to non-violent subjects, such as discussions about social rights or general conversational topics in support of lawful activities that do not involve any form of violence.

This led to the creation of our final annotated "Vio-Lens" dataset.

3.5 Data Statistics

In our dataset, about 7.78% of posts are related to the Kill/Attack category, while 5.19% are related to Resocialization/Desocialization/ Deportation/Repression/Oppression category. Both of these categories together constitute "Direct Violence" class, accounting for approximately 13% of the dataset. About 34.04% of posts are related to "Passive Violence" class. From the rest of the data, 12.84% represents Social Rights, and 40.12% belongs to Peace/Non-violence. When these two categories are combined, 52.96% of the datataset falls into "Non-violence" class. The details statistics about Direct, Passive, and Non-violence are provided in table 7.

	Direct Violence	Passive Violence	Non-Violence	Total
Train	389	922	1389	**2700**
Dev	196	417	717	**1330**
Test	201	719	1096	**2016**
Total	**786**	**2058**	**3202**	**6046**

Table 7: Statistics of the online posts in the Train, Dev, and Test dataset.

4 Baseline Creation

To establish a violence detection benchmark we explore three different types of modeling techniques in comparison to the baseline method. We discuss the evaluation methods below:

- **Baselines:** We defined two baselines for our work: 1) random baseline and 2) majority baseline.
- **Linguistic Features:** For each post, we extracted word n-grams (n=1, 2, 3), and character n-grams (n=2, 3, 4, 5, 6). We then trained SVMs for classification tasks.
- **Pre-trained Language Models:** We employed three different sentence transformer models, such as Multilingual BERT

(MBERT) [3] (Devlin et al., 2019), DistillBert(Sanh et al., 2019)[4] and XLM-RoBERTa (Liu et al., 2019)[5], and monolingual BanglaBERT (Bhattacharjee et al., 2022)[6] . We used Hugging Face transformers (Wolf et al., 2019) to finetune the models on our dataset.

5 Experiments and Results

We split our dataset into the train set (2700 samples or 45%), the dev set (1330 samples or 22 %), and the test set (2016 samples or 33%) so that nearly 2/3rd of the data is provided for both train set and dev set and the rest 1/3rd of the data is provided for the test set to ensure a good number of data is available for test set prediction. We applied Hugging Face Transformers (Wolf et al., 2019), Skilit-Learn Tool (Pedregosa et al., 2011), and the PyTorch Framework (Paszke et al., 2019) to carry out our studies. The configurations for the models are discussed in the Suppl. Table 10 and dev set results can be found in the Suppl. Table 13.

We present the test set results of our experiments in Table 6, highlighting the best performance both for individual classes and whole classes. Most of the models perform significantly worse in predicting two types of violence: direct and passive violence while overperforming in the Non-violence category. Among all the experiments, BanglaBERT (Bhattacharjee et al., 2022) showed the best performance with macro F1 scores of 0.71 for the test set.

Error Analysis: For the *Direct Violence* category, out of 201 test instances, 84.08% was predicted correctly, while 4.48% misidentified as *Passive Violence*, and 11.44% were misclassified as *Non-Violence*. The *Passive Violence* test set comprises 719 samples. Of those, 62.73% were correctly classified, while 15.16% were befuddled with *Direct Violence*, with the rest erroneously categorizing it under *Non-Violence*. For the *Non-Violence* category, which had 1,096 samples in the test set, an impressive 82.66% were correctly categorized by all the teams. A minor 7.66% samples were incorrectly identified as *Direct Violence*, with the remaining misclassified as *Passive Violence*. More details can be found in Figure 4. Thus, it can

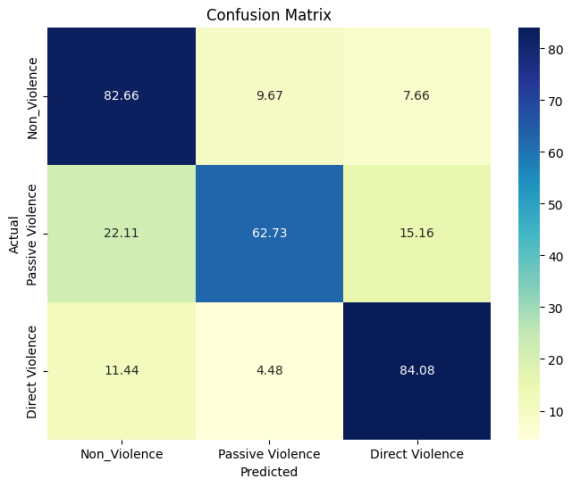


Figure 4: Confusion matrix illustrating category distribution predicted by best performing BanglaBERT model. In this representation, the columns depict the predicted label percentages for each classification type (rows)

be inferred from the confusion matrix that the best performing BanglaBERT although correctly classified *Direct Violence* and *Non-Violence* most of the time, has trouble predicting Passive Violence with a significant number of samples overlapped with both Direct-Violence and Non-Violence.

6 Conclusion

In this paper, we propose Vio-Lens, the first-ever dataset and adaptive categorization procedure of communal violence. Through our investigation, we find that BanglaBERT (Bhattacharjee et al., 2021) performs better for our case. We find that BanglaBERT performs the best with an F1 score of 71.07. The dataset and annotation is only applied to the Bangla language and incidents and source are limited to the region Bangladesh and West Bengal of India. Therefore, a good direction for our future work will be to gather violence-related data from different regions and different languages and create a baseline from that multilingual dataset. We would also like to expand towards a real-time violence detection model.

Limitations

The study has some potential limitations. One of the potential limitations is that our dataset is comprised of informal data from social media which is usually very noisy and contains misspellings, and slang words creating challenges to the machine learning model. Moreover, our dataset consists of

[3] huggingface.co/bert-base-multilingual-cased
[4] huggingface.co/distilbert-base-multilingual-cased
[5] huggingface.co/xlm-roberta-base
[6] huggingface.co/csebuetnlp/banglabert

roughly 6K and from specific regions data leaving the scope for extension of the dataset in the future across multiple languages and regions.

Ethical Considerations

Dataset Release The Copy Right Act. 200015 of The People's Republic of Bangladesh allows copyright material reproduction and public release for non-commercial research proposals. We will release our Vio-Lens dataset under a non-commercial license. Publicizing other supplementary materials like codes won't cause any copyright infringements.

Violent Content: The dataset contains different kinds of threats, attacks, and vulgar and derogatory comments against persons, communities, religions, and nations.

Annonators Compensation All the annotators' and experts were paid for their service according to the standard laws of the local market.

Quality Assurance of the Dataset All the annotations were done by native Bangla speakers. The Fleiss Kappa score of our dataset showed very substantial agreement, ensuring the quality of our dataset. To further ensure the quality the annotators were taken from diverse races and gender and an expert resolved the disagreements.

References

Lefteris Jason Anastasopoulos and Jake Ryland Williams. 2019. A scalable machine learning approach for measuring violent and peaceful forms of political protest participation with social media data. *Plos one*, 14(3):e0212834.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Samin, M Sohel Rahman, Anindya Iqbal, and Rifat Shahriyar. 2021. Banglabert: Combating embedding barrier for low-resource language understanding. *arXiv preprint arXiv:2101.00204*.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad Uddin, Kazi Mubasshir, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Lagnuage model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL*.

Vittorio Bufacchi. 2005. Two concepts of violence. *Political studies review*, 3(2):193–204.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 2930, New York, NY, USA. Association for Computing Machinery.

Johan Galtung. 1969. Violence, peace, and peace research. *Journal of peace research*, 6(3):167–191.

Johan Galtung. 1990. Cultural violence. *Journal of peace research*, 27(3):291–305.

Liangjie Hong and Brian D Davison. 2010. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. Banfakenews: A dataset for detecting fake news in bangla. *arXiv preprint arXiv:2004.08789*.

Qianjia Huang, Vivek Kumar Singh, and Pradeep Kumar Atrey. 2014. Cyber bullying detection using social and textual analysis. In *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia*, SAM '14, page 36, New York, NY, USA. Association for Computing Machinery.

Ting-Hao Kenneth Huang. 2014. Social metaphor detection via topical analysis. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 19, Number 2, June 2014*.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. Sentnob: A dataset for analysing sentiment on noisy bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271.

Marc-André Kaufhold and Christian Reuter. 2019. Cultural violence and peace in social media. *Information Technology for Peace and Security: IT Applications and Infrastructures in Conflicts, Crises, War, and Peace*, pages 361–381.

David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. 2018. The science of fake news. *Science*, 359(6380):1094–1096.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Daniel Mider. 2013. The anatomy of violence: A study of the literature. *Aggression and Violent Behavior*, 18(6):702–708.

Maya Mirchandani. 2018. Digital hatred, real violence: Majoritarian radicalisation and social media in india. *ORF Occasional Paper*, 167:1–30.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Mohammad Javed Kaisar Ibne Rahman. 2022. Religious nationalism in digitalscape: An analysis of the post-shahbag movement in bangladesh. *Open Journal of Social Sciences*, 10(5):201–218.

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2022. Bd-shs: A benchmark dataset for learning to detect online bangla hate speech in different social contexts. *arXiv preprint arXiv:2206.00372*.

Sajal Roy, Ashish Kumar Singh, et al. 2023. Sociological perspectives of social media, rumors, and attacks on minorities: Evidence from bangladesh. *Frontiers in Sociology*, 8:1067726.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Tenth international AAAI conference on web and social media*.

Nazia Tasnim, Md Istiak Hossain Shihab, Moqsadur Rahman, Sheikh Rabiul Islam, and Mohammad Ruhul Amin. 2021. Exploring the scope and potential of local newspaper-based dengue surveillance in bangladesh. *arXiv preprint arXiv:2107.14095*.

Nicole Tausch, Miles Hewstone, and Ravneeta Roy. 2009. The relationships between contact, status and prejudice: An integrated threat theory analysis of hindu–muslim relations in india. *Journal of Community & Applied Social Psychology*, 19(2):83–94.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

A Appendices

A.1 Events

The specific sources containing the description of violent incidents are detailed in the Table 8.

Event	Sources
Cumilla Durga Puja Incident	https://w.wiki/4Eti
India Hijab Incident	https://w.wiki/6FAb
Ramu Incident	https://w.wiki/6FAd
Blogger Avijit Murder	https://w.wiki/6FAf
Nasirnagar Violence	https://w.wiki/6FAh
Dhaka College Vs New Market	https://www.thedailystar.net/

Table 8: This table provides different sources containing the description of violent incidents based on which the proposed dataset was created.

A.2 Sources

We have analyzed data pertaining to online comments on popular YouTube news channels from Bangladesh and India. The specific number of comments collected from each channel is presented in Table 9.

Source	Number of Instances
Somoy Tv	28,241
Ekattor Tv	10,114
Independent Television	18,333
BBC News Bangla	14,339
ATN News	1,759
RTV News	1,717
Jamuna TV	64,853
India Today	3,314
Hindustan Times	16,922
Republic World	8,266
Zee 24 Ghanta	374
Total Instances	**168,232**

Table 9: The table presents the number of comments collected from various YouTube news channels that broadcasted videos on the violent incidents cited in this paper.

A.3 Model Hyperparameter

We have fine-tuned the pre-trained language model using a set of hyperparameter values. These values are presented in Table 10

Hyperparameter	Value
learning rate	1e-5
train batch size	8
evaluation batch size	8
epochs	50
evaluation steps	250
early stopping patience	5

Table 10: The table depicts the hyperparameter of the fine-tuned pre-trained language model

Figure 5: Social Rights

A.4 LDA Seeds

This section contains some final seeds used in the LDA which are provided in Table 11.

A.5 Word Cloud

In order to gain insight and potentially discover useful information, a word cloud analysis was conducted on each incident which are provided in Figure 5 to 9

Figure 6: Non-violence

Classification	Words
Kill/Attack	**Seed Word List:** হামলা, ভাঙচুর, হত্যা, মারা, আঘাত, ভাংচুর, ধ্বংস, দাংগা, মারার, যুদ্ধ, ভেঙে **English Translation:** Assault, vandalize, kill, kill, hurt, vandalize, destroy, riot, kill, fight, break
	Extended Word List:হত্যা, ধ্বংস , মারা, মেরে, খুন, যুদ্ধ, ধর্ষণ, রক্ত, জিহাদ, হামলা, সংঘাত, কেটে, বোমা, ধর্ষন, , যুদ্ধের, আক্রমণের, জিহাদেরধর্ষণের , জ্বালিয়ে, গণধর্ষণ, সংঘর্ষের , পুড়িয়ে, ভেংগে , ভাঙ , ধোলাই, গর্দান, গজব **English Translation:** Kill, Destroy, Kill, Kill, Kill, Fight, Rape, Blood, Jihad, Attack, Conflict, Cut, Bomb, Rape, Battle, Attack, Jihad, Rape, Burn, Gang Rape, Clash, Burn, Break, Break, Hit, Neck Chopping, Curse
Re/Desocialization/ Repression/ Oppression/ Deportation	**Seed Word List:**অত্যাচার, নির্যাতন, অন্যায়, জোর, জুলুম,,নির্যাতন,গ্রেফতার **English Translation:** Torture, torture, injustice, force, oppression, torture, arrest
	Extended Word List:বন্ধ ,ভয়, নির্যাতন , বয়কট , তান্ডব , অন্যায়ের , চাপিয়ে , শোষণ , চাপানোর , ক্রিতদাস , কৃতদাস , আটক , বাইন্ধা , বেঁধে, বর্বরোচিত, নির্যাতনের, হুমকি **English Translation:** Stop, Fear, Torture, Boycott, Tyranny, Injustice, Impose, Exploit, Oppress, Enslaved
Passive Violence/ Justification	**Seed Word List:**গুজব, নোংরামি, উচিত, জঙ্গি, নাস্তিক, উগ্রবাদী, জায়েজ, দালাল, দালালি, অবমাননা **English Translation:** Rumour, Filth, Should, Militant, Atheist, Extremist, Legitimate, Broker, Broker, Contempt
	Extended Word List: নোংরামি , অবমাননা , দালাল , পাগল , গুজব , মিথ্যা , বাজে , চোর, নোংরা , কাফের , সন্ত্রাসীদের , দায় , বাটপার , সাম্প্রদায়িকতা , উস্কানি , ব্যভিচারের ,জঙ্গিদের , জালিম , রাজাকার , ধামাচাপা , চামচা , কটাক্ষ , জালেম , কাফির , দালালরা , কুলাঙ্গারদের , উগ্রবাদীদের , বেহায়া , কুলাঙ্গাররাই **English Translation:** Scumbag, Scumbag, Scumbag, Lunatic, Rumour, Liar, Rascal, Thief, Filth, Infidel, Terrorists, Liability, Fraud, Communalism, Incitement, Adultery, Militants, Oppressors, Extortionists, Cover-up, Scoundrels, Slanderers, Tyrants, Infidels, Brokers, Curmudgeons, Extremists, Perverts, Curmudgeons
Social Rights	**Seed Word List:**প্রতিবাদ, অধিকার, স্বাধীনতা, দাবি, বিচার, আন্দোলন, স্বাধীন, মিছিল **English Translation:** Protest, Rights, Freedom, Demand, Trial, Movement, Independent, March
	Extended Word List:বিচার, স্বাধীনতা, অধিকার, আন্দোলন , স্বাধীন, তদন্ত, সমর্থন , বিচারের, নিরাপত্তা , গ্রেফতার , অভিযোগ , মিছিল , প্রতিরোধ , প্রতিবাদী ,আন্দোলনের , আন্দোলনে, গ্রেপ্তার, মর্যাদা, মানবাধিকার, জাগ্রত, গনতন্ত্র, হরতালে, বিক্ষোভ, চেতনা, আইনি, জবাবদিহি **English Translation:** Trial, Freedom, Rights, Movement, Independent, Investigation, Support, Trial, Security, Arrest, Complaint, March, Resistance, Protest, Movement, Agitation, Arrest, Dignity, Human Rights, Vigil, Democracy, Strike, Protest, Consciousness, Legal, Accountability
Non-Violence	**Seed Word List:**ধন্যবাদ, সম্মান, শান্তি, সৃষ্টি, সুন্দর, জন্ম, **English Translation:** Thanks, Honor, Peace, Creation, Beautiful, Birth
	Extended Word List:শিক্ষা, ধন্যবাদ, পবিত্র , সৃষ্টি, রক্ষা , সুন্দর, ভাল, জন্ম, আশা, চিন্তা, খুশি , একমত , প্রিয়, নিরপেক্ষ , পছন্দ, দুঃখজনক , শান্তিতে , মানবতা , সুযোগটাও , নিরপেক্ষতা, ভাই, সুস্থ , কল্যাণ , সত্যটা , আশ্রয় , রক্ষার , ভদ্র , গর্ব , সৌন্দর্য **English Translation:** Teaching, Thanking, Holy, Creating, Protecting, Beautiful, Good, Born, Hoping, Thinking, Happy, Agree, Dear, Neutral, Like, Sad, At Peace, Humanity, Opportunity, Impartiality, Brother, Health, Welfare, Truth, Shelter, Protection, Polite, Pride, Beauty

Table 11: The table presents each seed word list followed by respective final word list extended by Guided LDA with human in the loop for five different categories: Kill/Attack, Resocialization/Desocialization/Oppression/Deportation, Passive Violence, Social Justice and Peace/Non-Violence.

Bangla Comment	Question 1	Question 2	Question 3	Question 4	Label
ছাত্রদের আন্দোলন সঠিক,ব্যবসায়ীদেরকে কঠিন শাস্তি দেওয়া হোক,মারো আরো জোরে মারো ব্যবসায়ী মাগির পোলারা ডাকাত,মার আরো জোরে মার (The students' movement is correct; the businessmen should be severely punished. Beat them harder; beat those corrupted businessmen)	yes	yes	yes	-	Direct Violence
পূজা মণ্ডপে হামলা করার উদ্দেশ্যে বা পূজা উৎসবকে বানচাল করার উদ্দেশ্যে পরিকল্পিতভাবে এই কাজটি করা হয়েছে।বিরোধী দলগুলো কোন ইসু খুঁজে পাচ্ছে না সরকারকে ঘায়েল করার জন্য তাই জনগণকে ধর্মীয় সুরসুরি দিয়ে ঘোলা পানিতে মাছ শিকার করা যায় কিনা (This act was done deliberately to attack the puja pandal or to disrupt the puja festival. Opposition parties can't find any issues to blame the government)	yes	no	-	-	Passive Violence
নিউমার্কেটে দোকানের করমচারীরা মেয়েদের ইভটিজিং প্রতিদিনের ঘটনা এর আগেও ছাত্র/ছাত্রী দের সাথে এমন হয়েছে শক্ত স্টেপ না নিলে এসব দোকানের করমচারীদের সন্ত্রাসী মূলক কারজকলাপ বন্ধ হবে (Every day in New Market, the shop employees are eve-teasing the girls. This has happened with students before. If strict measures are not taken, these shop employees will continue their terrorist activities)	-	-	-	yes	Non-Violence
একদম মেরে ছাত্রদের হাড্ডি গুড়া করে দে।এরা ছাত্র না এরা আগামী দিনের সন্ত্রাস (Completely break the students' bones. They are not students; they are terrorists of the future)	yes	yes	yes	-	Direct Violence
এটা কোনো কথা হোলো সবাই দেখেছে কি হোয়েছে আর তোমরা বলছো গুজব আমার মনে হয় কোরআনের সব চেয়ে বড় শত্রু তোমরা (Is this a joke? Many people have seen what happened, and you are saying it's a rumor. I believe the biggest enemies of the Quran are people like you)	yes	no	-	-	Passive Violence
পুলির= চোর। এরা ব্যবসায়িদের পক্ষকেই বেছে নিবে। কারন ব্যবসায়িরা তো টাকা দিবে। ছাত্ররাতো আর টাকা দিতে পারবেনা। (Police = Thieves. They will always favor businessmen because businessmen will give money. Students, on the other hand, won't be able to)	-	-	-	no	Non-Violence

Table 12: The table displays Bangla comments from YouTube videos pertaining to various incidents, along with their labels determined by answers to four specific questions as presented in the data annotation framework. The decision process starts with one question at each level, leading to ramifications into a sub-tree based on types of violence.

Figure 7: Kill/Attack

Figure 8: Resocialization, Deportation or Opression

Figure 9: Passive Violence

Model Name	Direct			Passive			Non-Violence			Macro		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Random Baseline	0.1395	0.3162	0.1935	0.3435	0.3416	0.3426	0.4983	0.3233	0.3921	0.3271	0.3270	0.3094
Majority Voting	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5391	1.0000	0.7005	0.1797	0.3333	0.2335
Unigram (U)	0.7159	0.3214	0.4437	0.7087	0.5659	0.6293	0.6942	0.8801	0.7761	0.7063	0.5891	0.6164
Bigram (B)	0.7692	0.1020	0.1802	0.6232	0.2062	0.3099	0.5823	0.9470	0.7212	0.6583	0.4184	0.4038
Trigram (T)	0.6667	0.0102	0.0201	0.4364	0.0576	0.1017	0.5472	**0.9707**	0.6998	0.5501	0.3462	0.2739
U+B	0.7162	0.2704	0.3926	0.6896	0.5540	0.6144	0.6851	0.8801	0.7705	0.6970	0.5681	0.5925
B+T	0.8667	0.0663	0.1232	0.6404	0.1751	0.2750	0.5754	0.9637	0.7205	0.6941	0.4017	0.3729
U+B+T	0.7042	0.2551	0.3745	0.7006	0.5276	0.6019	0.6772	0.8926	0.7702	0.6940	0.5584	0.5822
Char-1-gram (C1)	0.5667	0.1735	0.2656	0.5917	0.5108	0.5483	0.6604	0.8382	0.7388	0.6063	0.5075	0.5176
Char-2-gram (C2)	0.7778	0.3571	0.4895	0.6554	0.6067	0.6301	0.7248	0.8633	0.7880	0.7193	0.6091	0.6359
Char-3-gram (C3)	0.8090	0.3673	0.5053	0.7046	0.6235	0.6616	0.7317	0.8898	0.8030	0.7484	0.6269	0.6566
Char-4-gram (C4)	0.8182	0.3214	0.4615	0.7478	0.6187	0.6772	0.7159	0.9066	0.8000	0.7606	0.6156	0.6462
Char-5-gram (C5)	**0.8519**	0.2347	0.3680	0.7219	0.5540	0.6269	0.6851	0.9135	0.7830	0.7530	0.5674	0.5926
Char-6-gram (C6)	0.8478	0.1990	0.3223	0.7355	0.4868	0.5859	0.6637	0.9331	0.7757	0.7490	0.5396	0.5613
C2+C3	0.7789	0.3776	0.5086	0.7008	0.6235	0.6599	0.7326	0.8828	0.8008	0.7375	0.6280	0.6564
C2+C3+C4	0.8353	0.3622	0.5053	0.7216	0.6403	0.6785	0.7349	0.8968	0.8078	0.7639	0.6331	0.6639
C2+C3+C4+C5	0.8182	0.3214	0.4615	0.7228	0.6379	0.6777	0.7299	0.9010	0.8065	0.7570	0.6201	0.6486
C2+C3+C4+C5+C6	0.8219	0.3061	0.4461	0.7210	0.6259	0.6701	0.7263	0.9066	0.8065	0.7564	0.6129	0.6409
Multilingual Bert (MBERT)	0.6752	0.5408	0.6006	0.7331	0.5731	0.6433	0.7400	0.8745	0.8018	0.7162	0.6628	0.6819
Xlm-RoBERTa (Base)	0.6882	0.6531	0.6702	0.7241	0.6859	0.7044	0.7957	0.8312	0.8131	0.7360	0.7234	0.7292
DistilBERT	0.5455	0.5510	0.5482	0.6300	0.6451	0.6374	0.7773	0.7643	0.7707	0.6509	0.6535	0.6521
BanglaBERT	0.7577	**0.7500**	**0.7538**	**0.7449**	**0.7842**	**0.7640**	**0.8580**	0.8340	**0.8458**	**0.7869**	**0.7894**	**0.7879**

Table 13: The table shows the outcomes classification using baselines, linguistic features, and pre-trained language models for the development set. All the experiments used the same dataset and parameters for a fair evaluation. We observe that BanglaBERT achieved the best F1-score for most of the individual classes and overall dataset.

BanglaCHQ-Summ: An Abstractive Summarization Dataset for Medical Queries in Bangla Conversational Speech

Alvi Aveen Khan[1], **Fida Kamal**[1], **Md. Abrar Chowdhury**[1], **Tasnim Ahmed**[1,2], **Md. Tahmid Rahman Laskar**[3], and **Sabbir Ahmed**[1]
[1]Islamic University of Technology, [2]Queen's University, [3]York University
[1]{alviaveen, fidakamal, abrar35, sabbirahmed}@iut-dhaka.edu
[2]tasnim.ahmed@queensu.ca, [3]tahmid20@yorku.ca

Abstract

Online health consultation is steadily gaining popularity as a platform for patients to discuss their medical health inquiries, known as Consumer Health Questions (CHQs). The emergence of the COVID-19 pandemic has also led to a surge in the use of such platforms, creating a significant burden for the limited number of healthcare professionals attempting to respond to the influx of questions. Abstractive text summarization is a promising solution to this challenge, since shortening CHQs to only the information essential to answering them reduces the amount of time spent parsing unnecessary information. The summarization process can also serve as an intermediate step towards the eventual development of an automated medical question-answering system. This paper presents 'BanglaCHQ-Summ', the first CHQ summarization dataset for the Bangla language, consisting of 2,350 question-summary pairs. It is benchmarked on state-of-the-art Bangla and multilingual text generation models, with the best-performing model, BanglaT5, achieving a ROUGE-L score of 48.35%. In addition, we address the limitations of existing automatic metrics for summarization by conducting a human evaluation. The dataset and all relevant code used in this work have been made publicly available[1].

1 Introduction

The answers to general health inquiries can often be obtained by utilizing internet search engines, but addressing queries by individual users in a manner that caters to their specific circumstances remains a manual process. Such queries, known as Consumer Health Questions (CHQs), are frequently found on online health forums, and answering them is becoming increasingly time-consuming and labour-intensive for medical professionals (Ma et al., 2018). The task is made even more difficult by the fact that patients are often overly descriptive when asking questions, providing unnecessary details (Roberts and Demner-Fushman, 2016). The ability to identify and discard these unnecessary details would save a lot of time for the response providers and would also be an important step towards the eventual development of an automated question-answering system (Abacha and Demner-Fushman, 2019a).

Abstractive text summarization is the task of generating a shortened and human-readable version of the original text that retains the important information (Nallapati et al., 2016). Despite the recent improvement in this domain due to the development of transformer-based architectures as well as the greater availability of data, progress has been somewhat limited in CHQ summarization (Abacha and Demner-Fushman, 2019b; Yadav et al., 2022a, 2021). This shortcoming is particularly notable for low-resource languages like Bangla (Alam et al., 2021), for which there is no existing work on this task.

Developing a dataset dedicated to this language presents a substantial challenge to existing architectures for several reasons. Firstly, Bangla is an exceedingly complicated language in comparison to English, allowing for more flexible sentence structuring (Sinha et al., 2016) and a significantly greater number of inflections (220 as opposed to just 9 in English (Bhattacharya et al., 2005)), resulting in noisier text. Furthermore, the diversified dialects exacerbate the issue, with the language used in one region frequently being entirely unintelligible in another (Shahed, 1993). Navigating this complexity and successfully identifying the relevant medical information is a significantly complicated task.

Unfortunately, the Bangla text summarization architectures currently available do not account for the complications of informal speech in medical contexts since they were mostly trained on

[1]https://github.com/alvi-khan/BanglaCHQ-Summ

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 85–93

news articles (Bhattacharjee et al., 2023; Hasan et al., 2021), making them unsuitable for summarizing medical text. This paper addresses the lack of medically relevant data by introducing the first human-annotated Bangla CHQ summarization dataset, 'BanglaCHQ-Summ', consisting of 2350 question-summary pairs. The data was collected from a public online health forum used by hundreds of native Bangla speakers, allowing it to present an accurate representation of the health questions that are generally present on online forums. In addition to the dataset, we also discuss the shortcomings of established evaluation metrics of text summarization tasks and explore a methodology for human evaluation that addresses the shortcomings.

2 Related Work

Although a large amount of work has been dedicated to text summarization in general (Allahyari et al., 2017; Nenkova and McKeown, 2012; El-Kassas et al., 2021), very limited literature is devoted to CHQ summarization. To the best of our knowledge, there are only two datasets available for the task, 'MeQSum' (Abacha and Demner-Fushman, 2019b) and 'CHQ-Summ' (Yadav et al., 2022b), and both consist exclusively of English text. The lack of work addressing CHQ summarization is a major limitation for the domain since domain-specific models are known to outperform general ones (Trewartha et al., 2022).

MeQSum was the first dataset for consumer health question summarization, consisting of 1,000 samples collected from the U.S. National Library of Medicine. The dataset has a relatively small size but was also the only medical question summarization dataset available at that time. Yadav et al. (2022b) attempted to address the lack of available datasets by introducing the 'CHQ-Summ' dataset. This dataset consists of 1,507 samples collected from the Yahoo community question-answering forum. The informal source of the data enhances its diversity and presents a more realistic depiction of the questions that medical professionals are likely to encounter.

A notable shortcoming of the existing literature is the lack of diversity in language. The advantages of CHQ summarization should prove extremely beneficial if its application can be extended to support overpopulated regions such as Bangladesh, where healthcare workers are frequently overwhelmed by the volume of patients (Razu et al., 2021). Introducing a Bangla dataset contributes towards solving this issue, and the knowledge gained is also transferable to other Indo-Aryan languages of the Indian subcontinent.

3 The BanglaCHQ-Summ Dataset

In this section, we demonstrate how we curated the proposed BanglaCHQ-Summ dataset.

3.1 Data Collection

We collected the questions from a renowned medical forum[2] that publicly releases questions posted by users, along with answers provided by medical professionals. Given that the data was collected from a public health forum, it can be reasonably assumed that the user base consisted of individuals with average medical knowledge. This user base consists of individuals from diverse backgrounds based on the linguistic variety of the questions, which is a particularly strong point for the dataset since it presents an accurate representation of the variety of the Bangla language discussed earlier. The forum contains questions belonging to a total of 32 categories, which allows the samples to cover a broad spectrum of health issues. However, the information related to the categories has been omitted from our dataset as the category assignment is done by the patients while posting the queries, which can often be inaccurate.

3.2 Pre-Processing

A portion of the collected data contained sensitive information. To protect the privacy of the patients, such personally identifiable information has been removed by utilizing regular expressions to identify email addresses and phone numbers and a Bangla Named Entity Recognition model[3] to identify names. The data was then also inspected manually. Additionally, duplicate entries, URLs, and spam text were also removed as part of the overall data-cleaning process.

3.3 Annotation

A team of 5 annotators with at least an undergraduate level of education was chosen after carefully evaluating their summarization capabilities in the Bangla language. The primary instruction provided to the annotators was to make the text as

[2] https://daktarbhai.com/

[3] https://pypi.org/project/bnlp-toolkit/

Question	I have chronic stress and anxiety, I am loosing everything in my life, but do not want pills, what can I do? I have problems with stress, however it is not just that, but the fact that every time I start with this condition it turns into a huge fear of choking and my mind starts telling me not to eat. The last time it happened I did not eat anything solid for four months and I suffered severe damage in other parts of my body like my stomach and my heart which is worst. This time it is starting again and I am two weeks under this condition. The last time I was using antidepressants and other drugs, but when I tried cutting them the anxiety made me feel worst. This is why I changed my treatment, now I use relaxation exercises with the help on my doctor. The last time it helped me a lot, but this time I think I need more help. I am taking meditation and tai chi courses and I am expecting to take yoga classes as well. The problem is that this is taking away my life, I have doubts on whether I will be cured one day or if it will take so long that everything I have now will be lost. I need help.
Summary	What are possible non-drug treatments for chronic stress and anxiety?

Table 1: Sample summary from the CHQ-Summ dataset (Yadav et al., 2022b)

concise as possible without discarding any information essential to answer the question accurately. The complete set of guidelines is provided in Appendix A.1.

Appendix A.2 showcases a few samples of the annotated summaries from the dataset. Each annotator was provided with a set of 500 questions, among which 6% was common. The summaries of the common questions were later used to calculate the inter-annotator agreement (IAA) using the ROUGE-L metric (Lin, 2004), with the average score being 50.11%. However, this score does not take semantic differences into account, an issue previously highlighted by Yadav et al. (2022b) when they found a similar score for their work. A manual evaluation of the summaries clearly demonstrates significant semantic overlap. To quantify this, we refer to the BERTScore metric (Zhang et al.), which calculates the semantic similarity between sentences. The average BERTScore for the common questions provided to the annotators is 90.84%. Hence, we conclude that despite there being differences in the phrasing used by the annotators, the content of their summaries is largely the same.

A portion of the annotated summaries, specifically the portion used as the test set for evaluation of the benchmark models, was further verified by a physician, who determined whether the annotated summaries were appropriate and medically relevant. Based on this, we found that they strongly agreed with 80% of the annotated summaries, with only minor issues being found in the remaining 20%, which they assured us do not make the summaries inaccurate.

3.4 Dataset Attributes

The final dataset consists of 2350 question-summary pairs. The average length of the original questions was 326 words, compared to the average length of 136 words for the annotated summaries. This large difference in lengths provides evidence of the fact that users on health forums tend to ask overly descriptive questions, which in turn require more effort to parse.

The annotated summaries from our dataset are noticeably longer than those found in existing work. This difference is deliberate. Analyzing the MeQSum and CHQ-Summ datasets, we found that they prioritized shorter lengths over information retention. An example of this is provided in Table 1, which shows a sample summary taken directly from the CHQ-Summ dataset. Although the annotated summary addresses the main question asked, it leaves out a large number of additional details, such as the patient having past issues with stress to the extent of not eating solids and that they have tried using antidepressants and other drugs. The summary only allows for a generic response without considering the patient's specific circumstances. To avoid this, our annotators were instructed to retain all medically relevant information. This allows us to obtain shortened questions while still addressing the specific situation being faced by the patient.

An important finding of the summarization process was that a significant portion of patients explicitly mentioned being unable to visit medical professionals in-person during the COVID-19 pandemic. Analyzing the data from the online platform reveals a correlation, visualized in Fig. 1. The diagram compares the daily count of questions posed on the platform with the number of

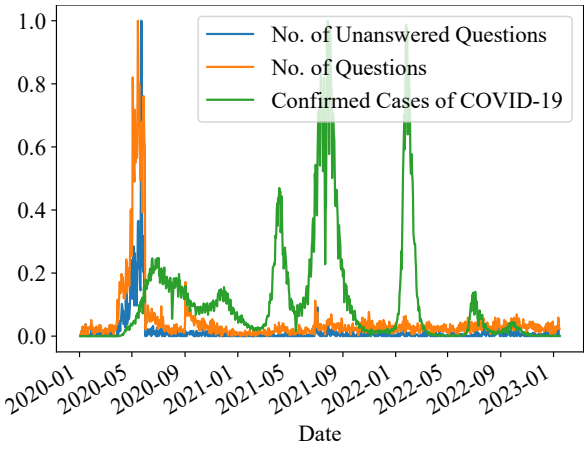


Figure 1: Comparison of time-frames for a rise in question count and unanswered questions on the online health platform with the rise in new cases of COVID-19 in Bangladesh. Values are normalized due to the large difference in scale.

confirmed cases of COVID-19 in Bangladesh[4], where the majority of the user base of the online platform resides. During the initial wave of the pandemic, there was a significant rise both in the number of questions asked and the number of questions remaining unanswered. The trend does not repeat itself during the latter waves, presumably due to the general public becoming well-informed by that time. This finding reinforces the need for Bangla CHQ summarization and, ideally, an automated question-answering system (Laskar et al., 2020) to provide support to the medical staff in unprecedented scenarios such as a pandemic.

4 Experiments

To benchmark model performance on our proposed dataset, we conduct two types of evaluation: (i) Automatic and (ii) Human. We split the dataset into training, validation, and test sets following an 80:10:10 ratio. Below, we present our findings.

4.1 Automatic Evaluation

For automatic evaluation, we experimented with one Bangla text generation model, BanglaT5 (Bhattacharjee et al., 2023), as well as two multilingual ones, mT5 (Xue et al., 2021) and mBART (Tang et al., 2020). The details of the experimental setup are provided in Appendix A.3.

To evaluate the model, we used the ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-L (RL), and BERTScore (BS) metrics. For the BERTScore metric, layer 12 of the BanglaBERT (Bhattacharjee et al., 2022) model was used. The ROUGE scores measure the degree of overlap between the generated and reference summaries and are a commonly used evaluation metric for text summarization tasks. The BERTScore metric measures the semantic similarity between generated and reference summaries and is known to correlate better with human evaluation.

Model	R1	R2	RL	BS
Bangla T5	50.05	29.11	48.35	89.91
mT5	40.99	22.84	39.76	88.50
mBART	47.23	27.15	45.86	89.38

Table 2: Automatic evaluation results of BanglaCHQ-Summ

Our results, presented in Table 2, show that BanglaT5 outperforms both multilingual models on all four metrics, demonstrating that models pretrained on a language-specific corpus outperform multilingual ones.

4.2 Human Evaluation

As discussed in section 3.3, the ROUGE score gives limited insight into the quality of the generated summaries. BERTScore can better capture semantic similarities but still does not account for several important factors, such as the coherence, logical flow, or overall correctness of the text. To address these limitations, we have explored a methodology to establish quantitative metrics to evaluate summaries following Laskar et al. (2022).

To carry out this evaluation, a group of 3 annotators was provided with the same set of 30 samples along with the summaries generated by each of the three models. They rated the generated summaries on a scale of 1 to 5 based on the following metrics:

Informativeness (I): Measures the extent to which the information required to answer the question was retained in the summary. Including unnecessary information does not lower this score.

Conciseness (C): Measures how short the summary is. Including unnecessary information or being verbose while describing the necessary information lowers this score.

Fluency (F): Measures how coherent and fluent the summary is.

Table 4 shows the average score assigned to each model based on the evaluation of the three annotators. The evaluation process revealed

[4] https://covid19.who.int/region/searo/country/bd

Sample 1	
Original Question	আমার বয়স ৩০। অনেক দিন ধরে কোমরে ব্যথা করে। আমি ডাঃ দেখিয়েছি। এক্সরে করানোর পর আমাকে ন্যাপ্রক্সিন ১০ দিন দু বেলা করে খেতে দেয়। আর সাথে ক্যালসিয়াম, নিবলন খেতে দেয় ১ মাস। কিন্তু এ পর্যন্ত ব্যাথা যায় নাই। কোমরে প্রেশার দিলে ব্যাথা করে। এখন কি করা যায়।
BanglaT5 Summary	বয়স ৩০। কোমরে ব্যথা। ডাক্তার ন্যাপ্রক্সিন ১০ দিন দু বেলা করে দেয়। সাথে ক্যালসিয়াম, নিবলন খেতে দেয় ১ মাস। কিন্তু ব্যাথা যায় নাই। প্রেসার দিলে ব্যাথা করে। কি করনীয়?
mT5 Summary	বয়স ৩০। অনেক দিন ধরে কোমরে ব্যথা করে
mBART Summary	বয়স ৩০। কোমরে ব্যাথা। কি করনীয়?
Sample 2	
Original Question	আমার আম্মুর পিঠের ঠিক মাঝে ভীষণ জ্বলে প্রায় এক বসর ধরে এমন হয়। ডাক্তার দেখানো হয়েছে শুধু গ্যাস্ট্রিকের ওষুধ দেয়। কিন্তু কোন কাজ হয়না। এখন খুব জ্বলে। মাঝে মাঝে একটু কম থাকে। কোন বিশেষজ্ঞ দেখালে ভাল হয়? আর কি সমস্যা হতে পারে? বিশেষজ্ঞদের পরামর্শ চাই।
BanglaT5 Summary	পিঠের ঠিক মাঝে ভীষণ জ্বলে প্রায় এক বছর ধরে। ডাক্তার গ্যাস্ট্রিকের ওষুধ দেয়। কাজ হয়না। এখন খুব জ্বলে। মাঝে মাঝে একটু কম থাকে। কোন বিশেষজ্ঞ দেখালে ভালো হবে? আর কি সমস্যা হতে পারে?
mT5 Summary	পিঠের ঠিক মাঝে ভীষণ জ্বলে প্রায় এক বসর ধরে। ডাক্তার শুধু গ্যাস্ট্রিকের ওষুধ দেয় কিন্তু কাজ হয়না
mBART Summary	আম্মুর পিঠের ঠিক মাঝে ভীষণ জ্বলে প্রায় এক বসর ধরে এমন হয়। ডাক্তার শুধু গ্যাস্ট্রিকের ওষুধ দেয়।

Table 3: Samples of summaries generated by the benchmark models

Model	I	C	F
Bangla T5	4.09	3.83	4.27
mT5	2.94	4.18	4.39
mBART	3.34	4.00	4.12

Table 4: Human evaluation results of BanglaCHQ-Summ

that summaries with high informativeness scores tended to have relatively low conciseness scores and vice versa. This indicates that the models struggled to retain all the correct information while also being concise. Amongst the models, BanglaT5 shows significant superiority in preserving required information in its summaries but has comparatively less proficiency in conciseness compared to the multilingual models. This can be demonstrated with reference to the samples of generated summaries in Table 3.

We find from Table 3 that the first sample shows a serious error made by the multilingual models. The patient complains of waist pain, which all three models capture in their summaries, but only BanglaT5 includes the additional information regarding medicine prescribed to the patient by a doctor, a critical piece of information. On the other hand, the second sample illustrates the tendency of BanglaT5 to be excessively descriptive. The patient describes a burning sensation in their back and mentions that the medicine given by doctors does not provide relief. The latter part of the question repeats this complaint, adding no new information. The summary generated by BanglaT5 accurately reflects the main complaint but retains the repetitive portions, while the summaries generated by the multilingual models exclude the repetitive portions.

5 Conclusion

In this paper, we propose the first CHQ summarization dataset for the Bangla Language. The source of the data used in the creation of the dataset also presents an advancement towards a more accurate representation of the diversity of the language. In addition, we explore a methodology for human evaluation that addresses the limitations of existing text summarization evaluation metrics. Given the sensitive nature of the public health domain, improvements in the performance of the summarization models, alongside evaluating how Large Language Models (Jahan et al., 2023) perform in this dataset could be a good direction for future research.

Limitations

One limitation of this work is that the size of the proposed dataset is quite modest. However, even the existing English question summarization datasets have limited sizes. In this regard, our dataset, although being for the low-resourced Bangla language, surpasses the sizes of similar datasets available in English.

Another limitation of this work is that, while our dataset has been benchmarked on widely used text summarization models, the use of such models assumes the availability of significant computational resources that many organizations may not be able to afford. Although utilizing computational resources from third-party institutions will likely be able to address this issue, the sensitive nature of medical data makes sharing the data with third parties an unfavorable solution.

Ethics Statement

The Consumer Health Questions (CHQs) collected to prepare our dataset are publicly available. As of October 17, 2023, the terms and conditions of the online health platform[5] also do not prohibit the usage of publicly available data for research purposes. Extensive measures were taken to safeguard the privacy of all patients involved. No personal information outside of the CHQs was collected. In addition to automated measures, the dataset was manually inspected to ensure no personally identifiable information was present.

The individuals involved in annotating the dataset were provided monetary compensation for their work, which is above the minimum wage. The annotation process has also been anonymized to prevent any violations of the privacy of the annotators.

Acknowledgement

We are grateful to Islamic University of Technology (IUT) for providing the necessary funding for this research. We would also like to express our utmost gratitude to Dr. Rubaiya Bari for her professional input while preparing our dataset, as well as our team of annotators. This work would not have been possible without them.

[5] https://daktarbhai.com/

References

Asma Ben Abacha and Dina Demner-Fushman. 2019a. On the role of question summarization and information source restriction in consumer health question answering. *AMIA Summits on Translational Science Proceedings*, 2019:117.

Asma Ben Abacha and Dina Demner-Fushman. 2019b. On the summarization of consumer health questions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2228–2234.

Firoj Alam, Arid Hasan, Tanvirul Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, and Rifat Shahriyar. 2023. Banglanlg and banglat5: Benchmarks and resources for evaluating low-resource natural language generation in bangla. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 714–723.

Samit Bhattacharya, Monojit Choudhury, Sudeshna Sarkar, and Anupam Basu. 2005. Inflectional morphology synthesis for bengali noun, pronoun and verb systems. In *Proc. of the National Conference on Computer Processing of Bangla (NCCPB 05)*, pages 34–43.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert systems with applications*, 165:113679.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703.

Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Huang. 2023. A comprehensive evaluation of large language models on benchmark biomedical text processing tasks. *arXiv preprint arXiv:2310.04270*.

Md Tahmid Rahman Laskar, Enamul Hoque, and Xiangji Huang. 2022. Domain adaptation with pre-trained transformers for query-focused abstractive text summarization. *Computational Linguistics*, 48(2):279–320.

Md Tahmid Rahman Laskar, Jimmy Xiangji Huang, and Enamul Hoque. 2020. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5505–5514, Marseille, France. European Language Resources Association.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Xiaojuan Ma, Xinning Gui, Jiayue Fan, Mingqian Zhao, Yunan Chen, and Kai Zheng. 2018. Professional medical advice at your fingertips: An empirical study of an online "ask the doctor" platform. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–22.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. *Mining text data*, pages 43–76.

Shaharior Rahman Razu, Tasnuva Yasmin, Taimia Binte Arif, Md Shahin Islam, Sheikh Mohammed Shariful Islam, Hailay Abrha Gesesew, and Paul Ward. 2021. Challenges faced by healthcare professionals during the covid-19 pandemic: a qualitative inquiry from bangladesh. *Frontiers in public health*, page 1024.

Kirk Roberts and Dina Demner-Fushman. 2016. Interactive use of online health resources: a comparison of consumer and professional questions. *Journal of the American Medical Informatics Association*, 23(4):802–811.

Syed Mohammad Shahed. 1993. Bengali folk rhymes: An introduction. *Asian folklore studies*, pages 143–160.

Manjira Sinha, Tirthankar Dasgupta, and Anupam Basu. 2016. Effect of syntactic features in bangla sentence comprehension. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 275–284.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Amalie Trewartha, Nicholas Walker, Haoyan Huo, Sanghoon Lee, Kevin Cruse, John Dagdelen, Alexander Dunn, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. 2022. Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. *Patterns*, 3(4):100488.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Shweta Yadav, Deepak Gupta, Asma Ben Abacha, and Dina Demner-Fushman. 2021. Reinforcement learning for abstractive question summarization with question-aware semantic rewards. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 249–255.

Shweta Yadav, Deepak Gupta, Asma Ben Abacha, and Dina Demner-Fushman. 2022a. Question-aware transformer models for consumer health question summarization. *Journal of Biomedical Informatics*, 128:104040.

Shweta Yadav, Deepak Gupta, and Dina Demner-Fushman. 2022b. Chq-summ: A dataset for consumer healthcare question summarization. *arXiv preprint arXiv:2206.06581*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Appendix

A.1 Annotation Guidelines

The annotators were instructed to make the questions as short as possible while ensuring that no information required to answer the question was discarded. Aside from this, they were also provided with a list of examples to serve as a guideline in their work. The examples, provided in Table 5, cover perfect, passable, and poor summaries as approved by a practising physician.

A.2 Summary Annotation Samples

A few samples of the annotated summaries, along with their reference questions from the dataset, are provided in Table 6.

Sample 1	
Question	আমার বড় আপার সমস্যা। বয়স ৩২, ডায়াবেটিস রোগী। সর্বশেষ ডায়াবেটিস পরীক্ষা করিয়েছিলেন সপ্তাহখানেক আগে। সুগার লেভেল ছিল ১০। গতকিছুদিন আগে উনার দাঁতের গোড়া ফুলে উঠেছিল। এখন মনে হচ্ছে ফুলে উঠা জায়গাটা পেকে গিয়েছে, শুকাচ্ছে না। লকডাউন পরিস্থিতির কারণে ডাক্তার দেখানোও সম্ভব হচ্ছে না। উনার ছোট একটা বাচ্চা আছে। বর্ণিত অবস্থায় কি চিকি□সা নেওয়া প্রয়োজন জানালে খুব উপকৃত হবো।
Summary	বয়স ৩২। ডায়াবেটিস আছে। সপ্তাহখানেক আগে ডায়াবেটিস পরীক্ষা করলে সুগার লেভেল ১০ হয়। দাঁতের গোড়া ফুলে পেকে গেছে, শুকাচ্ছে না। পরামর্শ চাই।
Analysis	**Perfect Summary**. The summary specifies the age of the patient (relevant to diabetes), the fact that they have diabetes, the sugar level during the last test as well as the issue the patient is currently facing. All the unnecessary information has been successfully removed, such as the relationship with the patient, the fact that they cannot visit a doctor due to the lockdown and that the patient has a child.
Sample 2	
Question	ডাক্টার বলছে রুট ক্যানেল করতে। রুট ক্যানেল করলে ভালো নাকি দাত ফেলে দিয়ে দাত লাগালে ভালো? ফিউচার এর জন্য কোনটা বেটার হবে। দাত ফেলতে ভয় পাচ্ছে চোখ বা মাথা ব্যাথার জন্য। আর রুট ক্যানেল করলেও নাকি কয়েকমাস পর ব্যাথা হয় দাতে। সাজেশন চাচ্ছি একটু কি করলে ভালো হয়। দাত ব্যাথায় টিকতে পারছে না।
Summary	প্রচুর দাঁত ব্যথা। ডাক্তার রুট ক্যানাল করতে বলেছে।
Analysis	**Poor Summary**. The summary does a poor job of retaining the actual questions the patient had. The patient wanted the doctor's opinions on various things such as whether to do a root canal or remove the tooth entirely and whether doing a root canal will cause pain after a few months.
Sample 3	
Question	আসসালামুয়ালাইকুম, সপ্তাহ খানেক আগে শীলা বৃষ্টিতে ভিজেছি। এরপর ৪ - ৫ চামচ আইস ক্রিম খেয়েছিলাম। ২৫ তারিখ থেকেই শরীরের অবস্থা ভালো মনে হচ্ছিল না। ২৬ তারিখ মাগরিবের পর জ্বর আসে। পরিমাপ করে দেখা যায় ১০২ ডিগ্রি। পরদিন জ্বর কমে যায়, কিন্তু অসহ্য রকম গলা ব্যাথা শুরু হয়, যা এখন পর্যন্ত আছে। ঢোক গেলা যাচ্ছে না। যেসকল ঔষধ খেয়েছি: 1, নাপা এক্সটেন্ড ট্যাব, খাওয়া শেষ 2, ফেকযো ট্যাব, খাওয়া শেষ 3, রেলিজিন ট্যাব, খাওয়া শেষ 4, মিউকলিট সিরাপ, অল্প বাকি 5, জি ম্যাক্স ট্যাব, খাওয়া শেষ 6, ভিকজিন, চলে
Summary	বৃষ্টিতে ভিজে জোর ১০২। ঔষধ খেয়েছি, নাপা এক্সটেন্ড ট্যাব,, ফেকযো ট্যাব, রেলিজিন ট্যাব, জি ম্যাক্স ট্যাব, খাওয়া শেষ। মিউকলিট সিরাপ, অল্প বাকি, লিভোকজিন, চলে।?
Analysis	**Passable Summary**. The summary accurately captures a large amount of information, but makes a critical mistake. The patient mentioned that their fever has decreased and that they are suffering from a throat ache now. The summary does not mention this.

Table 5: Examples used as annotation guidelines.

A.3 Experimental Setup

The experimental setup consisted of an Nvidia 3090 GPU with 24 GB of VRAM. The Trainer library, available through Hugging Face was utilized, along with CUDA Version 11.6. The dataset was divided into training, validation, and test sets using the split ratio $80 : 10 : 10$. The models were trained for 50 epochs using a cross-entropy loss function along with the AdamW optimizer. Input sequences were truncated to a maximum length of 512 tokens, and the output sequences were limited to 128 tokens. Other hyperparameters include a batch size of 16, a weight decay of 0.03, and a learning rate of 1e-4 used with a linear learning rate scheduler.

Original Question	Annotated Summary
আমাকে প্রশ্নের উত্তর দেয়া হয়েছে, এ জন্য আপনাদের অসংখ্য ধন্যবাদ জানাচ্ছি, বিষয়টি হল আমি একজন ডাক্তারের দেয়া ব্যবস্থাপত্র এই এ্যাপে আপলোড করেছি, এবং এই ব্যবস্থা পত্র অনুযায়ী এখন ও ঔষধ গ্রহন করছি, সে ক্ষেত্রে ১। নিউরো বি খাওয়া হলে কোন অসুবিধা হতে পারে কিনা। ২। ঘুমের সমস্যার জন্য রি লাইফ ট্যাবলেট খাওয়া যায় কিনা , কারন ঔষধটি বাসায় সংরক্ষিত আছে। দয়া করে ব্যবস্থা পত্র দিবেন।	নিউরো বি খেলে অসুবিধা হবে কি না এবং ঘুমের জন্যে রি লাইফ ট্যাবলেট খাওয়া যায় কি না?
আসসালামু আলাইকুম। স্যার অনেক দিন ধরে আমার মাথায় ও সরিরে চুলকানি। মাথা ও দারির ভেতরে ঘাও দিয়ে ভরে গেছে। ঘাও এর কারণ এ মাথার চুল ও পরে যাচ্ছে। আমি অনেক ওষুধ খেয়েছি কোন কাজ হয়নি। দয়া করে বলবেন কি ওষুধ খেলে ভালো হবে।????	অনেক দিন ধরে মাথায় ও শরীরে চুলকানি। মাথা ও দাড়ির ভেতরে ঘা, মাথার চুল পরে যাচ্ছে। কি করবো?
হ্যালো আসসালামুআলাইকুম, হার্টে কোলেস্টরল এর মাত্রা কিভাবে নিয়ন্ত্রণে রাখতে পারবো, কোন মেডিসিন গ্রহন করলে উপকার পাবো, দয়া করে একটু জানাবেন? এবং উচ্চ রক্তচাপ নিয়ন্ত্রণে রাখতে কোন ধরনের খাবার খাবো, এবং কোন ধরনের খাবার বর্জন করবো, সে ব্যাপারে একটু জানাবেন!!	কোলেস্টরল এবং উচ্চ রক্তচাপ নিয়ন্ত্রণ করতে কি ঔষুধ খাব এবং কি খাবার বর্জন করব?

Table 6: Samples of annotated summaries from the BanglaCHQ-Summ dataset

Contextual Bangla Neural Stemmer: Finding Contextualized Root Word Representations for Bangla Words

Md Fahim, Amin Ahsan Ali, M Ashraful Amin, A K M Mahbubur Rahman
Center for Computational & Data Sciences
Independent University, Bangladesh
Dhaka-1229, Bangladesh
fahimcse381@gmail.com, {aminali,aminmdashraful,akmmrahman}@iub.edu.bd

Abstract

Stemmers are commonly used in NLP to reduce words to their root form. However, this process may discard important information and yield incorrect root forms, affecting the accuracy of NLP tasks. To address these limitations, we propose a Contextual Bangla Neural Stemmer for Bangla language to enhance word representations. Our method involves splitting words into characters within the Neural Stemming Block, obtaining vector representations for both stem words and unknown vocabulary words. A loss function aligns these representations with Word2Vec representations, followed by contextual word representations from a Universal Transformer encoder. Mean Pooling generates sentence-level representations that are aligned with BanglaBERT's representations using a MLP layer. The proposed model also tries to build good representations for out-of-vocabulary (OOV) words. Experiments with our model on five Bangla datasets shows around 5% average improvement over the vanilla approach. Notably, our method avoids BERT retraining, focusing on root word detection and addressing OOV and sub-word issues. By incorporating our approach into a large corpus-based Language Model, we expect further improvements in aspects like explainability.

1 Introduction

Large Language Models (LLMs) like BERT (Devlin et al., 2019), GPT (Brown et al., 2020), and others have proven their efficacy in various Natural Language Processing (NLP) tasks. They excel at capturing contextual information and cultural subtleties in specific languages. These models exhibit strong capabilities for addressing diverse NLP tasks, especially during their unsupervised pretraining phase. However, in low resource language like Bangla, there are so many language specific problems that haven't been resolved yet since Bangla language lack comprehensive lexicons, word embeddings, or linguistic resources. **F**irstly, there may be a good number of out-of-vocabulary (OOV) words which may hamper the NLP tasks. **S**econdly, in LLMs, tokenizing one word can result splitting into different subwords that make the model difficult to explain. In the following paragraphs, we clarify these problems with examples.

Method	**Tokens**
Original Text	সে বাড়িতে যাওয়ার পর আর যোগাযোগ করেনি
BanglaBERT Tokenizer	['সে', '[UNK]', '[UNK]', 'পর' 'আর', 'যোগাযোগ', 'করেনি']
Bangla Stemmer	[সে, 'বাড়ি', 'যাওয়', 'পর' 'আর', 'যোগাযোগ', 'করেনি']

Table 1: The Limitations of Bangla BERT which gives [UNK] tokens for many informative words of a sentence and Bangla Stemmer sometimes produce a word with no meaning and also losses the context information.

In Table 1, we show an example of Bangla sentence and outputs of the tokenizer of the Bangla BERT (Bhattacharjee et al., 2022): a BERT model trained on the Bangla Corpus to demonstrate the first kind of problems. We can easily see that the occurrence of OOV tokens represented as "[UNK]" is very frequent. This significantly impacts the model's ability to comprehend semantic and linguistic information in the sentence. One possible solution to solve the OOV problem is to find root words.

Second set of problems are observed due to the use of bangla stemmer/lemmitizer. There are many existing way for finding the root words like stemming and lemmitizer. Lemmitizer needs ground truth word mapping to find the word. On the other hand, stemming algorithms typically use heuristics to identify the suffixes of words that can be removed to obtain the root form. However, by re-

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 94–103

Method	Tokens
Original Text	নটরডেম, হলিক্রস ও ভিকারুননিসা কলেজে ভর্তি হতে পারবে না ধূমপায়ী শিক্ষার্থীরা।
BanglaBERT Tokenizer	['নট', '##র', '##ডেম', ',', 'হলি', '##ক্র', '##স', 'ও', 'ভিক', '##ার', '##ুন', '##নি', '##সা','কলেজে' 'ভর্তি', 'হতে', 'পারবে', 'না', '[UNK]', 'শিক্ষার্থীরা' , '।']

Table 2: Subwords Problem in Bangla BERT.

ducing words to their root form, a stemmer discards important information that could be useful in natural language processing tasks. In cases of bangla, a stemmer may reduce a word to an incorrect root form, leading to incorrect results. For example in Table 1, for a given bangla sentence, the bangla stemmer creates some incorrect roots that have no vector representations at all. Moreover, LLMs like Bangla-BERT also faces OOV problem very recurrently because Bangla-Bert tokenizer splits a word into one or more subwords. It has been shown that splitting words into multiple subwords is not the best option all the time. There are some cases where this might not work well (Nayak et al., 2020), (Toraman et al., 2023). Some words might have a prefix or suffix that changes the meaning of the word, but BERT's subword tokenizer might split it into separate subwords. For example, in Table 2, Bangla BERT tokenizer splits words into multiple subwords, leading to a significant increase in the number of subwords. This excessive subword splitting makes it challenging to extract the actual meaning of individual words in the sentence, thereby affecting the overall interpret-ability and comprehensibility of the model.

Considering the aforementioned limitations of Rule Based Stemmer, we want to create a Contextual Bangla Neural Stemmer for Bangla language to find better representation of words. Especifically, in our proposed method, by splitting each word of a sentence into characters in Neural Stemming Block, we will get vector representation for not only the stem word but also the unknown vocab word. A loss is used to make sure that the representations of the words should be aligned with the Word2Vec representations. Then after a liner layer transformations those representations is passed into Universal Transformer (UT) (Dehghani et al., 2018) encoder to assure of getting contextual representation of a word via self attention. Mean Pooling is used to get a sentence level representation for a sentence from those contextual word representations. A MLP layer is used and a loss is defined to align the sentence representation with the BanglaBERT's one. The whole model pipeline is described in Section 3.

Our model employs character-based representations to find root word representations, which effectively addresses the issues of OOV tokens and subword tokenization. By combining these representations with BERT, our model is capable of obtaining contextual representations for these root words, enhancing its ability to capture the semantic nuances and context of the language. We evaluate our model performance in 5 different Bangla dataset. In every dataset, our model outperforms the vanilla approach by a good margin (around 5% improvement on average). More details are described in Section 5. Please note that our goal is not retraining the BERT at all. Instead of retraining the BERT, our proposed method is used finding the root words with contextual representations and address OOV & sub-word problem. If we create a LLM based on our methodology with a large corpus, our method may outperform the Vanilla BERT in different aspects and may improve the explainability also. Therefore, the summary of the contributions of this paper is given below.

- We propose a neural network based stemmer that can be contextualized
- We propose new losses to learn root word representations with contextual information
- We design a number of experiments to show the efficacy of the proposed appoach

2 Related Work

Finding root words for Bangla word is one of the most popular tasks in Bangla Natural Language Processing (NLP). Several works have been done already. We can categorize those works in two different perspective, one is morphological method base and another is heuristic base. In morphological method based approach for root word finding Lemmitizer and Stemming are used. In heuristic base, rule base or model base approaches are used. In (Mahmud et al., 2014), a rule based stemming technique is used for finding the root word in Bangla. They use different set of rules so that they can find the stem word by cutting down the prefixes. (Das et al., 2020) enhanced

the rule based stemming techniques by improving the rules for different categories. They also incorporate Bangla Corpus and different inflections for noun, verb, and other parts of speech. (Rahit et al., 2018) introduces BanglaNet which is an approach to make a WordNet for Bangla Language. (Chakrabarty et al., 2016) uses a single layer Multi Layer Perceptron MLP for finding the lemmitize word of a word along with its contextual neighbours. (Chakrabarty and Garain, 2016) uses a distance based algorithm to find the lemma of a word with respect to a given context and part of speech of the word. (Chakrabarty et al., 2017) and (Islam et al., 2022) propose algorithm to find lemma word based on the contextutal representations. The contextual representations are derived from Bi-LSTM or Bi-GRU. No works have been proposed to find the stemmed word representations from the contextual information.

3 Methodology

The tokenizers that are used in Transformer based model like BPE, Wordpiece, Unigram split a word into multiple subwords. This may cause information discrepancy between the actual meaning of the actual word which may affect the low resource language models like Bangla language model. Besides, there also may have a good amount of out of vocabulary words in those low resource language model. One option is to find the root form of the words but this approach miss the contexutal information. Considering all scenarios, we propose a character based contextual neural stemmer which not only find the stemmed root word to surpass subword techniques but also give the contextual embeddings. For being character based model, our proposed model can also deal with the out of vocabulary issues. Our model have two major components, Character Level Neural Stemming Block, Universal Transformer Encoding Block along with two different losses for fulfilling our criteria.

3.1 Character Based Neural Stemmer

After passing a sentence into the Basic Tokenizer, we get the tokens of the sentence. Let $S = [x_1, x_2, \ldots, x_m]$ represent a sentence, where each word token x_i is split into characters.

For each word token x_i, we denote the character embeddings as $C_i = [c_{1i}, c_{2i}, c_{3i}, \ldots]$, where c_{ij} represents the embedding of the jth character.

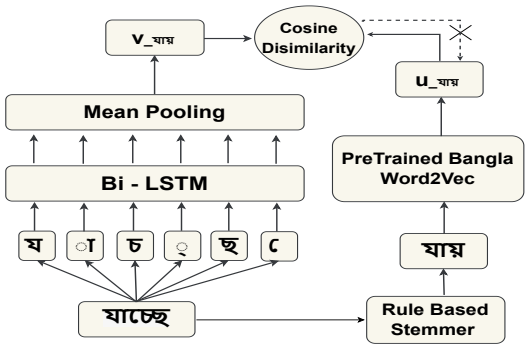


Figure 1: Model Architecture of Neural Stemmer Block

We pass C_i through an LSTM layer, which produces a sequence of hidden representations $H_i = [h_{1i}, h_{2i}, h_{3i}, \ldots]$. Each h_{ij} represents the hidden state at time step j.

The LSTM layer takes the character embeddings as input and generates hidden states using the following equations:

$$h_{ij} = \text{LSTM}(c_{ij}, h_{i(j-1)}) \tag{1}$$

Once we have obtained the sequence of hidden states H_i, we compute the mean of these hidden states to obtain the word embedding v_i:

$$v_i = \frac{1}{m} \sum_{j=1}^{m} h_{ij} \tag{2}$$

Here, m represents the total number of characters in the word token t_i. The mean aggregation operation captures the overall representation of the word by considering the contextual information contained in the LSTM hidden states.

This process allows us to derive word embeddings v_i from character embeddings, enabling us to capture fine-grained information and enhance the representation of word tokens within the given sentence. In this block, we also apply a stemming loss with the pre-trained Word2Vec representations of the stemming words. The stemming loss is described in Section 3.4.

3.2 Universal Transformer Encoder

After obtaining the neural stemming output $V = [v_1, v_2, \ldots, v_m]$ for a sentence from the Neural Stemming Block described in Section 3.1, we perform a linear transformation on each v_i to map them into a d-dimensional vector space. Now those transformed representations $V' = [v'_1, v'_2, \ldots, v'_m]$, is fed into the Universal Transformer (UT) encoder which consists of several UT

encoder blocks for finding contextual representations. We choose UT because it is a tied weight model which also uses Adaptive Computational Time (ACT). So we need less computational time for training and fine-tuning the Universal Transformer model than the vanilla Transformer model. The components of the UT encoder blocks are:

3.2.1 Positional Encoding and Time Signal

To incorporate positional and temporal information, the combined positional encoding and time signal embeddings $P_t \in \mathbb{R}^{m \times d}$, are applied where m represents the total number of positions and d is the dimensionality of the embeddings. The combined embeddings are obtained by computing the sinusoidal position and time embeddings separately for each vector dimension $1 \leq j \leq d$ and summing them:

$$P_t[i, 2j] = \sin\left(\frac{i}{10000^{(2j/d)}}\right) + \sin\left(\frac{t}{10000^{(2j/d)}}\right) \quad (3)$$

$$P_t[i, 2j+1] = \cos\left(\frac{i}{10000^{(2j/d)}}\right) + \cos\left(\frac{t}{10000^{(2j/d)}}\right) \quad (4)$$

where i represents the position index ($1 \leq i \leq m$), t represents the time-step index ($1 \leq t \leq T$), and j represents the vector dimension index ($1 \leq j \leq d$).

3.2.2 Attention Mechanism

At each step t, the UT computes revised representations $H_t \in \mathbb{R}^{m \times d}$ for all m input positions. This is done by applying the scaled dot-product attention mechanism, which combines queries Q, keys K, and values V as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (5)$$

Here, d is the number of columns of Q, K, and V. In the Universal Transformer, a multi-head version of the attention mechanism is used, with k heads:

$$\text{MultiHeadSelfAttention}(H_t) = \text{concat}(\text{head}_1, \ldots, \text{head}_k) \times W_o \quad (6)$$

Each head head_i is calculated as $\text{Attention}(H_t W_{Qi}, H_t W_{Ki}, H_t W_{Vi})$, where $W_Q \in \mathbb{R}^{d \times d/k}$, $W_K \in \mathbb{R}^{d \times d/k}$, and $W_V \in \mathbb{R}^{d \times d/k}$ are learned parameter matrices. The output of the multi-head attention is then transformed using the weight matrix $W_o \in \mathbb{R}^{d \times d}$.

3.2.3 Encoder Block Representation

After applying the multi-head self-attention, the UT computes the revised representations H_t by combining the attention output A_t with the previous representation H_{t-1} and the positional encoding and time signal embeddings P_t:

$$A_t = \text{LayerNorm}((H_{t-1} + P_t) + \text{MultiHeadSelfAttention}(H_{t-1} + P_t)) \quad (7)$$

Here, LayerNorm() represents the layer normalization function. Finally, the revised representations are obtained by applying a transition function:

$$H_t = \text{LayerNorm}(A_t + \text{Transition}(A_t)) \quad (8)$$

The transition function Transition() applies non-linear transformations to the attention output A_t and integrates it with the previous representation. The resulting revised representations H_t capture the refined information at step t.

The UT encoder utilizes an iterative computation process, repeating for a total of T steps. This iterative process progressively refines the representations of the input sequence, capturing intricate dependencies. To determine the number of steps, the Universal Transformer employs the Adaptive Computation Time (ACT) mechanism. After undergoing T steps, where each step updates all positions of the input sequence simultaneously, the final output of the Universal Transformer encoder is a matrix $H^T \in \mathbb{R}^{m \times d}$. This matrix consists of d-dimensional vector representations for the m tokens present in the input sequence.

By considering the hidden representation obtained after T iterations, we obtain the contextual

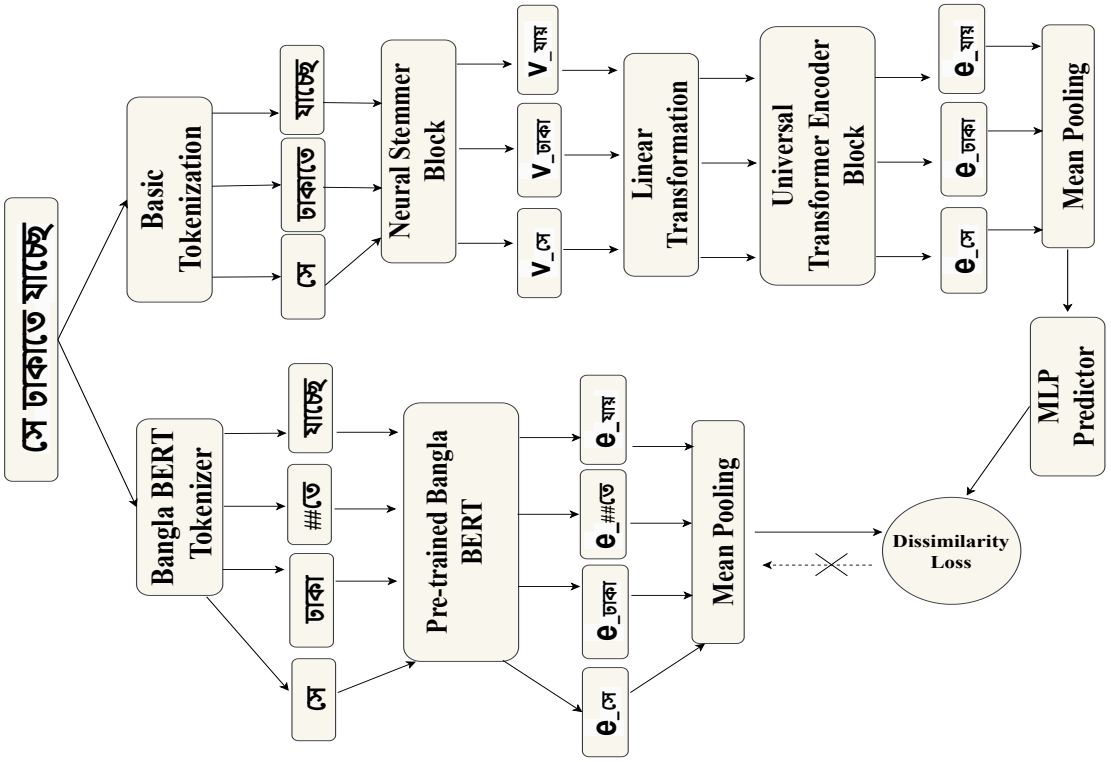


Figure 2: Model Architecture of Contextual Neural Stemmer Block

embeddings of the word tokens. Denoting the embeddings as $E = [e_1, e_2, \ldots, e_m]$, we can equivalently express E as $E = H^T$. Therefore, the matrix H^T represents the contextual embeddings for the sentence S.

3.3 Mean Pooling

Let $E = [e_1, e_2, \ldots, e_m]$ be the input sequence S with embedding $e_i \in \mathbb{R}^d$ that we get from the universal transformer encoder block. The sequence may contain padded values for equal length. Define the mask vector $M = [\text{mask}_1, \text{mask}_2, \ldots, \text{mask}_m]$ to indicate valid tokens. $\text{mask}_i = 1$ for valid tokens and 0 for padded values. The masked mean pooling operation is:

$$\text{MeanPooling}(X, M) = \frac{1}{\sum_{i=1}^{n} \text{mask}_i} \sum_{i=1}^{n} \text{mask}_i \cdot e_i \quad (9)$$

After applying mean pooling to the sentence S, the sentence-level representation e_S from the UT encoder are obtained. An MLP (Multi Layer Percptron) is applied to the e_S to get the final sentence level representation e_S.

3.4 Stemming Loss

We utilize a loss function called *Stemming Loss* in the Neural Stemmer Block, as described in Section 3.1. The main objective of this loss is that the character based representations for the word tokens should be similar with their word2vec representation of their stemmed words. Given a sentence $S = [x_1, x_2, \ldots, x_m]$, we feed it into the neural stemmer block, which generates character-based representations v_i for each word token x_i in the sentence S. Additionally, each token in sentence S is passed through a rule-based stemmer to obtain the root form, resulting in the stemmed version $S = [r_1, r_2, \ldots, r_m]$ of the sentence.

Subsequently, we input each root word r_i into a pre-trained word2vec model, which produces a static embedding u_i for the word r_i. We didn't train the pre-trained word2vec model during training. To align the predicted embedding v_i with the static embedding u_i, we employ *Cosine Similarity* based loss as follows:

$$\textit{Stemming\_Loss}(u_i, v_i) = 1 - \frac{u_i \cdot v_i}{\|u_i\| \|v_i\|} \quad (10)$$

This loss ensures that the representation from the Neural Stemmer Block should be aligned with the stemming representation from pre-trained

Word2vec. If the word2vec representations of a stemmed words is not found, we simply ignore the word while calculating the stemming loss.

3.5 Dissimilarity Loss

During training our model, we also employ another loss named *Cosine Dissimilarity Loss*. The objective to use the loss is that the contextual embeddings for a sentence from the UT encoder block should be aligned with the pre-trained BERT contextual embeddings for that sentence. To calculate the loss, we also feed our input text into the pre-trained BERT and we get contextual embeddings for that sentences S, $E' = [e'_1, e'_2, \ldots, e'_m]$. We apply Mean Pooling described in Section 3.3 to get the sentence level representation e'_S from the pre-trained BERT for sentence S. The pre-trained BERT model is not trained during the training process.

On the other hand, we also get another sentence level representation e_S from UT encoder as described in Section 3.3. Then, we apply the cosine dissimilarity based loss as follows:

$$\textit{Cosine\_Dissimilarity}(e_S, e'_S) = 1 - \frac{e_S \cdot e'_S}{\|e_S\|\|e'_S\|} \tag{11}$$

3.6 Model Loss

To obtain high-quality contextual representations from our model, we rely on the Stemming Loss (Section 3.4) and cosine dissimilarity (Section 3.5). The cosine dissimilarity is based on the pretrained BERT representations, which face challenges such as the subword problem and out-of-vocabulary (OOV) problem. To ensure effective training of our model, we adopt a guided training schema. In this schema, we prioritize training our model on samples where the BERT tokenizer yields a lower number of OOV and subword tokens. Additionally, we incorporate a penalty score based on BERT tokenization techniques when calculating the final loss. Hence, our final training loss is defined as:

$$\begin{aligned} \textit{Loss} &= \gamma \times \textit{Stemming\_Loss} \\ &+ \left(1 - \frac{a+b}{m}\right) \times \beta \times \textit{Dissimilarity\_Loss} \end{aligned} \tag{12}$$

Her, a represent the number of subword tokens, b denotes the count of unknown ([UNK]) tokens, and m indicate the total number of tokens. The weights γ and β determine the contribution of the stemming loss and cosine dissimilarity loss, respectively, to the main loss.

4 Experimental Setup

4.1 Experimental Design

Our model follows a two-step training process for each experimental dataset. In the first step, we engage in unsupervised training to learn contextual representations between words. The primary objective of this unsupervised training is to transfer the knowledge from the pre-trained BERT model to our contextual neural stemmer. In the second step, we conduct supervised fine-tuning, where we further train our model in a supervised fashion, focusing solely on the classification loss. To prioritize the development of semantic/contextualized representations for stemming words only, rather than building a language model (LM), we opted not to train our model extensively on a large corpus during the unsupervised training phase.

To identify stemming words in Bangla, a rule-based stemmer is employed, utilizing the Bangla Stemmer library. In unsupervised training, we choose Bangla-NLP Toolkit for find finding representations of the stemmed words as Bangla Pre-trained Word2vec. BanglaBERT (Bhattacharjee et al., 2022) model is used as pre-trained BERT algorithm in our model. Bangla-Word2vec provides 100-dimensional vector representations for each word. Consequently, we set the character embedding size to 100. For contextualized embedding, we define an embedding dimension of 768 to align with the 768-dimensional word representations obtained from Bangla BERT. To convert the 100-dimensional vectors to 768 dimensions, we employ a linear transformation block comprising a single linear layer. If we don't have the word2vec representations of a stemmed word, we neglect the word representations while calculating *Stemming Loss*. We evaluated our model's performance in different evaluation metrics like accuracy, macro f1 score and roc-auc. The details can be found about at Appendix B.

4.2 Model Training Setup and Training Scheme

We choose AdamW (Loshchilov and Hutter, 2017) optimizer for our training where $\beta_1 = 0.9$ and $\beta_2 = 0.99$. Character embedding size is 100 dim

Dataset	Experiment	Pretraining Perplexity	Performance Metrics Accuracy	Macro F1	Weighted F1
BanFake News	Rule Based Stemmer	-	88.1	87.2	91.1
	Neural Stemmer	-	90.4	89.6	93.6
	CNS	125.51	93.1	92.2	94.8
Sarcasm Detection	Rule Based Stemmer	-	85.4	45.1	89.9
	Neural Stemmer	-	87.8	46.0	91.4
	CNS	117.47	90.3	48.7	95.6
SentNoB	Rule Based Stemmer	-	64.5	60.8	64.1
	Neural Stemmer	-	69.2	62.3	68.8
	CNS	134.97	73.3	68.3	72.2
Emotion Detection	Rule Based Stemmer	-	62.5	35.4	61.2
	Neural Stemmer	-	64.6	39.7	62.1
	CNS	87.38	68.4	40.26	64.41
Sentiment Classification	Rule Based Stemmer	-	48.5	31.7	32.4
	Neural Stemmer	-	50.1	32.2	32.8
	CNS	103.49	52.3	34.5	35.9

Table 3: Experimental result for CNS in 5 different dataset. In every dataset, 3 different experiments along with CNS are done, Rule Based Stemmer, Neural Stemmer, and CNS. In every dataset, our CNS method outperform rule based stemmer with a good margin. Here, *CNS* means the *Contextual Neural Stemmer*

and contextual word representations has 768 dim as described in Section 4.1. We use a learning rate of $2 * 10^{-5}$ for unsupervised training and 10^{-3} for supervised finetuning. A LSTM (Hochreiter and Schmidhuber, 1997) based decoder with dropout of 0.1 while finetuning the model. A batch size of 32 is used for unsupervised training and 16 for supervised finetuning. We also experiment with the different combinations of γ and β and found that $\gamma = 0.7$ and $\beta = 0.5$ gives better performance most of the cases. All the experiments run with Python (version 3.8) and Pytorch with free NVIDIA Tesla K80 GPU in Google Colab and single Nvidia Tesla P100 GPU provided by Kaggle. The training time for both unsupervised and supervised varies but on average it takes around 6 mins on average for training one epoch in unsupervised training and 4 mins in supervised training.

5 Result and Discussion

5.1 Effects in Different NLP Datasets

To measure the performance of our model, we consider five different Bangla dataset. The dataset tasks and information are listed in the Appendix A. In every dataset, we run three different experiment.

- **Rule Based Stemmer**: In this experiment, we use a rule based stemmer to find the stemmed word of a word in a sentence. We consider the stemmed words as the tokens of a sentence. Finding the embeddings of the tokens a single LSTM layer is used to find contextual representations. We consider last lstm cell output as sentence representation and passed it into MLP for classification.

- **Neural Stemmer**: Instead of rule based stemmed word, we use Neural Stemmer Block described in Section 3.1. After finding neural stemming representations, we passed them into a single LSTM layer and MLP layers for classifcation as same as Rule Based Stemmer.

- **CNS**: CNS stands for Contextual Neural Stemmer which is our proposed model as described in Figure 2. We use last MLP layer representations for classification. In every dataset, we first pretrained our model in un-

supervised fashion and then finetune it using classification loss.

Table 3 shows the experimental result in five different dataset. In every dataset Neural Stemmer slightly outperforms the rule based one. This is because, sometimes the rule-based stemmer results in the stemmed words which may have no meaning. In this case, character based word representations improves model performance. In every dataset, our method surpass both rule based and neural stemmer based approaches by a good margin (around 2 - 7% improvement on average in different metrics). The reason behind this is, the stemming word (either rule based or neural model based) losses the contextual and semantic information like tense, expression, grammatical context which are very useful for a model to find a good representations. CNS captures those information along with the stemmed word and that's why our model surpassed the other methods.

Dataset Name	Average Cosine
BanFake News	0.7014
Sarcasm Detection	0.7862
SentNoB	0.6776
Emotion Detection	0.7569
Sentiment Classification	0.7980

Table 4: Average Cosine Similarity from CNS Model in Test Sentences between Word2vec of Stemming word and Word Representations from Neural Stemmer Layer.

5.2 Preserving the Stemming Words in Neural Stemmer

We also further investigate on how much stemming information are captured by our model. To find this, we consider the test dataset in aforementioned datasets. We find the pretrained Word2vec presentations of the word in text sentences. We find average cosine similarity between those stemmed word's pretrained word2vec representations and the representations from Neural Stemmer layer. The results are reported in Table 4. From this table we can see our model is able to capture stemming information. By tuning γ we can control how much stemming information should be captured by our model.

Dataset Name	Average Cosine
BanFake News	0.6572
Sarcasm Detection	0.7284
SentNoB	0.6397
Emotion Detection	0.7128
Sentiment Classification	0.7329

Table 5: Average Cosine Similarity from CNS Model in Test Sentences between Token Representations of Pre-Trained Bangla BERT and Contextual Word Representations from UT Encoder.

5.3 How Contextualized the Contextual Neural Stemmer

We were also interested in experimenting how much contextual information is capturing like BERT. For doing this we reported two experiment. For the first one, we average cosine similairties between the word representations of a sentence of pretrained Bangla-BERT in the test samples and the Universal Transformer (UT) encoder representations from CNS. The results are in Table 5. Another experiment is done on the sentence level representations. We consider mean of the word representations of pretrained model as sentence level representations and measure a cosine similarities with MLP representations from CNS in Tabale 6. From this experiment, we can see that our CNS model is also able to capture contextual information.

Dataset Name	Average Cosine
BanFake News	0.9563
Sarcasm Detection	0.9790
SentNoB	0.9227
Emotion Detection	0.9673
Sentiment Classification	0.9872

Table 6: Average Cosine Similarity of Sentences in Test Sentences between Mean Pooling Output from Pretrained Bangla BERT Representations and Last Layer MLP Representations from CNS.

6 Conclusion

In this research, we proposed a Contextual Bangla Neural Stemmer to overcome the limitations of traditional rule-based stemmers. By obtaining vector representations for both stem words and unknown vocabulary words, our method offers improved word representations for Bangla language processing tasks. The model leverages the Uni-

versal Transformer encoder and Mean Pooling to capture contextual word and sentence-level representations. Our evaluation on five Bangla datasets demonstrated significant performance gains, outperforming the vanilla approach. Notably, our approach focuses on root word detection and addressing OOV and sub-word problems rather than retraining the BERT.

Our findings suggest that a large corpus-based language model incorporating our methodology could further enhance NLP tasks and potentially improve explainability. By addressing the limitations of stemmers and providing better word representations, our proposed approach opens new avenues for research in Bangla language processing and contributes to advancing natural language understanding in the context of Bangla text.

Limitations

As we mentioned above, the proposed method works well against the stemming method but it can't beat the finetuning BanglaBERT. (The performance of BanglaBERT is reported in Appendix C.) The reason behind this BanglaBERT is a language model which was trained on huge corpus. As our method isn't trained on the huge corpus so our model can't beat the BanglaBERT. If we trained our proposed model in a huge corpus, it may be possible to beat BanglaBERT.

Acknowledgements

This project has been jointly sponsored by Independent University, Bangladesh and the ICT Division of the Bangladesh Government.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Abhisek Chakrabarty, Akshay Chaturvedi, and Utpal Garain. 2016. A neural lemmatizer for bengali. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2558–2561.

Abhisek Chakrabarty and Utpal Garain. 2016. Benlem (a bengali lemmatizer) and its role in wsd. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 15(3):1–18.

Abhisek Chakrabarty, Onkar Arun Pandit, and Utpal Garain. 2017. Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1481–1491.

Souvick Das, Rajat Pandit, and Sudip Kumar Naskar. 2020. A rule based lightweight bengali stemmer. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 400–408.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2018. Universal transformers. *arXiv preprint arXiv:1807.03819*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. BanFakeNews: A dataset for detecting fake news in Bangla. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2862–2871, Marseille, France. European Language Resources Association.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Md Ashraful Islam, Md Towhiduzzaman, Md Tauhidul Islam Bhuiyan, Abdullah Al Maruf, and Jesan Ahammed Ovi. 2022. Banel: An encoder-decoder based bangla neural lemmatizer. *SN Applied Sciences*, 4(5):138.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Md Redowan Mahmud, Mahbuba Afrin, Md Abdur Razzaque, Ellis Miller, and Joel Iwashige. 2014. A rule based bengali stemmer. In *2014 international conference on advances in computing, communications and informatics (ICACCI)*, pages 2750–2756. IEEE.

Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. 2020. Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5, Online. Association for Computational Linguistics.

KM Tahsin Hassan Rahit, Khandaker Tabin Hasan, Md Al-Amin, and Zahiduddin Ahmed. 2018. Banglanet: Towards a wordnet for bengali language. In *Proceedings of the 9th Global Wordnet Conference*, pages 1–9.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(4):1–21.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6. IEEE.

A Dataset Description

- **BanFake News** - (Hossain et al., 2020) introduces a dataset for detecting fake news. The dataset is consisted of 48K authentic and 1k fake news articles of different category. The tasks is classification tasks to find if a news is fake or not.

- **Sarcasm Detection** - This is Kaggle Compeition Dataset [1] where the organizer curated a dataset comprised of around 50K news headlines labeled in two categories: Sarcastic (1) or Not-Sarcastic (0).

- **SentNoB** - In SentNoB(Islam et al., 2021), public comments on news and videos were collected from social media for detecting the sentiment. The sentiment were labeled as Positive, Negative and Neutral. The training dataset size is 13.5K where validation and test dataset size is 1.5K

- **Emotion Detection** - (Tripto and Ali, 2018) collected user emotion dataset from YouTube user comments. The emotion detection dataset has 5 types of emotion: anger/disgust, joy, sadness, fear/surprise, and none.

- **Sentiment Classification** - (Tripto and Ali, 2018) also find the sentiment of the comments in the pervious dataset. We use five class sentiment dataset in this case. The sentiment were labeled as Strongly Positive ,Positive, Strongly Negative, Negative and Neutral.

B Evaluation Metrics

In our experiment, we calculate Perplexity Score (PPL Score) for evaluation the model performance. It measures how well a probability distribution or language model predicts a given sample.

$$\text{Perplexity} = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 p(x_i)} \quad (13)$$

Here, N represents the number of samples, and $p(x_i)$ is the probability assigned by the language model to the i-th sample x_i. A lower perplexity indicates better predictive performance, as the model can more accurately predict the given samples.

For the downstream tasks, we trace down Accuracy, F1 Score and ROC-AUC Score. The ROC-AUC metric measures the ability of a model to distinguish between positive and negative classes based on the area under the receiver operating characteristic curve.

C BanglaBERT Baseline

For most of the dataset the performance of BanglaBERT isn't reported. For each dataset that mentioned above we finetuned BanglaBERT. The baseline result for BanglaBERT is reported below:

Model Name	Acc ↑	Macro F1 ↑
BanFake	96.65	92.99
Sarcasm Detection	93.30	49.00
SentNoB	74.46	69.55
Emotion Detection	70.78	41.26
Sentiment Analysis	54.11	42.59

Table 7: BanglaBERT baseline performance after fine-tuning it on afermentioned datasets.

[1] https://www.kaggle.com/competitions/nlp-competition-cuet-ete-day-2022/data

Investigating the Effectiveness of Graph-based Algorithm for Bangla Text Classification

Farhan Noor Dehan, Md Fahim, Amin Ahsan Ali,
M Ashraful Amin, A K M Mahbubur Rahman
Center for Computational & Data Sciences
Independent University, Bangladesh
Dhaka-1229, Bangladesh
{1920269, md.fahim, aminali, aminmdashraful, akmmrahman}@iub.edu.bd

Abstract

In this study, we examine and analyze the behavior of several graph-based models for Bangla text classification tasks. Graph-based algorithms create heterogeneous graphs from text data. Each node represents either a word or a document and each edge indicates the relationship between any two words or word to document. We applied the BERT and different graph-based models including TextGCN, GAT, BertGAT, and BertGCN on five different Bangla text datasets including SentNoB, Sarcasm detection, BanFakeNews, Hate speech detection, and Emotion detection datasets. The performance with the BERT model surpassed the TextGCN and the GAT models by a large difference in terms of accuracy, Macro F1 score, and weighted F1 score. On the other hand, BertGCN and BertGAT outperformed the standalone graph models and the BERT. BertGAT excelled in the Emotion detection dataset and achieved a 1%-2% performance boost in Sarcasm detection, Hate speech detection, and BanFakeNews datasets from BERT's performance. Whereas BertGCN outperformed BertGAT by 1% for SentNoB and BanFakeNews datasets while beating BertGAT by 2% for Sarcasm detection, Hate Speech, and Emotion detection datasets. Furthermore, We examined different variations in graph structure and analyzed their effects.

1 Introduction

Natural language processing (NLP) has become very significant in recent years and text classification is one of the most crucial tasks in this domain. Text classification is the process of classifying text based on specific labels utilized in document categorization. It has applications in many diverse problems, including hate speech detection, spam detection, sentiment analysis, topic modeling, question answering, intent recognition, medical text analysis, legal document classification, social media analysis, fake news detection, and many more.(Zhou et al., 2020; Dwivedi and Arya, 2016; Patel and Mistry, 2015)

Graph algorithms can capture complex relationships and dependencies in various text structures and represent better semantic and syntactic relationships. Thus graph algorithms help create a more accurate understanding and interpretation of the text. (Wang et al., 2023) Moreover, these models can identify the grammatical relationship and contextual information between words and documents. By enabling these models to understand how the meaning of words changes based on context, for example, BertGCN performed better than Bidirectional Encoder Representations (BERT) (Devlin et al., 2018) and RoBERTa on different English datasets because of its contextual understanding. Graph-based models (Lin et al., 2021) can also identify sentiment-related relationships between words in sentiment analysis and can produce more accurate predictions based on sentiments. Furthermore, graph algorithms (Liang et al., 2022) can capture user interactions, mentions, and relationships during social media text analysis, hate speech detection, sentiment analysis, influence identification, and community recognition tasks. (Patel and Mistry, 2015; Akhter et al., 2018)

In recent times, extensive research has been conducted on Bangla text using different machine learning and deep learning models (Farhan et al., 2023; Bitto et al., 2023; Sadat Aothoi et al., 2023). However, graph-based Bangla text classification has remained largely unexplored. On the contrary, numerous graph-based models and structures have been implemented worldwide, particularly in English. The research works (Do et al., 2022; Zhang et al., 2023; Wu et al., 2023) motivate us to conduct our study on how different graph-based models with state-of-art models perform on Bangla datasets used for various tasks BERT. In this study, the previous best-performing

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 104–116

models for each dataset were also observed alongside the implementation of various graph-based methods. Graph methods include Graph Convolutional Networks for text (TextGCN) (Yao et al., 2019), Graph Attention Networks(GAT) (Velickovic et al., 2017), BertGCN (Lin et al., 2021), and BertGAT (Lin et al., 2021). These algorithms were applied to five datasets containing SentNoB, Sarcasm detection, BanFakeNews, Hate speech detection, and Emotion detection datasets.

In this study, BERT outperformed the previously leading models and demonstrated significantly superior performance compared to GCN and GAT across all the datasets. BertGAT surpassed BERT's performance by 1%-5% for all datasets except SentNoB. However, BertGCN achieved better performance by 1%-2% over BertGAT for all datasets, which was attributed to BertGCN's superior ability to comprehend local contextual and global semantic relationships (Lin et al., 2021). As part of the ablation study, using BERT embeddings as node features showed better performance for GAT and GCN, while one-hot embeddings performed better for the integrated models. Two types of edge sets: document-word only (d2w) and document**to**word + wordtoword (d2w+w2w) were utilized for all the graph-based models. The edge set d2w+w2w outperformed all the models by 1%-2%. BertGCN outperformed all other models in this study. This research also aimed to find the right balance between graph models and BERT. BertGCN exhibited the highest performance when λ ranged from 0.3 to 0.7. BertGAT showed better performance when λ ranged from 0.1 to 0.5. Ultimately, this study was conducted to demonstrate that graph-based models can outperform traditional models for Bangla text classification and pave the way for future research in this domain.

Our main contributions are:

- We compare different graph methods for text classification and compare them with BERT. This study might be the first to compare graph-based models for Bangla text classification.

- We analyze results with several benchmarks of the datasets. We perform result analysis, ablation study, and identify the best graph-based models for Bangla text classification.

2 Related Work

Text classification is one of the classical problems for NLP. Naive Bayes, SVM, and other orthodox approaches for text classifications faced challenges in effectively learning meaningful text representations. Addressing these constraints, the application of deep learning models such as Convolutional Neural Networks (CNNs) (Kim, 2014) and Recurrent Neural Networks (RNNs) (Sherstinsky, 2020), materialized. These models demonstrated the ability to capture intricate features from text data.

BERT from Transformers achieves superior performance on sentence-level and token-level tasks (Devlin et al., 2018). Global structure refers to the information of the whole document, and graph-based models utilize an adjacency matrix to capture this information. To address the constraint of BERT models, researchers explored the usage of Graph Neural Networks (GNNs) and graph embeddings. Some of the most used GNNs are Graph convolutional network (GCN)(Kipf and Welling, 2016), GAT, Graph Sample and Aggregated (Hamilton et al., 2017), and, MoNet (Thekumparampil et al., 2018). One such model used for text classification is the TextGCN. TextGCN focuses on global word co-occurrence information. However, TextGCN has significant drawbacks, including a small receptive field, a lack of edge characteristics, over-smoothing, and an inability to accommodate varying neighborhoods. GAT overcame these restrictions by utilizing self-attentional layers. The limitations associated with pretraining in GCN and GAT are noteworthy. To address this, an innovative approach known as BertGCN has been developed, strategically amalgamating the advantages of BERT and GCN. BertGCN has the power of Large-scale pretraining on enormous unrefined data. In addition, by spreading label influence through graph convolution, transductive learning concurrently learns representations for training data and unlabeled test data.(Lin et al., 2021)

In recent years, there has been a significant amount of research conducted on Bangla text classification because of its importance. Inverse class frequency along with TF-IDF (Dhar et al., 2018) was proposed for Bangla text classification. Machine learning algorithms such as Naive Bayes, J48, KNN, and SVM were also used for Bangla texts (Akhter et al., 2018; Chy et al., 2014). Alam

and Islam (2018) used Logistic Regression, SVM, LIBLINEAR, and Neural Networks for a massive volume of data with 300k samples in datasets. Transformer was used on six different datasets in Bangla by Alam et al. (2020). Ahmed et al. (2022); Rahman and Chakraborty (2021) implemented deep learning RNN attention layer and RNN with BiLSTM. Furthermore, the BERT and ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) models were tested by Rahman et al. (2020). However, graph-based models are very rarely used in Bangla text classification, which motivated our investigation to check how graph-based algorithms perform on different Bangla text datasets.

3 Methodology

Graph-based models use graphs as the representation of textual information. These models build their graphs based on specific criteria. Usually, in these text-derived graphs, individual words and entire documents serve as nodes, while connections (edges) represent co-occurrences, semantic similarity, grammatical dependencies, or any other relevant relationship among the nodes.

3.1 Building Graph from Text

The construction of the graph serves as a pivotal precursor, setting the stage for essential operations like message passing and aggregation in the graph-based framework. Textual data is transformed into graph structures, establishing an organized format that graph-based models can directly process. Graph-based models produce embedding vectors for nodes by considering the characteristics of their neighboring nodes within the graph. Similar to figure 1, the graph-building process starts by preprocessing the textual data. Then, unique words and the entire document are converted into a set of nodes V for each document, while every node is assumed to be connected to other nodes and itself (creating a self-loop), meaning that for any node v there exists an edge $(v, v) \in E$ in the graph. The edges connect word nodes belonging to the same document and that document's document node. The connections (edges) E represent any relevant relationship between the nodes. Formally, a graph is denoted as $G = (V, E)$, where $V(|V| = N)$ represents the set of N number of nodes and E represents the set of edges within the graph.

Next, the graph-based models take a few things from the created graph. The models consider nodes with their associated features, while the relationships or edges between nodes are captured by the adjacency matrix A. Adjacency matrices represent relationships between nodes using a binary matrix of size $N \times N$ for the size of N number of nodes. During this time, edge weights (usually co-occurrence for words and TF-IDF for documents) of the graph are calculated. So, formally, the adjacency matrix is,

$$A_{i,j} = \begin{cases} \text{PMI}(i,j), & \text{if } i,j \text{ are words} \\ \text{TF-IDF}(i,j), & \text{if } i \text{ is doc \& } j \text{ is word} \\ 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The co-occurrence between two words is calculated using the PMI value. The PMI value of a word pair i, j is computed as follows:

$$PMI(i,j) = \max(\log \frac{p(i,j)}{p(i)p(j)}, 0) \tag{2}$$

$$p(i,j) = \frac{\#W(i,j)}{\#W} \tag{3}$$

$$p(i) = \frac{\#W(i)}{\#W} \tag{4}$$

$$p(j) = \frac{\#W(j)}{\#W} \tag{5}$$

Where $\#W(i)$ is the number of sliding windows in a corpus that contains word i, $\#W(j)$ is the number of sliding windows in a corpus that contains word j, $\#W(i,j)$ is the number of sliding windows that contain both word i and j, and $\#W$ is the total number of sliding windows in the corpus. For the BertGCN and BertGAT, the adjacency matrix is created almost similarly, and the only change is positive pointwise mutual information (PPMI) for word co-occurrence.

3.2 GCN

For TextGCN, an identity matrix $X = I_{n_{\text{doc}}+n_{\text{word}}}$ is the initial node feature, where n_{doc} is the number of document nodes and n_{word} is the number of word nodes. Once the graph construction is complete, the initial input is then introduced to the primary GCN layer. Subsequently, this input undergoes the ReLU activation function. The outcome

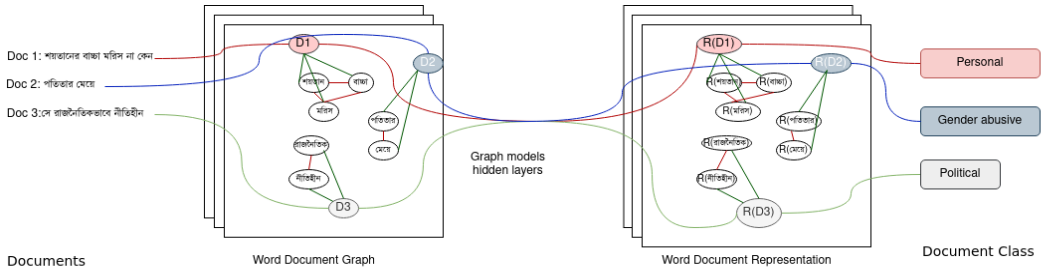


Figure 1: The process of text classification using Graph-based models

of this activation function is then directed to an additional GCN layer to ascertain logits. These logits are then transferred to the softmax function for classification, similar to the approach followed by the GCN model in the work by (Kipf and Welling, 2016),

$$Z = softmax\left(\tilde{A}ReLU\left(\tilde{A}XW_0\right)W_1\right) \quad (6)$$

Where $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix using degree matrix D, where $D_{ii} = \sum_j A_{ij}$ (i,j represent row and column of A,respectively) and W_0 and W_1 are weight parameters that are trained via gradient descent.(Yao et al., 2019) The loss function is defined as the cross-entropy error over all labeled documents,

$$\mathcal{L} = -\sum_{d\in\mathcal{Y}_D}\sum_{f=1}^{F} Y_{df}\ln Z_{df} \quad (7)$$

Where d is the variable that iterates over the set of document indices Y_D, Y_D is the set of document indices that have labels, F is the dimension of the output features, which is equal to the number of classes, Y is the label indicator matrix, Y_{df} is the element of the label indicator matrix Y at row d, and column f, and Z_{df} is the predicted probabilities or scores produced by the model for the document d being in class f.

3.3 GAT

The GAT model primarily operates on a collection of node features as its input.(Velickovic et al., 2017) The node features are expressed as:

$$\mathbf{h} = \left\{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\right\}, \vec{h}_i \in R^F \quad (8)$$

Where $\vec{h}_i$ is the node features of i^{th} node ($i \in N$), h is the set of node features, N is the number of nodes, and F is the number of features in each node. The model produces a new set of node features (of potentially different cardinality, F') as its output,

$$\mathbf{h}' = \left\{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\right\}, \vec{h}'_i \in R^{F'} \quad (9)$$

Firstly, to calculate the final representation in GAT, a weight matrix W, is applied to every initial node. After that, a shared attentional mechanism computes attention coefficients,

$$e_{ij} = a\left(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j\right) \quad (10)$$

Here, a represents the self-attention of the nodes, and the importance of node j's features to node i is calculated. Then, coefficients are normalized by using a softmax function,

$$\alpha_{ij} = \text{softmax}_j\left(e_{ij}\right) = \frac{\exp\left(e_{ij}\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(e_{ik}\right)} \quad (11)$$

Here, k is a neighboring node of i from its neighborhood N_i. Finally, The attention coefficient α_{ij} for node i and its neighbor j is calculated using,

$$\alpha_{ij} = \sigma\left(LeakyReLU\left(\mathbf{a}^T\left[Wh_i \| Wh_j\right]\right)\right) \quad (12)$$

Where W is a learnable weight matrix, h_i and h_j are the feature vectors of nodes i and j, respectively, $\mathbf{a}$ is a learnable attention vector, LeakyReLU is the leaky rectified linear unit activation function, $\|$ denotes concatenation, ^{T} represents transposition and σ refers to the sigmoid function.

The feature representation of node i is updated by aggregating the features of its neighbors, weighted by the attention coefficients,

$$h_i^{(l+1)} = \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(l)} \cdot Wh_j^{(l)} \quad (13)$$

Where $h_i^{(l+1)}$ is the updated feature vector of node i in layer $l+1$, $\alpha_{ij}^{(l)}$ is the attention coefficient between nodes i and j in layer l, $h_j^{(l)}$ is the feature vector of node j in layer l, and $N(i)$ represents the neighbors of node i. Where each final representation,

$$\vec{h}_i' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right) \quad (14)$$

Here, K depicts independent attention mechanisms (heads). An output layer in GAT models may translate the final node or graph representations into the required output format, such as class probabilities for classification tasks.

3.4 BERT-GCN

The BERT model generates the document embeddings for BertGCN.(Lin et al., 2021) The initial node feature matrix,

$$X = \begin{pmatrix} X_{\text{doc}} \\ 0 \end{pmatrix}_{(n_{\text{doc}}+n_{\text{word}})\times d} \quad (15)$$

Here, $X_{\text{doc}} \in R^{n_{\text{doc}}\times d}$ is the document node embeddings and d is the embedding dimension. X is the input of the GCN model and the output feature matrix of the i-th GCN layer,

$$L^{(i)} = \rho\left(\tilde{A}L^{(i-1)}W^{(i)}\right) \quad (16)$$

Where $L^{(i-1)}$ is the output feature matrix of the $i-1$ layer, ρ is an activation function, $\tilde{A}$ is the normalized adjacency matrix, and $W^{(i)} \in R^{d_{i-1}\times d_i}$ is a weight matrix of the layer i. Then, the outputs of GCN are fed into the softmax layer Z_{GCN}, similar to 3.2 for classification.

In practice, improving BertGCN with an auxiliary classifier that directly acts on BERT embeddings leads to rapid convergence and improved outcomes. An auxiliary classifier is created by directly feeding document embeddings (denoted by X) to a dense layer activated using softmax:

$$Z_{\text{BERT}} = \textit{softmax}(WX) \quad (17)$$

The final training objective is the linear interpolation of the prediction from BertGCN and the prediction from BERT, which is given by:

$$Z = \lambda * Z_{\text{GCN}} + (1-\lambda) * Z_{\text{BERT}} \quad (18)$$

Where λ is the balance between BertGCN model and BERT module.

3.5 BERT-GAT

BertGAT(Lin et al., 2021) follows the structure and implementation process specified in Section 3.4. Notably, it is fundamentally comparable to BertGCN, comprising similar methodologies from initial node embedding to final output generation. The primary difference is that it uses the GAT model, as described in Section 3.3, rather than the GCN model, for its underlying graph modeling architecture. In BertGAT, λ is also used to control the influence of GAT and BERT.

4 Experiment Setup

4.1 Dataset

- **BanFake News** - Hossain et al. (2020) introduces a dataset for detecting fake news. The dataset consists of 48K authentic and 1K fake news articles from different categories. The tasks are classification tasks to find out if news is fake or not.

- **Sarcasm Detection** - This is Kaggle Competition Dataset [1] where the organizer curated a dataset comprised of around 50K news headlines labeled in two categories: Sarcastic (1) or Not-Sarcastic (0).

- **HateSpeech Detection** - Dataset provided by Karim et al. (2020) has raw texts collected from different sources with around 3k samples for training and 1k samples for testing. This dataset categorized into political, personal, gender-abusive, geopolitical, and religious hates

- **SentNoB** - In SentNoB(Islam et al., 2021), public comments on news and videos were collected from social media to detect the sentiment. The sentiments were labeled as Positive, Negative, and Neutral. The training dataset size is 13.5K, whereas the validation and test dataset size are 1.5K.

- **Emotion Detection** - For the emotion detection task, we use an emotion detection dataset provided by Trinto and Ali (2018) on which Bengali text data were extracted from

[1] https://www.kaggle.com/competitions/nlp-competition-cuet-ete-day-2022/data

YouTube comments in different kinds of Bengali videos. The emotion dataset contains around 3k samples and 5 classes representing different emotions such as anger/disgust, fear/surprise, joy, sadness, and none.

4.2 Preprocessing & Setup

In this experiment, we preprocessed the datasets by removing the number, URL, other language symbols or words, punctuation, and emojis. Five different models were used in this experiment including four graph-based models. BERT, BertGAT, and BertGCN Model used *csebuetnlp/banglabert* base (Bhattacharjee et al., 2022). BERT model contains hidden dimension 768, learning rate 1×10^{-5}, batch-size 16, and the maximum length of each sequence considered was 128. For the GCN model used in this experiment, the layers considered were 3, hidden dimension 200, drop-out rate 0.5, and learning rate considered 1×10^{-3}. GAT was also used in this experiment and includes 8 heads, learning rate 1×10^{-3}, batch size 64, hidden dimension 200, and epochs used 200. On the other hand, the BERT model in BertGCN and BertGAT includes max length for inputs 128, batch size 128, the learning rate of 1×10^{-5}, and 60 epochs. GCN model integrated into BertGCN includes layers 3, hidden dimension 200, drop-out rate 0.5, and learning rate considered 1×10^{-3}. Finally, the GAT model combined with BERT in BertGAT contains 8 heads, learning rate 1×10^{-3}, batch size 64, hidden dimension 200, and epochs used 200. Different ablation studies were done to find the right graph structures for GCN and GAT in Table 3 and 4. This experiment was done using Python 3.10 and experimented on Google Colab with NVIDIA Tesla $T4$ GPU and Kaggle with a single NVIDIA Tesla $P100$ GPU. To evaluate the model performance we use Accuracy, Macro-F1 Score & Weighted F1 score matrices in this experiment.

4.2.1 Computational Efficiency Analysis

Graph-based algorithms impose substantial computational demands. The feasibility of deploying these models at scale relies on several critical factors, encompassing model complexity, graph dimensions, and scalability considerations. Increasing the complexity of BertGCN models, especially through the enlargement of BERT embeddings, inevitably mandates a significant allocation of computational resources. The size of the graph serves as a pivotal determinant influencing computational efficiency. Addressing very large graphs poses significant challenges due to heightened computational and memory requirements, potentially leading to scalability issues. It's worth noting that graph-based models require a longer computation time (more than 2 to 5 times) compared to the BERT model.

4.2.2 Assessing Past Leading Models

In Table 1, an overview of the performance of previous state-of-the-art models is provided. These outcomes offer valuable insights and establish a foundation for performance bench marking.

Dataset	**Performance**	
	Model	Macro F1
SentNoB	n-gram fusion	**64.61**
Sarcasm	BERT	**89.93**
HateSpeech	SVM	**60.78**
BanFakeNews	SVM	**91.00**
Emotion	LSTM	**59.23**

Table 1: Performance of Previous Leading Models

5 Result & Discussion

In this section, we measured performance metrics across all five models for each of the five distinct datasets, facilitating a comprehensive comparative analysis. We evaluated edge features, where we assessed the effects of d2w-only relationships exclusively, as well as the d2w+w2w relationships. Subsequently, we turned our attention to a thorough investigation into the utilization of both one-hot embeddings and BERT embeddings, with a focus on how these variations influenced the overall model performance. Finally, we embarked on the quest to identify the optimal values for the parameter λ, particularly within the BertGCN and BertGAT models. Our pursuit aimed to unravel the intricacies of their behavior and performance under varying λ values.

5.1 Performance Analysis of Graph NLP Models

In the study, table 2 represents performance matrices of different models and datasets. Models include BanglaBert which is generally used for Bangla language classification tasks. BanglaBert is compared with various graph-based models including TextGCN, GAT, BanglaBertGAT, and BanglaBertGCN. Different datasets including

Dataset	Model	Performance Metrics		
		Accuracy	Macro F1 Score	Weighted F1
SentNoB	BanglaBERT	74.46	69.55	73.03
	GCN	41.60	29.66	33.69
	GAT	42.03	33.98	36.97
	BanglaBERT-GAT	74.65	70.65	74.65
	BanglaBERT-GCN	**75.66**	**71.70**	**74.72**
Sarcasm Detection	BanglaBERT	93.30	49.00	98.31
	GCN	61.40	44.22	50.91
	GAT	77.59	44.69	87.38
	BanglaBERT-GAT	95.85	48.94	97.88
	BanglaBERT-GCN	**98.22**	**49.55**	**99.10**
HateSpeech Detection	BanglaBERT	69.33	41.65	65.41
	GCN	44.07	14.57	30.87
	GAT	47.37	25.63	42.58
	BanglaBERT-GAT	71.44	57.05	70.32
	BanglaBERT-GCN	**73.33**	**60.80**	**72.81**
BanFakeNews	BanglaBERT	96.65	92.99	96.51
	GCN	84.60	75.12	77.54
	GAT	87.10	77.16	78.09
	BanglaBERT-GAT	97.14	92.91	97.02
	BanglaBERT-GCN	**98.55**	**96.69**	**98.55**
Emotion Detection	BanglaBERT	70.78	41.26	65.52
	GCN	46.68	14.79	34.97
	GAT	47.29	16.96	35.42
	BanglaBERT-GAT	75.30	45.67	71.63
	BanglaBERT-GCN	**76.81**	**46.70**	**72.67**

Table 2: Performance of Graph base NLP Model in Different Bangla Text Classificaiton Dataset

SentNoB, Sarcasm detection, hate speech detection, BanFakeNews, Emotion detection, and sentiment analysis were used for text classification using these models. In table 2, BERT's accuracy, macro F1 score, and weighted F1 score are very superior to TextGCN and GAT models for all the datasets. BERT shows this excellence due to a strong contextual understanding of text and pretraining on a large number of data. GCN and GAT perform well when the data is a graph. However, these models are not able to properly understand and represent local contextual information. Thus, they didn't perform well in the classifying task of the datasets. GAT was better compared to TextGCN.

GAT was able to use attention mechanisms to identify the importance of neighbors. This enables GAT to capture complex relationships and local sequences better than GCN. Bangla BertGAT outperformed BERT by 1% to 5% for Sarcasm Detection, Hate speech detection, Emotion detection, and Sentiment analysis datasets. For the SentNoB dataset, BertGAT shows a very slight improvement over BERT. The reason is BertGAT's attention mechanism of GAT and Bangla BERT's pretraining. Finally, BanglaBertGCN bested all the models for the datasets. BanglaBertGCN outperformed BanglaBertGAT and gained superior results by 1% to 3% for all the datasets. Bangla BertGCN captures local contextual information as well as the global relationship among all the words and documents. Which accounts for its greater accuracy,

Figure 2: Attention heatmap for all models of the sentence

macro F1 score, and weighted F1 score.

Figure 2 represents each model's focus on individual word tokens from text "শালা লুচ্চা দেখতে পাঠার মত" for hate speech detection. The tokens 0: '[CLS]', 1: 'শালা', 2: 'লু', 3:'##চ্চা', 4:'দেখতে', 5:'পাঠা', 6:'##র', 7:'মত', 8: '[SEP]' are generated. For graph models, the word tokens were also considered as nodes. The BertGCN model provided more attention scores on the hate words (highlighted by deeper colors).

5.2 Edge Features Effect in Graph based Models

Table 3 depicts the effect of different edge feature structures on the graph-based models. Two types of edge features were evaluated: (1) d2w only, which is the edges created from only the word and document edges, and (2) d2w+w2w, which contains edges from word and document relationships as well as word and word relationships. The effects are measured in terms of accuracy. The d2w+w2w edge features showed better performance than the d2w-only structure for all the datasets in all four graph models.

Word-word edge structure creates a similar semantic cluster of similar words, providing more information about the context.(Han et al., 2022) Thus, an edge set containing a d2w+w2w structure captures more contextual information from a text, providing it with a greater performance. BertGCN bested all the models for both edge feature structures. BertGAT slightly lags behind BertGCN in terms of performance.

5.3 Node Features Effect in Graph based Models

One hot embedding is usually used to determine node features for graph-based models. BERT embedding was also used in this study to compare with One hot embedding. BERT embedding is learned during pretraining. In this study, the evaluation of test accuracy for One-hot embedding and BERT embeddings were used as initial node features on five datasets for four different Graph models. Specifically, In table 4 comparison between test accuracy against One hot embedding and BERT embeddings for the SentNoB dataset is shown. Firstly, TextGCN and GAT models give better results with BERT embedding than one-hot embedding. This may be because sentiment analysis tasks gain better leverage from the broader semantics knowledge learned from an extensive external text.(Han et al., 2022)

Model Name	Edge Features	
	d2w only	d2w+w2w
TextGCN	41.31	**41.60**
GAT	41.55	**42.03**
BERT-GAT	74.46	**74.65**
BERT-GCN	74.84	**75.66**

Table 3: Comparing different SentNoB Edge Features architecture

Model Name	Node Features	
	One Hot	BERT
TextGCN	41.60	**41.98**
GAT	42.03	**56.18**
BERT-GAT	**74.65**	58.13
BERT-GCN	**75.66**	62.55

Table 4: Comparing different SentNob Node Features architecture

Moreover, GCN and GAT aren't able to capture local semantic features with one hot embedding. BERT provides local attention as well as identifies long-term dependencies in a text.(Devlin et al., 2018) Thus, BERT features to improve the overall performance of GCN and GAT. Secondly, BertGAT and BertGCN show better performance in one hot embedding than BERT embedding. This finding can be ascribed to the hypothesis that providing BERT embedding results in additional redundancies and complexity. Thus, resulting in poor performance when compared with one hot embedding. Finally, in this research endeavor edge

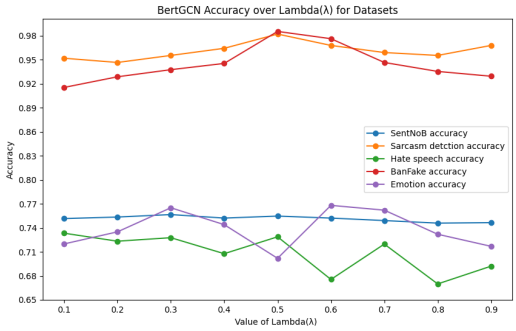


Figure 3: Effects of λ in BanglaBERT-GCN over accuracy for all datasets

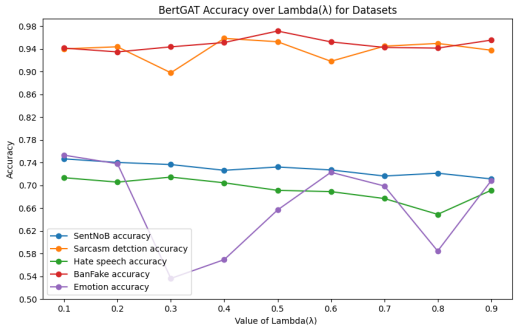


Figure 4: Effects of λ in BanglaBERT-GAT over accuracy for all datasets

sets (d2w+w2w) were used because they gave the best result in a full environment.

5.4 Effects of Lambda(λ)

The values of lambda λ have a significant effect on the overall performance of both BertGCN and BertGAT models. λ has values from 0 to 1. BertGCN/BertGAT's final output is determined by a linear interpolation of predictions from BERT and BertGCN (or BertGAT), which is defined by the parameter (λ). The value of λ varies for different tasks. When λ is set to 1, it represents sole dependence on BertGCN or BertGAT, whereas a λ value of 0 represents exclusive reliance on the BERT model.

In this study, the λ values were measured against accuracy to determine the effects it has on all five different datasets. In figure 3 and figure 4, the curves represent the influence of λ on each dataset for BertGCN and BertGAT models. Each of the curves contains accuracy for λ value from 0.1 to 0.9. Usually, very high or very low values of λ are ignored. Because it removes a significant portion of either BERT or GCN (or GAT) influence from the BertGCN (or BertGAT) structure. In figure 4 we also examined similar phenomena for the BERTGAT model. It is observed in the study that the highest accuracy values can be observed for λ values from 0.3-0.7. This is observed because of the balanced performance of graph-based and BERT methods. However, when the BertGAT model was considered, the picture was slightly different. In figure 4, maximum accuracy was obtained for BertGAT throughout a range of λ values ranging from 0.1 to 0.5. While the SentNoB, Hate Speech, and Emotion datasets performed best at $\lambda = 0.1$. The overall behavior can be explained by the high performance of the BERT method in the integrated structure.

6 Conclusion

This study evaluates various graph-based models for Bangla text classification and assesses their performance. TextGCN and GAT exhibit comparatively lower performance when compared to BERT. However, the integration of these models with BERT yields superior results in comparison to other models for classification tasks. BertGCN incorporates BERT's large-scale pretraining and fine-tuning, enhanced by transductive learning. BertGCN and BertGAT exhibit improved comprehension of local semantics through their integration with BERT. We advocate for the adoption of graph-based models, particularly BertGCN and BertGAT, for Bangla text classification, given their comparatively heightened predictive accuracy when contrasted with traditional text classification models.

In conclusion, it's noteworthy that the domain of Bangla text remains relatively unexplored in the context of graph-based algorithms and concepts. Numerous unexplored avenues including knowledge graphs and alternative graph models beyond GCN and GAT demand further exploration. However, it's essential to acknowledge that graph-based models entail significant computational resources, leading us to consider these avenues for future research endeavors.

Limitations

Graph models exhibit certain limitations that should be considered in academic research.

First, their scalability is often constrained, as handling large-scale graphs can be computationally intensive. Additionally, these models may struggle with sparse or incomplete data, impacting their accuracy in real-world scenarios. Interpretability can be challenging, making it hard to discern the rationale behind their predictions. Moreover, graph models may not effectively capture temporal dynamics, limiting their applicability in time-dependent problems. Lastly, they may require significant domain-specific expertise for effective deployment, posing a barrier to their widespread adoption in diverse fields.

Acknowledgements

This project has been jointly sponsored by Independent University, Bangladesh and the ICT Division of the Bangladesh Government.

References

Mostaq Ahmed, Partha Chakraborty, and Tanupriya Choudhury. 2022. Bangla document categorization using deep rnn model with attention mechanism. In Cyber Intelligence and Information Retrieval: Proceedings of CIIR 2021, pages 137--147. Springer.

Shahin Akhter et al. 2018. Social media bullying detection using machine learning on bangla text. In 2018 10th International Conference on Electrical and Computer Engineering (ICECE), pages 385--388. IEEE.

Md Tanvir Alam and Md Mofijul Islam. 2018. Bard: Bangla article classification using a new comprehensive dataset. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pages 1--5. IEEE.

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Bangla text classification using transformers. arXiv preprint arXiv:2011.04446.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 1318--1327, Seattle, United States. Association for Computational Linguistics.

Abu Kowshir Bitto, Md Hasan Imam Bijoy, Md Shohel Arman, Imran Mahmud, Aka Das, and Joy Majumder. 2023. Sentiment analysis from bangladeshi food delivery startup based on user reviews using machine learning and deep learning. Bulletin of Electrical Engineering and Informatics, 12(4):2282--2291.

Abu Nowshed Chy, Md Hanif Seddiqui, and Sowmitra Das. 2014. Bangla news classification using naive bayes classifier. In 16th Int'l Conf. Computer and Information Technology, pages 366--371. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ankita Dhar, Niladri Sekhar Dash, and Kaushik Roy. 2018. Classification of bangla text documents based on inverse class frequency. In 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), pages 1--6. IEEE.

Phuc Do, Trung Phan, Hung Le, and Brij B Gupta. 2022. Building a knowledge graph by using cross-lingual transfer method and distributed minie algorithm on apache spark. Neural Computing and Applications, pages 1--17.

Sanjay K Dwivedi and Chandrakala Arya. 2016. Automatic text classification in information retrieval: A survey. In Proceedings of the second international conference on information and communication technology for competitive strategies, pages 1--6.

Niloy Farhan, Ishrat Tasnim Awishi, Md Humaion Kabir Mehedi, MD Mustakin Alam, and Annajiat Alim Rasel. 2023. Ensemble of gated recurrent unit and convolutional neural network for sarcasm detection in bangla. In 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), pages 0624--0629. IEEE.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. Advances in neural information processing systems, 30.

Soyeon Caren Han, Zihan Yuan, Kunze Wang, Siqu Long, and Josiah Poon. 2022. Understanding graph convolutional networks for text classification. arXiv preprint arXiv:2203.16060.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. Banfakenews: A dataset for detecting fake news in bangla. arXiv preprint arXiv:2004.08789.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. Sentnob: A dataset for analysing sentiment on noisy bangla texts. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3265--3271.

Md. Rezaul Karim, Bharathi Raja Chakravarthi, Mihael Arcan, John P. McCrae, and Michael Cochez. 2020. Classification benchmarks for

under-resourced bengali language based on multichannel convolutional-lstm network. 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pages 390--399.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Bin Liang, Hang Su, Lin Gui, Erik Cambria, and Ruifeng Xu. 2022. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. Knowledge-Based Systems, 235:107643.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. Bertgcn: Transductive text classification by combining gcn and bert. arXiv preprint arXiv:2105.05727.

Priyanka Patel and Khushali Mistry. 2015. A review: Text classification on social media data. IOSR Journal of Computer Engineering, 17(1):80--84.

Md Mahbubur Rahman, Md Aktaruzzaman Pramanik, Rifat Sadik, Monikrishna Roy, and Partha Chakraborty. 2020. Bangla documents classification using transformer based deep learning models. In 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), pages 1--5. IEEE.

Saifur Rahman and Partha Chakraborty. 2021. Bangla document classification using deep recurrent neural network with bilstm. In Proceedings of International Conference on Machine Intelligence and Data Science Applications: MIDAS 2020, pages 507--519. Springer.

Mehzabin Sadat Aothoi, Samin Ahsan, Najeefa Nikhat Choudhury, and Annajiat Alim Rasel. 2023. Supervised hybrid model for rumor classification: A comparative study of machine and deep learning approaches. In International Conference on Big Data Analytics and Knowledge Discovery, pages 281--286. Springer.

Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena, 404:132306.

Kiran K Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. 2018. Attention-based graph neural network for semi-supervised learning. arXiv preprint arXiv:1803.03735.

Nafis Irtiza Trinto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pages 1--6.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. stat, 1050(20):10--48550.

Kunze Wang, Yihao Ding, and Soyeon Caren Han. 2023. Graph neural networks for text classification: A survey. arXiv preprint arXiv:2304.11534.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long, et al. 2023. Graph neural networks for natural language processing: A survey. Foundations and Trends in Machine Learning, 16(2):119--328.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In Proceedings of the AAAI conference on artificial intelligence, 01, pages 7370--7377.

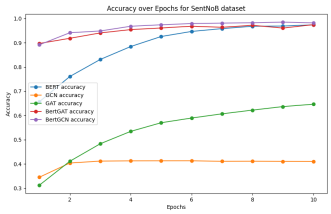
Yazhou Zhang, Dan Ma, Prayag Tiwari, Chen Zhang, Mehedi Masud, Mohammad Shorfuzzaman, and Dawei Song. 2023. Stance-level sarcasm detection with bert and stance-centered graph attention networks. ACM Transactions on Internet Technology, 23(2):1--21.

Xujuan Zhou, Raj Gururajan, Yuefeng Li, Revathi Venkataraman, Xiaohui Tao, Ghazal Bargshady, Prabal D Barua, and Srinivas Kondalsamy-Chennakesavan. 2020. A survey on text classification and its applications. In Web Intelligence, 3, pages 205--216. IOS Press.

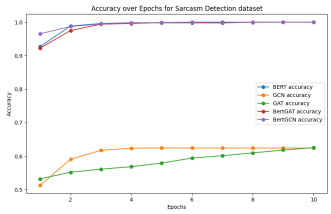
A Accuracy & Loss Plots

A.1 Accuracy Plots

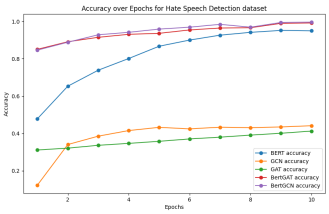
Figure 5 illustrates the training accuracy versus epochs for all the different datasets. Initially, the training accuracy for these datasets applying all the models ranges from 0.1-0.9. GCN and GAT show very low training accuracies compared to all the models for all the epochs. GAT shows better performance due to its attention mechanism.



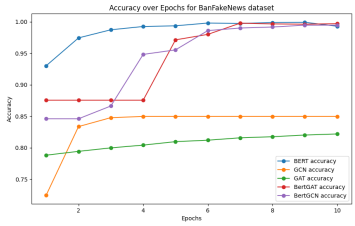
SentNOB Accuracy Plot



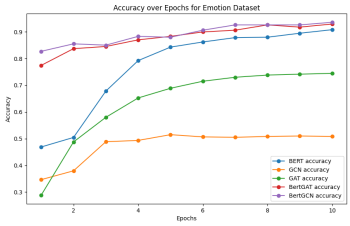
Sarcasm Accuracy Plot



Hate Accuracy Plot



BanFake Accuracy Plot



Emotion Accuracy Plot

Figure 5: Accuracy Plots of Different Models in Aforementioned Datasets

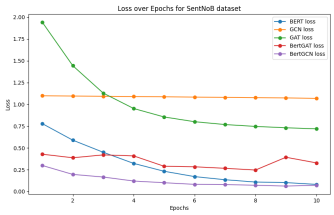
BERT, BertGAT, and BertGCN show the highest accuracies. BertGCN and BertGAT outperform the BERT model. One of the reasons for this outperformance is the fine-tuning of the BERT model before integrating it with the graph-based model. In the fine-tuning phase, the model adapts to the specific characteristics of each dataset. In this figure, a consistent increase in accuracy can be seen as training continues until the highest training accuracy is achieved. It is crucial to highlight that beyond this point, there is a risk of over-fitting.

A.2 Loss Plots

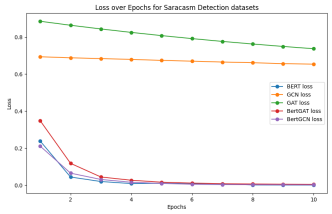
In figure 6, training loss versus epochs for each dataset was observed. The curves represent different models used in this study. The training loss starts from a very high value initially. With the increase of epochs, the training loss sharply decreases.

Then, the losses decrease and become steady until it reaches a point where the training accuracy reaches its maximum value. If the training continues overfitting may occur resulting in a large gap between training loss and test loss. For GAT and GCN, the loss curves are consistently situated higher on the graph across all datasets.

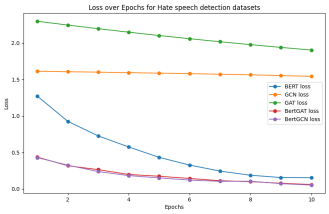
The training loss for all the datasets for GCN as well as Sarcasm, BanFake, and Hate Speech for GAT is not decreasing significantly, showing that the training loss may be not learning effectively from the data. For SentNOB, BanFake, and Emotion datasets, GAT's loss curves decrease below those of GCN, while Sarcasm detection and Hate speech datasets exhibit the opposite behavior. Particularly, for the SentNOB dataset, the BERT loss decreases below BertGAT, and BertGCN's loss curve decreases below BERT's.



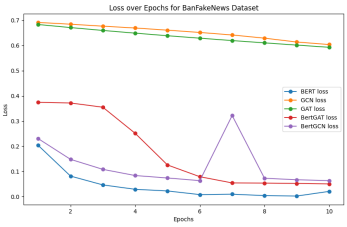
SentNOB Loss Plot



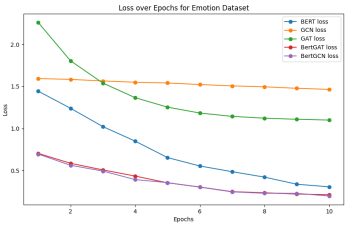
Sarcasm Loss Plot



Hate Loss Plot



BanFake Loss Plot



Emotion Loss Plot

Figure 6: Loss Plots of Different Models in Aforementioned Datasets

On the other hand, in the Emotion and Hate Speech datasets, the loss curves decrease in the following order: BERT, BertGAT, and BertGCN. Finally, Sarcasm and BanFake datasets, BertGAT and BertGCN both exhibit decreased losses compared to BERT. But, overall BERT, BertGAT, and BertGCN show a steep fall in the initial epochs suggesting that the model is learning quickly. However, as training progresses, the pace of decline may drop, suggesting that the model is approaching an ideal answer.

SynthNID: Synthetic Data to Improve End-to-end Bangla Document Key Information Extraction

Syed Mostofa Monsur*, **Shariar Kabir***, **Sakib Chowdhury***
Celloscope Ltd.
{mostofa.monsur,shariar.kabir,sakib.chowdhury}@cellosco.pe

Abstract

End-to-end Document Key Information Extraction models require a lot of compute and labeled data to perform well on real datasets. This is particularly challenging for low-resource languages like Bangla where domain-specific multimodal document datasets are scarcely available. In this paper, we have introduced *SynthNID*, a system to generate domain-specific document image data for training OCR-less end-to-end Key Information Extraction systems. We show the generated data improves the performance of the extraction model on real datasets and the system is easily extendable to generate other types of scanned documents for a wide range of document understanding tasks. The code for generating synthetic data is available at https://github.com/dv66/synthnid

1 Introduction

Document Key Information Extraction (KIE) is a very crucial task to extract structured or semi-structured information from printed documents and images (Luo et al., 2023; Shi et al., 2023; Lee et al., 2022). Numerous user-facing applications nowadays require scanning and extracting information as key-value pairs from raw images of invoices, receipts, ID cards etc. Previously, these types of information extraction systems required a mandatory OCR engine and rule-based approaches in the pipeline. However OCR-based systems become error-prone easily because of the lack of contextual understanding (Kim et al., 2022). Also, hand-picked rule-based extraction systems cannot handle all possible transformations and variations that can be found in scanned documents.

Recently, a surge of pre-trained models has been witnessed in the area of document understanding. These models overcome the problems of OCR-powered Key Information Extraction systems by completely removing the need to apply character-level recognition and information aggregation. These models are essentially vision transformers which are pre-trained on huge datasets of scanned documents across multiple languages (Kim et al., 2022). These pre-trained models achieve state-of-the-art in various document understanding tasks on the particular languages they were pre-trained on. It is possible to fine-tune these models for downstream extraction tasks in other languages with sub-optimal performances. In the case of low-resource languages like Bangla, it is a severe issue because there are almost no datasets for the Document Key Information Extraction task in Bangla. Although a number of works are found in the literature regarding Bangla OCR and/or text detection systems (Safir et al., 2021; Hossain et al., 2022; Rabby et al., 2019; Alam et al., 2020), none of them perform end-to-end Key Information Retrieval which is essential to retrieve necessary fields from the scanned document.

Collecting annotated data for Document Key Information Extraction is also quite expensive and time-consuming. To reduce labelling effort and cost of labelling, synthetic data is often used alongside real data to train models for various Document Understanding tasks (Gupta et al., 2016). Unfortunately, general-purpose synthetic image generators do not focus on Key Information Extraction only but on general-purpose document understanding tasks. Most of the state-of-the-art generators support only English or rich-resource corpus thus low-resource languages like Bangla are completely ignored. Also, the lack of availability of high-quality Bangla corpus for Key Information Extraction tasks is another reason for the absence of end-to-end models in this area. The end-to-end Key Information Extraction model requires huge datasets with millions of samples (Kim et al., 2022) and hundreds of GPU hours which also contributes to this issue. One option is to fine-tune the pre-trained multi-

* Equal contribution

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 117–123

lingual end-to-end models on the target language e.g. Bangla to perform Key Information Extraction on real documents like the National ID card of Bangladesh, License Plates etc. This approach is particularly useful when the whole document image is used as an information source and the target output is a structured data format like JSON. Because then an extra field extraction or linking stage is no longer required to add to the extraction pipeline.

In this work, we propose a system *SynthNID*, to generate domain-specific synthetic data which improves the performance of end-to-end document key information extraction tasks when fine-tuned alongside real data. Our primary focus for this work was to extract key values from the National ID Card of Bangladesh which contains a mixture of English and Bangla text in the document but using our approach a wide range of scanned documents can also be generated for Key Information Extraction tasks. We demonstrate the effectiveness of the generated data by fine-tuning end-to-end Key Information Extraction models. Our synthetic data increases the model's performance in extracting key-value pairs from real datasets.

2 Related Work

Document Key Information Extraction is a widely studied task in the literature. Most popular models incorporate the output of an OCR engine and learn to parse them from scanned documents (Hwang et al., 2021b,a). (Hwang et al., 2019) and (Majumder et al., 2020) have applied Document Key Information Extraction to various real-world applications. Most of these approaches introduce a learning framework where the text is detected separately using an off-the-shelf OCR engine and a sequence model takes the input from the previous stage considering the text content and locality of the information. Despite the convenience of end-to-end models for KIE tasks, only a few are available in the literature like OCR-free document understanding transformer (Kim et al., 2022). This model takes the whole document image as an input and applies a visual attention mechanism to learn the output sequence which is essentially a key-value structure like JSON. However, the end-to-end model variants are only pre-trained in Chinese, Japanese, Korean and English.

Although there are a number of works present in literature regarding Bangla OCR and a few in Document Understanding, almost none of them address end-to-end Key Information Extraction on Bangla scanned documents or multilingual scanned documents where Bangla is present. *bbOCR* is a scalable document OCR that employs a Bangla text recognition model using synthetic datasets (Zulkarnain et al., 2023). *BaDLAD* is a large multi-domain Bangla Document Layout Analysis dataset which contains more than 33k manually labeled documents from a wide range of sources including books, magazines, newspapers etc. (Shihab et al., 2023). (Ataullha et al., 2023) improves Document Layout Analysis performance leveraging Mask R-CNN architectures. They show competitive results in segmenting Bangla Documents.

Most of the existing works in Bangla Document Understanding have tackled problems like text extraction or layout analysis by segmenting the image components, whereas none of them considered extracting key information in a structured format which can be easily used by independent user-facing applications. In this effort, we have addressed this gap in the existing literature and aimed to solve the issue by new approaches to generate Bangla synthetic documents for domain-specific KIE tasks.

3 Datasets

3.1 Synthetic Dataset Generation

Our synthetic data generation system depends on named entities and random background images. We have collected a dataset of Bangladeshi Bangla first names, middle names and last names for males and females. We developed an empty layout of the 'overlay' which is proportionally similar to the national ID card of Bangladesh. For this work, we have skipped the image and signature part of the ID card because we are only interested in the text here. On the Bangladeshi national ID card's front side, the person's name, mother's name, father's name, date of birth, and identification number these fields are dynamic. Other texts don't change mostly. Our tool picks random names from a dataset of names, generates proper names, and fills the dynamic slots on the overlay. The name dataset was collected manually by labeling named entities from publicly available Bangla corpus. Identification number and date of birth fields are randomly generated according to their standard formats and inserted. Then we pick a random background image from an image store and put the data-filled overlay on the back-

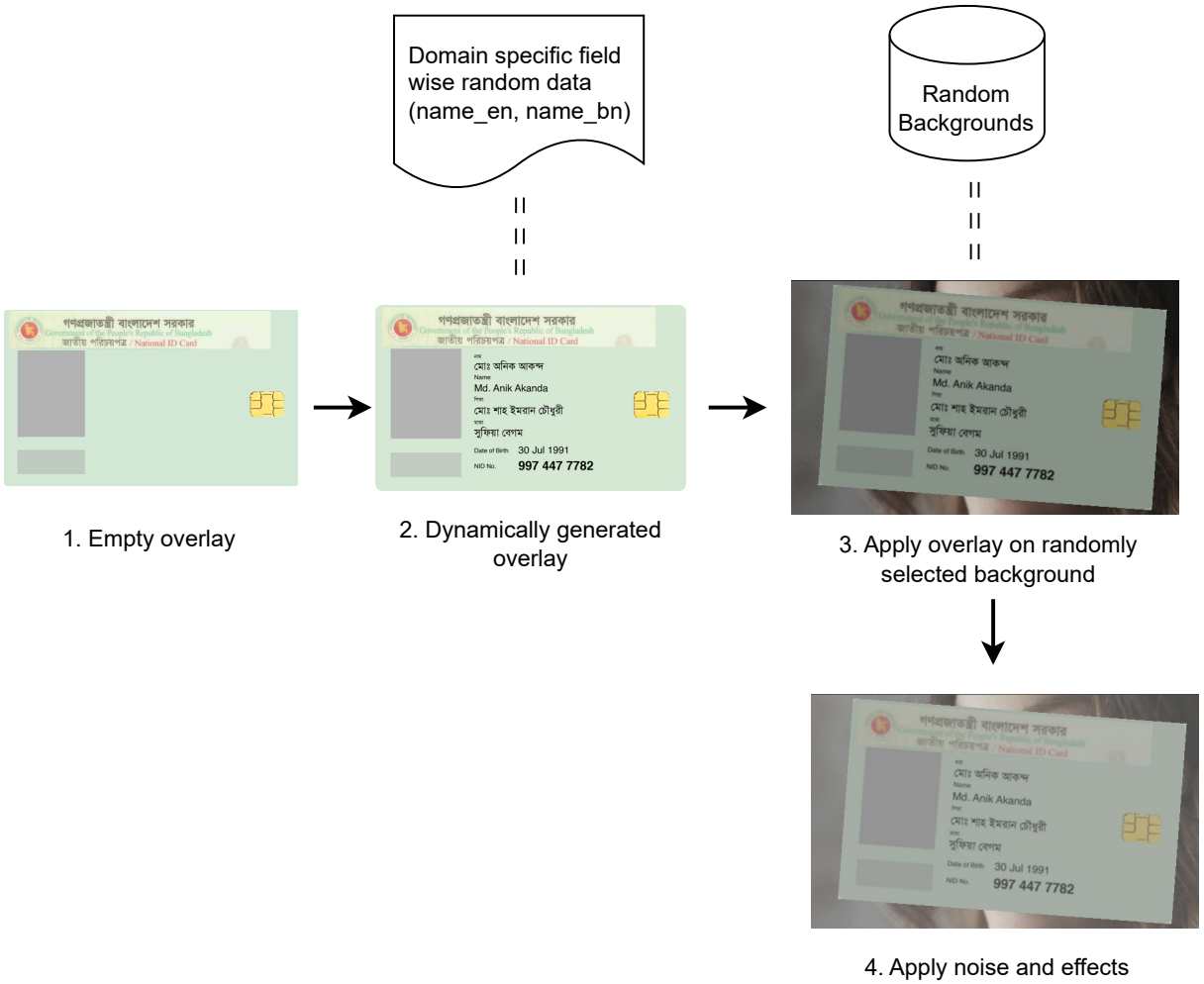


Figure 1: Steps for preparing synthetic NID data for KIE fine-tuning.

ground image. While doing this, we ensured that the overlay covered from 75% to 90% of the background after applying a rotational transformation. Finally, we apply random blur noise effects using (Jung et al., 2020).

With simple customization, this tool can be used to generate a wide range of Bangla-scanned synthetic documents for downstream document understanding tasks.

3.2 Real Dataset

In our experiments, we have used a real dataset of 11,390 images. From this, 10,890 are used for the training and validation and 500 for testing. The end users were provided with a mobile application for data collection. The mobile application allows end users to capture an image using a guiding rectangle box. Users had the option to review the captured image and, if necessary, retake it to ensure data quality. Although most of the real data was of good quality, there were some unavoidable noisy, faded, and rotated or tilted images which made the real data more challenging for the model than synthetic data.

4 Experiments

For our end-to-end model, we used the OCR-free document transformer (Kim et al., 2022) which is the current state-of-the-art in KIE. The model outperforms various other models like *LayoutLM* (Xu et al., 2019), *LayoutLMv2* (Xu et al., 2020), *LayoutXLM* (Xu et al., 2021), *SPADE* (Hwang et al., 2021b), *WYVERN* (Hwang et al., 2020) in document Key Information Extraction tasks. The model is essentially a vision transformer (Dosovitskiy et al., 2020) which is pre-trained on a huge dataset containing real and synthetic data 13M in total. The real dataset IIT-CDIP (Lewis et al., 2006) contains around 11M samples of complex scanned English

Training Dataset	Performance					
	Real Test Data Acc.			Synthetic Test Data Acc.		
	Bangla Fields	English Fields	Overall	Bangla Fields	English Fields	Overall
synth:real-50K:0K	25.96%	31.10%	25.02%	90.71%	94.55%	92.37%
synth:real-0K:10K	76.55%	82.92%	79.28%	80.08%	92.8%	85.91%
synth:real-2K:10K	78.76%	83.95%	81.14%	83.57%	96.54%	89.54%
synth:real-5K:10K	80.56%	83.79%	82.01%	85.58%	98.52%	91.55%
synth:real-10K:10K	**81.53%**	83.73%	82.5%	85.79%	98.79%	91.59%
synth:real-50K:10K	81.35%	**84.6%**	**82.74%**	**89.06%**	**99.39%**	**93.72%**

Table 1: Performance of the models trained on different splits of the dataset in terms of TED

documents. They created a synthetic dataset generation system *SynthDoG* which generates synthetic document samples in Chinese, Japanese, Korean, and English 0.5M per language.

4.1 Fine-tuning Process

We fine-tune the Donut-base model (Kim et al., 2022) for the Key Information Extraction (KIE) task in a mixed scheme where we use splits of both synthetic and real data in our fine-tuned training set. The input resolution is set to 345×575 pixels and the max length in the decoder is set to 100. For the English and Bangla multilingual tokenizer, we use the Banglabert-large model (Bhattacharjee et al., 2022). We finetune the model in early stopping setup for a maximum epoch of 30, using Pytorch-Lightning module (Falcon, 2019) and with one NVIDIA RTX 3070 GPU. We use the Adam (Kingma and Ba, 2014) optimizer, training and validation batch size is set to 512 and 8 respectively and the learning rate is set to 3×10^{-5}. We use a number of 1,000 training samples per epoch. For the evaluation of the models, we use the *tree edit distance* (TED) metric (Zhang and Shasha, 1989), by representing the extracted field values of the NID as a tree.

4.2 Performance on Split Datasets

We evaluated 6 different models trained on different splits of the dataset containing different mixes of the real with synthetic data. Our dataset contains a total of 10,890 (10K) real and 50,000 (50K) synthetic NID images. The real dataset contains more than one images from a user (but in a slightly different orientation). An NID contains 3 Bangla fields (name_bn, father_name and mother_name) and 3 English fields (name_en, dob and nid_no). For establishing a baseline the first two models were trained on only synthetic and real data respectively. For the rest of the 4 models, we used a mix of the 10K real data with different quantities of synthetic data for training. For the evaluation of the models, we used an unseen test set containing 500 synthetic and 500 real data across all the tests.

The results are shown in Table 1. The performance over the Bangla fields (name_bn, father_name and mother_name) and English fields (name_en, dob and nid_no) are shown separately along with the overall performance. For the models' performance over each field please refer to Appendix A. We found that, although the first model trained purely on the 50K synthetic data performs well over the synthetic data it performs poorly over real data. This suggests even with our different approaches to make the synthetic data represent real data the model was not able to learn how to work with real data. In the second model where we used all of the 10K real data with no synthetic data, we see a significant improvement in the performance over real data. However, the performance was poor over the Bangla fields where the second model only achieved an accuracy of 76.55% over the Bangla fields of real data.

We start to see performance improvement in the third case where the model was trained on all of the 10K real data along with 2K synthetic data. In fact, the model outperformed the second model where the train set contained only real data. This proves that a mix of synthetic with real data indeed improves the performance of the model. The performance was more prominent in the case of the Bangla fields where the accuracy improved from 76.55% to 78.76% over real data. The performance improves consistently over the Bangla fields as well as the English fields as more synthetic data is mixed in the train set. Most of the improvement

was found in the extraction of Bangla fields with the addition of synthetic data and we were able to achieve a best of 81.53% accuracy over the Bangla fields of the real data in the fifth model. In the sixth model where all 50K synthetic data is mixed with the 10K real data, we start to see a slight drop in the accuracy of the Bangla field extraction from real data. This suggests a case of diminishing returns in the performance over real data when there is more synthetic data than real data in the mix.

5 Ethical Considerations

While developing our system we prioritized end users' privacy protection. The app was developed and used to collect data inside the organization. Informed consent was obtained from the app users and stringent data anonymization measures were applied while using real data for testing the models' performances. While generating the synthetic data, every field is generated in a completely random strategy. Our ethical framework was aimed to develop high-quality Bangla KIE models while protecting user privacy, maintaining transparency, and ensuring responsible data handling thus strengthening our commitment to conduct ethical AI research.

6 Conclusion

In this paper, we have introduced a scheme to generate high-quality domain-specific synthetic data for the Key Information Extraction task on Bangla scanned documents. We have shown the synthetic data generated using our approach enhances the performance of end-to-end KIE models. In future, we will investigate the areas where effective labelling strategies can be employed to learn good models with a low amount of data using active learning techniques.

References

Samiul Alam, Tahsin Reasat, Asif Shahriyar Sushmit, Sadi Mohammad Siddiquee, Fuad Rahman, Mahady Hasan, and Ahmed Imtiaz Humayun. 2020. A large multi-target dataset of common bengali handwritten graphemes.

Md Ataullha, Mahedi Hassan Rabby, Mushfiqur Rahman, and Tahsina Bintay Azam. 2023. Bengali document layout analysis with detectron2.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale.

William Falcon. 2019. Pytorch lightning.

Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. 2016. Synthetic data for text localisation in natural images.

Md. Ismail Hossain, Mohammed Rakib, Sabbir Mollah, Fuad Rahman, and Nabeel Mohammed. 2022. Lila-boti : Leveraging isolated letter accumulations by ordering teacher insights for bangla handwriting recognition.

Alyssa Hwang, William R. Frey, and Kathleen McKeown. 2020. Towards augmenting lexical resources for slang and African American English. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 160–172, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Wonseok Hwang, Seonghyeon Kim, Minjoon Seo, Jinyeong Yim, Seunghyun Park, Sungrae Park, Junyeop Lee, Bado Lee, and Hwalsuk Lee. 2019. Post-{ocr} parsing: building simple and robust parser via {bio} tagging. In *Workshop on Document Intelligence at NeurIPS 2019*.

Wonseok Hwang, Hyunji Lee, Jinyeong Yim, Geewook Kim, and Minjoon Seo. 2021a. Cost-effective end-to-end information extraction for semi-structured document images. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3375–3383, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. 2021b. Spatial dependency parsing for semi-structured document information extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 330–343, Online. Association for Computational Linguistics.

Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung

Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. 2020. imgaug. https://github.com/aleju/imgaug. Online; accessed 01-Feb-2020.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. 2022. FormNet: Structural encoding beyond sequential modeling in form document information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3735–3754, Dublin, Ireland. Association for Computational Linguistics.

D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 665–666, New York, NY, USA. Association for Computing Machinery.

Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. 2023. Geolayoutlm: Geometric pre-training for visual information extraction.

Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. 2020. Representation learning for information extraction from form-like documents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6495–6504, Online. Association for Computational Linguistics.

AKM Shahariar Azad Rabby, Sadeka Haque, Md. Sanzidul Islam, Sheikh Abujar, and Syed Akhter Hossain. 2019. Ekush: A multipurpose and multitype comprehensive database for online off-line bangla handwritten characters. In *Recent Trends in Image Processing and Pattern Recognition*, pages 149–158, Singapore. Springer Singapore.

Farisa Benta Safir, Abu Quwsar Ohi, M. F. Mridha, Muhammad Mostafa Monowar, and Md. Abdul Hamid. 2021. End-to-end optical character recognition for bengali handwritten words.

Dengliang Shi, Siliang Liu, Jintao Du, and Huijia Zhu. 2023. Layoutgcn: A lightweight architecture for visually rich document understanding. In *Document Analysis and Recognition - ICDAR 2023*, pages 149–165, Cham. Springer Nature Switzerland.

Md. Istiak Hossain Shihab, Md. Rakibul Hasan, Mahfuzur Rahman Emon, Syed Mobassir Hossen, Md. Nazmuddoha Ansary, Intesur Ahmed, Fazle Rabbi Rakib, Shahriar Elahi Dhruvo, Souhardya Saha Dip, Akib Hasan Pavel, Marsia Haque Meghla, Md. Rezwanul Haque, Sayma Sultana Chowdhury, Farig Sadeque, Tahsin Reasat, Ahmed Imtiaz Humayun, and Asif Shahriyar Sushmit. 2023. Badlad: A large multi-domain bengali document layout analysis dataset.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2020. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2019. Layoutlm: Pre-training of text and layout for document image understanding.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding.

K Zhang and D Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262.

Imam Mohammad Zulkarnain, Shayekh Bin Islam, Md. Zami Al Zunaed Farabe, Md. Mehedi Hasan Shawon, Jawaril Munshad Abedin, Beig Rajibul Hasan, Marsia Haque, Istiak Shihab, Syed Mobassir, MD. Nazmuddoha Ansary, Asif Sushmit, and Farig Sadeque. 2023. bbocr: An open-source multi-domain ocr pipeline for bengali documents.

A Appendix

A.1 Models' Performance Variation Across Bangla Fields

Table 2 and 3 shows the performance of the models over the three Bangla fields: name_bn, father_name and mother_name across the real and synthetic test sets respectively. In the NID card of Bangladesh, all of the three fields consist of only Bangla letters. Although the models perform a little worse than English fields across the Bangla field which is due to the absence of Bangla data in the pre-trained base Donut model, we can see a steady increase in performance as more synthetic data is used in the fine-tuning process. The performance improvement was almost equal across the real (Table 2) and synthetic test data (Table 3).

A.2 Models' Performance Variation Across English Fields.

Table 4 and 5 shows the performance of the models over the 3 English fields: name_en, dob and nid_no

Training Dataset	Performance		
	name_bn	father_name	mother_name
synth:real-50K:0K	27.67%	30.03%	28.92%
synth:real-0K:10K	77.23%	72.9%	81.2%
synth:real-2K:10K	78.98%	75.69%	82.7%
synth:real-5K:10K	81.23%	77.2%	84.59%
synth:real-10K:10K	82.02%	79.07%	83.98%
synth:real-50K:10K	81.9%	79.98%	83.55%

Table 2: Performance of the models over Bangla fields in terms of TED across real test data

Training Dataset	Performance		
	name_bn	father_name	mother_name
synth:real-50K:0K	90.63%	90.57%	91.4%
synth:real-0K:10K	82.1%	78.08%	81.05%
synth:real-2K:10K	84.35%	82.54%	84.68%
synth:real-5K:10K	85.57%	84.94%	86.92%
synth:real-10K:10K	85.9%	85.92%	87.86%
synth:real-50K:10K	89.9%	89.31%	90.25%

Table 3: Performance of the models over Bangla fields in terms of TED across synthetic data

across the real and synthetic test sets respectively. In the NID card of Bangladesh, name_en consists of only English letters, dob consists of a mix of English letters and numbers (DD Month YYYY) and

Training Dataset	Performance		
	name_en	dob	nid_no
synth:real-50K:0K	24.33%	52.42%	37.72%
synth:real-0K:10K	78.84%	96.02%	77.87%
synth:real-2K:10K	79.44%	96.67%	79.97%
synth:real-5K:10K	82.11%	96.37%	76.09%
synth:real-10K:10K	81.71%	95.88%	76.75%
synth:real-50K:10K	81.49%	95.85%	79.85%

Table 4: Performance of the models over English fields in terms of TED across real test data

Training Dataset	Performance		
	name_en	dob	nid_no
synth:real-50K:0K	99.41%	99.82%	81.82%
synth:real-0K:10K	89.54%	97.23%	93.09%
synth:real-2K:10K	92.19%	99.93%	99.36%
synth:real-5K:10K	96.74%	99.98%	99.69%
synth:real-10K:10K	97.74%	99.5%	99.73%
synth:real-50K:10K	99.17%	99.45%	99.91%

Table 5: Performance of the models over English fields in terms of TED across synthetic test data

nid_no consists of only English numbers. The base Donut model being pre-trained on English data performs better across the English fields, except for the nid_no field of real test data where the model seems to struggle more than any other fields. Like before we see a steady increase in performance as more synthetic data is used in the fine-tuning process with equal performance improvement across the real (Table 4) and synthetic test data (Table 5).

BaTEClaCor: A Novel Dataset for Bangla Text Error Classification and Correction

Nabilah Tabassum Oshin*, Syed Mohaiminul Hoque*, Md Fahim, Amin Ahsan Ali, M Ashraful Amin, A K M Mahbubur Rahman
Center for Computational & Data Sciences
Independent University, Bangladesh
Dhaka-1229, Bangladesh
{1830668, 1830032, md.fahim,}@iub.edu.bd
{aminali, aminmdashraful, akmmrahman}@iub.edu.bd

Abstract

In the context of the dynamic realm of Bangla communication, online users are often prone to bending the language or making errors due to various factors. We attempt to detect, categorize, and correct those errors by employing several machine learning and deep learning models. To contribute to the preservation and authenticity of the Bangla language, we introduce a meticulously categorized organic dataset encompassing 10,000 authentic Bangla comments from a commonly used social media platform. Through rigorous comparative analysis of distinct models, our study highlights BanglaBERT's superiority in error-category classification and underscores the effectiveness of BanglaT5 for text correction. BanglaBERT achieves accuracy of 79.1% and 74.1% for binary and multiclass error-category classification while the BanglaBERT is fine-tuned and tested with our proposed dataset. Moreover, BanglaT5 achieves the best Rouge-L score (0.8459) when BanglaT5 is fine-tuned and tested with our corrected ground truths. Beyond algorithmic exploration, this endeavor represents a significant stride in enhancing the quality of digital discourse in the Bangla-speaking community, fostering linguistic precision and coherence in online interactions. The dataset and code is available at https://github.com/SyedT1/BaTEClaCor.

1 Introduction

The Bangla language is an Indo-Aryan language with deep historical roots. It is spoken by approximately 230 million people globally and is the 6th most spoken language in the world as stated by the CIA World Factbook [1]. Bangla is renowned for its intricate and unique style, holding cultural and literary significance, and reflecting a rich heritage spanning generations. However, within the contemporary world of communication, particularly on platforms like social media, the fluidity of making typographical errors often results in deviations from the language's original form. So, The complexity of the Bangla script with its 50 letters comprising 11 vowels and 39 consonants is often reflected in the digital landscape .[2]

Among the set of Bangla letters, certain complex characters contribute to the challenge of writing that results in a divergence between written and spoken communication. Phonetically similar alphabets in Bangla share the same pronunciation or phonetic utterance that allows interchangeability and consequently leads to errors within words as shown in Figure 1 (Mittra et al., 2019). For instance, Figure 1 shows the interchange of letters having similar phonetic qualities that generate error words impacting the language's authenticity and coherence (Sifat et al., 2020).

Phonetically Similar Letters	:	"ন" and "ণ" ; "শ" and "স"
Vowel Characters	:	"ি" and "ী" ;"ৌ" and "ৈ"
Consonant Clusters	:	"ঞ্জ" and "জ্ঞ" ; "ম্ভ" and "ম্ব"
Informal Style	:	"খাইতেসি" ; "করতেসিলাম"

Figure 1: Examples of Different types of errors

In the realm of online platforms, such as YouTube and other social media networks, users frequently embrace an informal variant of the Bangla language that is characterized by regional speech patterns and influenced by local dialects or colloquial expressions typical to the residents of the area. This informal variant derived from the original standard Bangla tends to deviate from its roots and originality. This shift can be attributed to the fast-paced and dynamic nature of online communication where brevity, quickness, and informal

* These authors contributed equally to this work.

[1] https://www.cia.gov/the-world-factbook/countries/world/

[2] https://en.wikipedia.org/wiki/Bengali_alphabet

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 124–135
December 7, 2023

expression often take precedence over traditional linguistic norms as an example shown in Figure 1.

Textual error detection and correction of the Bangla language hold significant importance as corrected text preserves language integrity, promotes literacy, and conveys professionalism. Online interactions further underscore the necessity of Bangla text correction as it enables clear communication on a global scale, enhances brand reputation, facilitates cross-cultural communication, and reduces the chances of misinterpretation. Notably, there have been datasets used for similar purposes, predominantly consisting of samples collected from Bangla newspapers, blogs, or synthetically generated.(Mridha et al., 2019) (Sifat et al., 2020). However, they may not fully represent the day-to-day informal and formal interactions of Bangla language speakers on various online platforms where several types of errors can be more prevalent. To address this gap in the existing resources, we introduce a novel dataset for Bangla text error correction named **BaTEClaCor**: A Novel Dataset for **Ba**ngla **T**ext **E**rror **Cla**ssification and **Cor**rection. The dataset is licensed under CC -BY-NC 4.0 (Creative Commons Attribution)

Through a comprehensive approach, this research aligns itself with the larger goal of fostering a digitally literate and linguistically precise digital space for the Bangla community. Our contributions are:

- Introduction of an expansive and authentic dataset comprising 10k of diverse Bangla comments from YouTube videos. The dataset can enhance the generation capability of transformer-based models by providing valuable insights into the informal and regionally influenced Bangla language.
- Performance analysis of several advanced machine learning and deep learning models including BanglaBERT, LSTM, and XLM-RoBERTa to detect errors within Bangla YouTube comments and classify them based on specific error categories while the models are fine-tuned and tested with the proposed dataset.
- Analyzing the performance of BanglaT5 to correct different categories of textual errors including phonetic and grammatical errors while fine-tuning and testing with our proposed dataset.

These contributions enhance the quality of linguistic interactions online and pave the way for a more precise and digitally literate environment for Bangla speakers, fostering meaningful communication, and understanding in the digital realm.

2 Related Work

Numerous endeavors have been undertaken to enhance Bangla text correction despite its status as a low-resource language. Notably, a Bangla spell-checking technique was proposed and tested on a dictionary consisting of pairs of 50,000 correct and incorrect Bangla words. N-gram models were generated for each candidate word. To identify non-word errors, a comprehensive Bangla word dictionary of around 600,000 words was compiled from various online repositories, newspapers, social networking sites, and Bangla blogs (Mittra et al., 2019). The study primarily addresses word-level errors and may lack in encompassing the full spectrum of errors, including contextual and informal errors

H.A.Z. Sameen presented a novel approach for Bangla grammatical error detection using a T5 Transformer model. The training set comprised 9385 sentence pairs, while the testing set included 5,000 test sentences (Shahgir and Sayeed, 2023). It's mentionable that the incorrect sentences in the paired samples were not explicitly categorized to identify specific error types, and instead, errors were indicated using a particular symbol without detailed error categorization.

Chowdhury Rafeed introduced BSpell, a CNN-blended BERT-based Bangla spell checker (Rahman et al., 2022). The synthetic dataset of The Prothom-Alo 2017 online newspaper was used for training, Additionally, 6,300 errorful sentences from Nayadiganta online newspaper were annotated for testing. It's essential to note that the training data's synthetic nature and the usage of newspaper text may not effectively capture the nuances of informal online interactions.

Another method for synthetic error dataset generation was presented using a few sets of popular newspapers mimicking Bangla writing patterns. The study employed a Bangla corpus consisting of 6.5 million sentences. From this corpus, 8,637 frequently occurring words were selected for analysis (Sifat et al., 2020). The study's outcomes revealed the stochastic nature of error generation.

Although these studies collectively contribute

significantly to the advancement of Bangla text correction techniques encompassing various methodologies and datasets, we aimed to address their limitations by constructing a distinct dataset that encompasses the specific error types and reflects the real-world informality prevalent in online communication.

3 Introducing A New Dataset

3.1 Motivation Behind Creation of a New Dataset:

As discussed earlier, existing Bangla datasets used for textual error correction mostly featured samples derived from newspaper articles, blogs and bangla repositories or were synthetically generated. Such sources often portray a formal, official use of the language, which may deviate significantly from its common application in online interactions. Recognizing the need to capture the intricacies of language as it is typically used, we turned to social platforms such as YouTube and Facebook. The driving force behind crafting this new dataset arises from the vital significance of linguistic precision, coupled with the evolving digital environment that defines modern communication particularly among Bangla-speaking internet users of Bangladesh.

3.2 Source of Data Samples:

We selected YouTube as our primary source of sample collection due to its immense popularity in Bangladesh, boasting approximately 34.50 million Bangladeshi users, according to Google's advertising resource. This platform serves as a microcosm of the country's linguistic diversity attracting users from various backgrounds, different levels of literacy, and typing patterns.

To compile this unique dataset, we performed web scraping utilizing YouTube's API on randomly listed videos as shown in Figure 3 and 2 having more than 500k views within August 2023. The random selection minimizes potential bias and ensures a variety of linguistic expressions and errors. Around 60 comments per video were taken to collect ample data for analysis from each video and to provide a balanced dataset size. Selecting comments with three or more words ensures that the dataset contains substantial content for meaningful analysis, and it also minimizes unnecessary padding. This approach optimizes dataset efficiency and is well-suited for machine learning and deep learning models that require fixed-length input sequences.

Algorithm 1 Pseudocode for Comment Scrapping

```
Input: API_KEY = Youtube's API ,
       video_list = ["video_id1", "video_id2", ...]
Output: comments
Initialize comments [ ]
Initialize existing_comments { }
For each video_id in video_list do:
    Retrieve video details from(video_id, API_KEY)
    Extract video title from video details
    Initialize comments_counter = 0
    WHILE comments_counter < 60:
        Retrieve comments
        Preprocess comments
            For each comment do:
                IF comment (Is bengali = True) &&
                (Length_of_comment >= 3 )      &&
                comment NOT IN existing_comments{ }:
                    Append comment TO comments [ ]
                    Add comment TO existing_comments{ }
                    comments_counter += 1
```

Figure 2: Pseudocode for data collection

3.3 Labeling and Annotation:

The labeling and annotations in this dataset were carried out by three of the authors through a careful manual process, ensuring a high level of precision and reliability. The team extensively referred to linguistic references, particularly the authoritative work **Bangla Byakaran O Nirmiti** by Dr. Solaiman Kabir, and the **Bangla Ovidhan** dictionary. These resources played a vital role in guaranteeing the accuracy and linguistic correctness of the dataset, making it a valuable asset for the Bangla language community. A detailed overview of the labeling and annotation procedures is presented in Figure 3.

3.4 Structure and Features of Dataset:

BaTEClaCor dataset aims to serve as a valuable resource for researchers and practitioners seeking to enhance the accuracy and performance of Bangla typing error detection and correction models.

The dataset comprises 10,000 comments, meticulously filtered to include only those written in Bangla letters. Comments containing irrelevant emojis and symbols were discarded, ensuring the dataset's quality and utility. In Table 1, the dataset's composition reflects its comprehensive nature. Of the 10,000 entries, 4224 pertain to incorrect com-

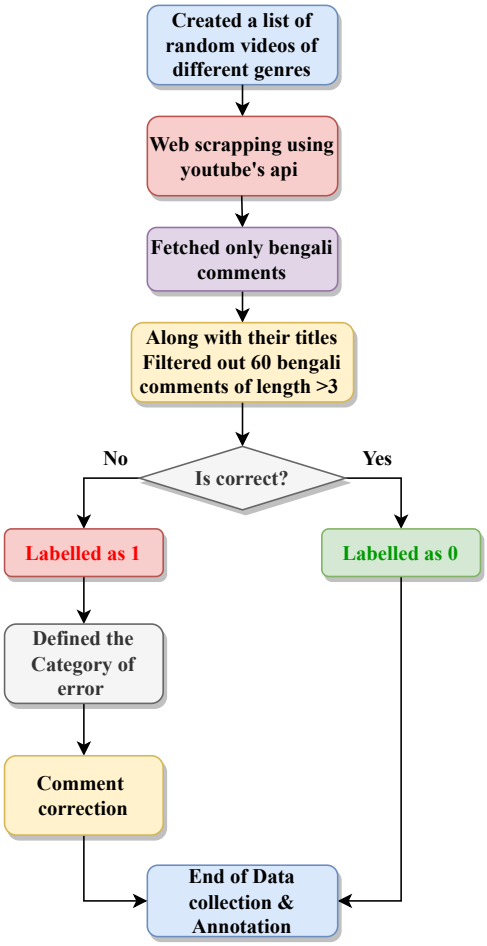


Figure 3: Flowchart of data collection and annotation

ments, while the remaining 5,776 constitute accurate comments. These comments span a diverse array of video genres, including News, Entertainment, Politics, Sports, and Miscellaneous as shown in Table 2.

Label	No. of Comments
0	5776
1	4224

Table 1: Distribution of Labels in the Dataset

Table 2 shows that the selection of video categories for this dataset is carefully orchestrated to encompass a broad spectrum of topics that hold immense significance within the Bangla context. While News, Entertainment, Politics, and Sports constitute the bedrock of societal discourse, allowing individuals to voice their opinions and ideas through comments, the Miscellaneous category transcends conventional boundaries, embracing topics such as Lifestyle, Philosophy, Nature, etc to reflect the diverse interests and passions of Bangladeshi people.

Genre	No. of Comments
Entertainment	3450
News	2009
Miscellaneous	1932
Politics	1885
Sports	771

Table 2: Distribution of Comments by Genre

In Table 3, errors within the dataset are categorized into four distinct and most prevalent types, reflecting the intricate nature of the Bangla script and its potential pitfalls.

- **Spelling**: Spelling being the most commonly occurring category of errors, encompass instances of incorrect spellings.
- **Grammatical**: Grammatical errors denote mistakes related to the structural and syntactical aspects of the Bangla language.
- **Code-Switching** :Code-switching, often referred to as the mixing of English and Bangla within a single comment, a phenomenon known as Banglish. These instances may not constitute conventional text errors in terms of comprehension or meaning. However, their categorization aims to maintain linguistic authenticity by preserving the true essence of the Bangla language, ensuring adherence to standard and widely accepted linguistic norms.
- **Multiple Errors**: Multiple errors encompass comments featuring a combination of error types, such as misspellings alongside code switching or grammatical mistakes intertwined with spelling errors.

Error Category	No. of Comments
Spelling	2502
Code Switching	786
Grammatical	638
Multiple Errors	345

Table 3: Distribution of Comments by Error Category

Each record in the dataset features vital information such as video title, genre, original comment, label, and error category. We can see a sample data in Table 4 which contains a comment with an error under a sports video, more specifically a spelling

error highlighted in red. The comment is corrected precisely and marked where the correction was made in green.

Comment	মাহমুদউল্লাহ রিয়াদ ভাই কে দলে দেকতে চাই
Video Title	সেরা ম্যাচ!!! Bangladesh vs Sri Lanka Sports Talkies
Genre	Sports
Label	Error
Error category	Spelling
Correct Comment	মাহমুদউল্লাহ রিয়াদ ভাই কে দলে দেখতে চাই

Table 4: Sample Data

This novel dataset presents an invaluable contribution to the realm of Bangla NLP. By amalgamating accurate and erroneous comments from diverse genres, our dataset provides a nuanced view of real-world language usage and common typing errors. It serves as a resource that can facilitate the development and fine-tuning of typing error detection models, ultimately improving the linguistic quality and effectiveness of online communication in Bangla.

4 Baseline

4.1 Classification Models

4.1.1 Using ML Models

Initially, TF-IDF segments the text into words. Then TF-IDF scores for each word are calculated and utilized to construct a feature vector.

$$TF - IDF(w, d) = \text{TF}(w, d) \cdot \text{IDF}(w)$$

Here, *TF(w, d)* represents the term frequency of the word w in the dataset d. *IDF(w)* is the inverse document frequency of the word w. The feature vectors x are calculated from the TF-IDF score and then used to train the classifier models. The SVM model makes predictions by finding the class of the hyperplane that is closest to the sample data(Dadgar et al., 2016). Random forest model learns to predict the class of a sample by finding the class with the highest probability utilizing the class label and feature vector(Sjarif et al., 2019). For an input feature vector, the XGBoost model predicts the text's class by selecting the class with the highest predicted value(Qi, 2020).

4.1.2 Using DL Models

LSTM: LSTM processes an input sentence $S = x_1, x_2, ...x_n$ from the dataset of x_i words and passes to an embedding layer to get embedded representations $E = e_1, e_2, ...n$. These are taken as input by LSTM model to find hidden representations $H = h_1, h_2, ...n$. The last layer's hidden representations of LSTM model are passed to a linear layer to perform classification (Hochreiter and Schmidhuber, 1997).

LSTM with Attention: The resulting hidden states h_i from an input sequence processed by LSTM are then used in the Attention mechanism to calculate Attention score α_i.

$$\alpha_i = Softmax(W_{hi} + b)$$

$$c_i = \sum_{j=1}^{n} \alpha_{ij} \hat{h}_j$$

Here, the context vector for each sentence is calculated by taking a weighted sum of the hidden states ($\hat{h}_j$) based on the attention weights (α_{ij}) for each time step (Vaswani et al., 2017). The context vectors c are then used for classification.

CNN-LSTM: In the CNN with LSTM architecture, the input sequence x is first processed by a convolutional neural network (CNN), resulting in feature maps f (Kim, 2014). These feature maps are then used by the LSTM model to calculate hidden states h to be passed into a linear layer for classification.

4.1.3 Using Transformer Models

BanglaBERT and XLM-RoBERTa: These deep-learning transformer models are Pre-trained and further fine-tuned. To obtain a fixed-size representation for an input sentence, we typically use the special [CLS] token representation h_{CLS}.

$$P = Pooling(H)$$

Global pooling is applied to obtain a fixed-size representation P. The final hidden states H from the transformer layers, capture the essence of the input text. This pooled representation is used for classification (Bhattacharjee et al., 2022) (**?**).

4.2 Error Corrector Model

Let X represent the set of input sequences (comments) and Y represent the set of target sequences (corrected forms of the comments). For each input

sequence $x_i \in X$, which is a sequence of tokens, the sequence-to-sequence model f, specifically the T5 base model fine-tuned for error correction, generates an output sequence y_i'. This output sequence y_i' corresponds to the corrected version of the input comment x_i. Mathematically, the task can be defined as follows:

$$y_i' = f(x_i)$$

The primary objective of this task is to train the model f in such a way that it minimizes a suitable loss function (e.g., cross-entropy loss) that quantifies the dissimilarity between the predicted sequence y_i' and the actual target sequence y_i. The training dataset with input comments and their corresponding corrected forms, allows the model to learn the mapping from erroneous comments to their accurate versions(Bhattacharjee et al., 2022).

5 Experimental Design

5.1 Preprocessing & Settings

Our initial focus was on text preprocessing to ensure data quality. For binary classification using machine learning models, we explored text encoding techniques paired with specific classifiers. Such as TF-IDF with Random Forest and with XGBoost with 6000 max features and 100 decision trees. For deep learning models, we investigated LSTM networks on a batch size of 100 with varying configurations and optimization using the Adam optimizer. The LSTM model featured an embedding layer of 6 dimensions. Additionally, we explored LSTM with Attention, utilizing an embedding size of 128. LSTM with CNN with an embedding dimension of 300 including a convolutional layer with 128 filters and a kernel size of 5. In addition, we also explored transformer-based models like XLM-RoBERTa and BanglaBert, employing tokenization with a maximum sequence length of 128. These models were trained with batch sizes of 16.

For multiclass classification, The ML models were applied similarly to binary classification. DL model LSTM was incorporated with an embedding layer with dimensions of 50000x100, an input length of 3000, and an LSTM layer of 100 units operated on a batch size of 64. We also explored LSTM with attention and with CNN employing an LSTM layer of 64 units on a batch size of 16. LSTM with CNN included a convolutional layer with 128 filters and a kernel size of 5. In parallel, the transformer models BanglaBert and XLM-Roberta employed tokenization with a maximum sequence length of 128. Both the models utilized the Adam optimizer with a learning rate of 1×10^{-5} and operated over a batch size of 16.

For error correction, we used two pre-trained models named BanglaT5 and BanglaT5-small respectively, and fine-tuned them in our dataset. The batch size for training and evaluation was set to 16. The learning rate used for training the model was set to 2e-5. The weight-decay parameter helps prevent overfitting which is set to 0.01. We also used fp16 which speeds up training and reduces memory usage while maintaining training stability.

5.2 Evaluation Metrics

To compare the model performance on the predictions, we use the following performance-based metrics:

- **Accuracy**: This metric measures the proportion of correctly classified samples over the total number of samples.
- **Macro Precision**: This metric measures the average of the calculation of precision of each class. It treats all the classes equally regardless of their size or prevalence in the dataset.
- **Macro Recall**: This metric calculates the average of the calculation of the recall for each class.
- **Rouge-1**: This metric calculates the number of overlapping unigrams (single words or tokens) between the generated text and the reference text.
- **Rouge-2**: Rouge-2 calculates the number of overlapping bigrams (two-word sequences) between the generated text and the reference text. Similar to Rouge-1, its score ranges from 0 to 1.
- **Rouge-L**: This metric calculates the length of the longest common subsequence between the generated text and the reference text.

The chosen metrics were selected for their suitability in evaluating text error detection and correction tasks. Accuracy is a fundamental metric for classification tasks, while macro precision and macro recall account for class imbalances. On the other hand, Rouge-1, Rouge-2, and Rouge-L are widely used in assessing the quality of the generated text,

and their usage here reflects the nature of the correction task, aligning closely with real-world applications.

6 Result and Analysis

6.1 Binary Classification

6.1.1 Machine Learning Models:

Table 5 shows that among the ML models we applied, the TF-IDF with XGBoost model demonstrated a slight advantage in both accuracy and macro precision compared to the TF-IDF with SVM model due to the capability to handle non-linear relationships effectively through its ensemble learning approach. In contrast, the TF-IDF approach used by both models tends to exhibit limitations in capturing complex linguistic patterns present in Bangla text. On the other hand, the TF-IDF with Random Forest model displayed slightly inferior results, suggesting its struggle with capturing the intricacies of textual data.

6.1.2 Deep Learning Models:

In DL models, LSTM showed considerable results as shown in as shown in Table 5. When compared to the LSTM model, LSTM with Attention demonstrated a 4% higher accuracy, 2% higher macro precision, and 1% higher macro recall highlighting its ability to capture more complex dependencies in the text. Additionally, the hybrid model, CNN-LSTM, outperformed the LSTM model by almost 6% in accuracy, 3% in macro precision, and 4% in macro recall, showcasing its prowess in identifying patterns in sequences of text.

Originally, some of the models were not subjected to fine-tuning. Subsequently, these models were refined based on specific parameters, leading to enhanced results.

6.1.3 Transformer Models:

BanglaBert demonstrated remarkable accuracy and macro precision, outshining the LSTM model by a significant margin as shown in Table 5. In comparison, XLM - Roberta Base, a versatile multilingual Transformer, delivered competitive results, albeit falling slightly short of BanglaBert's performance. These Transformer models capitalized on their advanced architecture and pre-trained representations to effectively handle the intricacies of Bangla text.

6.2 Multiclass Classification

6.2.1 Machine Learning Models:

For the multiclass classification for error categories, the TF-IDF with SVM model showcased moderate performance with TF-IDF as the feature extraction method and Support Vector Machines (SVM) as the classifier. The TF-IDF with Random Forest model displayed results on par with the SVM model, both sharing the TF-IDF feature extraction approach. Conversely, the TF-IDF with XGBoost model showed a marginal improvement, performing around 1.6% better than the SVM model.

6.2.2 Deep Learning Models:

The DL models displayed varying degrees of proficiency in multiclass classification. Remarkably, LSTM with Attention emerged as the top performer, showcasing a significant 5.3% higher accuracy than the LSTM model. This notable lead can be attributed to the enhanced sequence modeling capabilities of LSTM with Attention. Additionally, CNN + LSTM delivered promising results, outperforming the LSTM model by approximately 3.7% in accuracy. This outcome underscores CNN + LSTM's ability to detect intricate patterns within text sequences, making it a valuable asset for multiclass classification tasks.

6.2.3 Transformer Models:

Once again, BanglaBert emerged as the best performer, showcasing a notable 9.5% higher accuracy compared to the LSTM model. This substantial lead can be attributed to BanglaBert's deep learning architecture and its prowess in capturing complex linguistic patterns and semantic meanings, which are crucial for multiclass classification tasks. While XLM - Roberta Base followed closely, performing around 3.7% better than the LSTM model, it still trailed BanglaBert in accuracy Table 5.

During the sample collection process, we encountered a relatively lower number of instances for the grammatical and multiple error categories compared to code-switching and spelling. As a result, we observed that the model is comparatively less proficient in sentences where these categories of errors are present. From our extensive evaluation, we observed that DL models outperformed the ML models, underscoring their ability to capture essential linguistic nuances and long-term dependencies within the text, crucial for classification tasks. Transformer models, including BanglaBert and XLM-RoBERTa Base, further exemplify the

Classification Types	Model Name	Performance Metrics		
		Accuracy	Macro Precision	Macro Recall
Binary Classification	TF-IDF + SVM	62.8	62.4	58.3
	TF-IDF + RandomForest	62.7	61.4	59.3
	TF-IDF + XGBoost	63.7	66.8	58.0
	LSTM	64.0	65.0	64.0
	LSTM + Attention	68.0	67.0	65.0
	CNN + LSTM	69.7	68.0	68.0
	XLM-Roberta	74.2	73.6	73.8
	BanglaBERT	79.1	79.7	77.1
Multiclass Classification	TF-IDF + SVM	60.8	59.5	30.1
	TF-IDF + RandomForest	60.3	53.1	32.9
	TF-IDF + XGBoost	61.1	60.5	29.6
	LSTM	62.7	55.2	46.7
	LSTM + Attention	59.4	44.0	39.2
	CNN + LSTM	55.4	41.2	40.0
	XLM-Roberta	69.4	37.6	43.2
	BanglaBERT	74.1	70.7	52.4

Table 5: Performance of different models in Error Classification

power of deep learning in enhancing classification accuracy.

6.3 Corrector Model

The performance of the error corrector model is reported in Table 6 where BanglaT5 and BanglaT5-Small were experimented. Both models perform better in the dataset. BanglaT5 gives 1% improvement rather than BanglaT5 small.

Best Predicted

Comment : কাউকে কষ্ট দিয়ে কেউ কখনো সুখী হতে পাড়ে না

Predicted : কাউকে কষ্ট দিয়ে কেউ কখনো সুখী হতে পারে না

Ground Truth : কাউকে কষ্ট দিয়ে কেউ কখনো সুখী হতে পারে না

Worst Predicted

Comment : জাতীয় পাখি দোয়েল কিন্তু দোয়েল নেই চিরাকানায়

Predicted : জাতীয় পাখি দোয়েল কিন্তু দোয়েল নেই চিরাকানায়

Ground Truth : জাতীয় পাখি দোয়েল কিন্তু দোয়েল নেই চিড়িয়াখানায়

Figure 4: Best predicted and worst predicted input

We obtained better scores in ROUGE-1 and ROUGE-L because the dataset we created consisted of single-word errors mostly. Due to this reason, the best 5 predicted sentences of the dataset have a ROUGE-L score of 1.0 and the worst 5 have ROUGE-L scores between the range of 0.2667 and 0.7500. In Figure 4, we can see how sentences with multiple errors performed poorly. More insights on the ROUGE-L scores can be found in Appendix A.3.

In training the BanglaT5 (Bhattacharjee et al., 2022) model, it took 2.25 minutes per epoch. The average inference time on the test dataset was about 0.2614 seconds. We used another pretrained model BanglaT5 Small for training on the dataset which took almost 0.79 minutes per epoch. The average inference time was about 0.1281 seconds which is almost half of the inference time of BanglaT5 model.

Model	Rouge-1	Rouge-2	Rouge-L
BanglaT5	0.8461	0.4430	0.8459
BanglaT5 Small	0.8343	0.4246	0.8344

Table 6: ROUGE Scores(F1)

When comparing the two models numerically, BanglaT5 consistently outperforms BanglaT5-small in all three Rouge metrics: Rouge1, Rouge2, and RougeL. However, the differences between the

models are relatively small, with BanglaT5 having only a slight edge in terms of these specific evaluation scores.

7 Conclusion

In this study, we embarked on a comprehensive journey to address the critical challenge of Bangla text correction leveraging both traditional machine learning and deep learning techniques along with Transformer models. A pivotal milestone was the creation of a novel dataset from Youtube comments that was meticulously curated and annotated. The dataset serves as the cornerstone for our investigation.

We conducted a rigorous evaluation of machine learning models and deep learning models including transformer models for binary and multiclass error-category classification. The standout performance of BanglaBert showcased its ability to navigate complex linguistic semantics. Additionally, the experimental results underscore the potential of BanglaT5 for improving the accuracy and robustness of correction systems in Bangla user-generated text. BanglaBERT achieves accuracy of 79.1% and 74.1% for binary and multiclass error classification while the BanglaBERT is fine-tuned and tested with our proposed dataset. Moreover, BanglaT5 achieves the best Rouge-L score (0.8459) while BanglaT5 is fine-tuned and tested with our corrected ground truths. Our findings underscored the transformative potential of deep learning models and emphasized the importance of dataset curation. The proposed dataset stands as a unique resource set apart from its predecessors, offering a representation of language use in online settings that are more aligned with the language patterns of Bangla speakers in digital communication.

Limitations

The primary constraint of this study lies in the size of the dataset. While being valuable for Bangla textual error detection and correction tasks, it remains insufficient for broader applications such as classification, complex NLP tasks, and large-scale error correction. Additionally, it would have been advantageous to have more incorrect samples compared to correct ones for enhanced model training. We have excluded comments with an excessive number of emojis, potentially leading to the loss of crucial context in informal communication. we will consider incorporating emojis and special symbols in our future data collection endeavor. Moreover, The dataset's focus remains rooted in the specific linguistic context of Bangladesh. It may not comprehensively represent the linguistic patterns and variations found in other regions where Bangla is spoken.

Future Plan

We look ahead to exploring advanced NLP techniques with an expanded dataset containing more errorful samples to enhance correction systems in Bangla user-generated text. It may have the potential to address a previously underrepresented aspect of Bangla language correction, filling a gap in traditional language model training, especially for generative tasks. Our future plans also involve broadening the scope to accommodate variations in Bangla language as spoken in different regions. We also would like to incorporate the Elo rating system in our experiments.

Ethical Considerations

BaTEClaCor dataset is licensed under CC -BY-NC 4.0 (Creative Commons Attribution). It is important to note that the comments are solely collected for research purposes, in compliance with YouTube's Terms of Service. The anonymity of the commenters was rigorously maintained, with no personal information related to the commenters being captured or stored.

Acknowledgements

This project has been jointly sponsored by Independent University, Bangladesh and the ICT Division of the Bangladesh Government.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Seyyed Mohammad Hossein Dadgar, Mohammad Shirzad Araghi, and Morteza Mastery Farahani. 2016. A novel text mining approach based on tf-idf and support vector machine for news classification. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 112–116. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Tanni Mittra, Sadia Nowrin, Linta Islam, and Deepak Chandra Roy. 2019. A bangla spell checking technique to facilitate error correction in text entry environment. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–6. IEEE.

MF Mridha, Md Abdul Hamid, Md Mashod Rana, Md Eyaseen Arafat Khan, Md Masud Ahmed, and Mohammad Tipu Sultan. 2019. Semantic error detection and correction in bangla sentence. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 184–189. IEEE.

Zhang Qi. 2020. The text classification of theft crime based on tf-idf and xgboost model. In *2020 IEEE International conference on artificial intelligence and computer applications (ICAICA)*, pages 1241–1246. IEEE.

Chowdhury Rafeed Rahman, MD Rahman, Samiha Zakir, Mohammad Rafsan, and Mohammed Eunus Ali. 2022. Bspell: A cnn-blended bert based bengali spell checker. *arXiv preprint arXiv:2208.09709*.

HAZ Shahgir and Khondker Salman Sayeed. 2023. Bangla grammatical error detection using t5 transformer model. *arXiv preprint arXiv:2303.10612*.

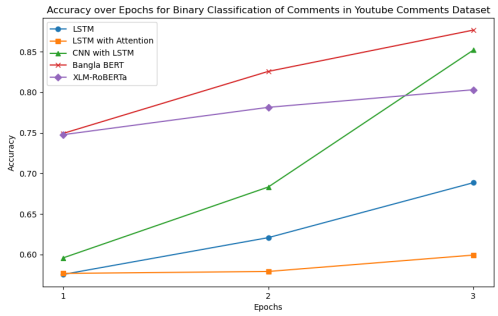
Md Habibur Rahman Sifat, Chowdhury Rafeed Rahman, Mohammad Rafsan, and Hasibur Rahman. 2020. Synthetic error dataset generation mimicking bengali writing pattern. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 1363–1366. IEEE.

Nilam Nur Amir Sjarif, Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat, Haslina Md Sarkan, Yazriwati Yahya, and Suriani Mohd Sam. 2019. Sms spam message detection using term frequency-inverse document frequency and random forest algorithm. *Procedia Computer Science*, 161:509–515.

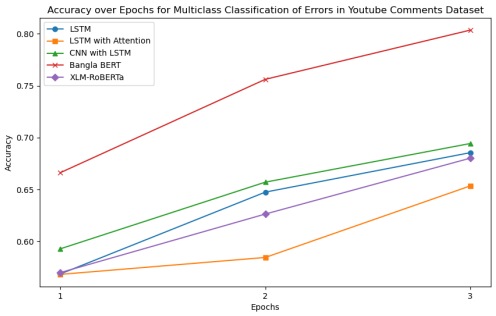
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A Accuracy & Loss Plots

A.1 Accuracy Plots



Binary Classification Accuracy Plot



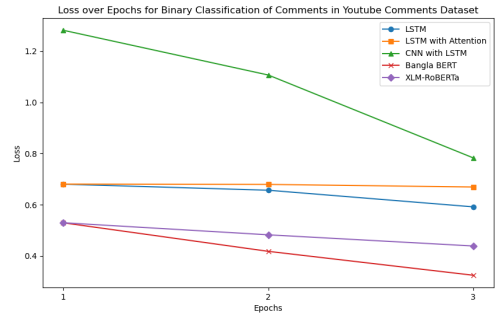
Multiclass Classification Accuracy Plot

Figure 5: Accuracy Plots of Different Models

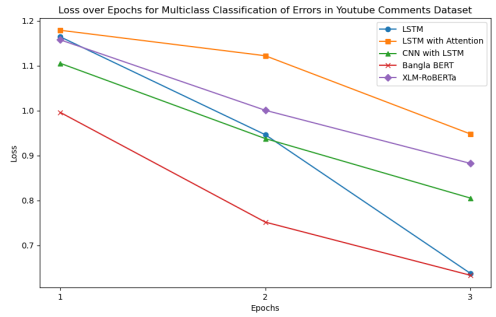
The accuracy plots show, in case of binary classification of comments, we see that BanglaBERT outperformed all the other models with about 88 percent accuracy towards the 3rd epoch. We can also see that,CNN with LSTM had a steady increase of accuracy per epoch i.e from 59.6% in the 1st epoch to 85.2% by the end of 3rd epoch . We see that LSTM with attention had less improvement over the epochs.

In case of multiclass classification of errors, we see that BanglaBERT has better accuracy than other models which is almost 80.35% . CNN with LSTM also gets around 70% accuracy by the end of 3rd epoch. Both LSTM and LSTM with attention's accuracy has minimal improvement over the epochs.

A.2 Loss Plots



Binary Classification Loss Plot



Multiclass Classification Loss Plot

Figure 6: Loss Plots of Different Models

The loss plots show, during the training session of models for binary classification of comments, CNN with LSTM's loss decreasing significantly after each epoch. The change in loss was almost similar for LSTM with Attention and LSTM. A minor reduction in loss was observed for BanglaBERT by the end of 3rd epoch.

Furthermore, during the training session of models for multiclass classification of comments, we see BanglaBERT's loss decreasing significantly in every epoc upto 63 percent after the 3rd epoch. We also see the same for LSTM where there is a significant decrease of loss. The most minimal loss was observed here for LSTM with attention.

A.3 Corrector Model Prediction Analysis

A.3.1 Top 5 Best Predicted Outputs

From top 5 best predicted outputs, we see that common single-word errors were predicated properly which is indicated by the ROUGE-L score of 1.0.

Comment	Predicted	Ground Truth	Rouge-L
ভাই **কোই** পাওয়া যাবে দাম কত টাকা	ভাই **কই** পাওয়া যাবে দাম কত টাকা	ভাই **কই** পাওয়া যাবে দাম কত টাকা	1.0
একদম **আমাকে** সাথে যা যা হয় হুবহু মিলে যায়	একদম **আমার** সাথে যা যা হয় হুবহু মিলে যায়	একদম **আমার** সাথে যা যা হয় হুবহু মিলে যায়	1.0
বাংলাদেশের মানুস জিয়া কে ভালবাসে	বাংলাদেশের মানুষ জিয়া কে ভালবাসে	বাংলাদেশের মানুষ জিয়া কে ভালবাসে	1.0
খুব ভালো লাগলো থ্যাংক ইউ আপনাকে	খুব ভালো লাগলো ধন্যবাদ আপনাকে	খুব ভালো লাগলো ধন্যবাদ আপনাকে	1.0
একটা **মানুস** আরেকটা মানুষকে কেমনে ফেলে চলে যায় এটা তো আমার জানা নাই	একটা **মানুষ** আরেকটা মানুষকে কেমনে ফেলে চলে যায় এটা তো আমার জানা নাই	একটা **মানুষ** আরেকটা মানুষকে কেমনে ফেলে চলে যায় এটা তো আমার জানা নাই	1.0

Figure 7: The Top 5 Best Predicted Outputs

A.3.2 Top 5 Worst Predicted Outputs

Comment	Predicted	Ground Truth	Rouge-L
অসাধারণ একটি ডকুমেন্টারি চ্যানেল	অসাধারণ একটি ডকুমেন্টারি চ্যানেল	অসাধারণ একটি তথ্যচিত্র চ্যানেল	0.7419
ডায়লোগ গুলো বেশি বেশি হইছে	ডায়লোগ গুলো বেশি বেশি হইছে	সংলাপ গুলো বেশি বেশি হইছে	0.7407
ভাইরে ভাই কি এক্সপ্রেসন	ভাইরে ভাই কি এক্সপ্রেসন	ভাইরে ভাই কি অভিব্যাক্তি	0.6667
এমন ভিডিও অরো চাই প্লিজ	এমন ভিডিও অরো চাই প্লিজ	এমন ভিডিও আরো চাই প্লিজ	0.5821
লিজেন্ড কে হারিয়ে ফেরেছি আমরা	লিজেন্ড কে হারিয়ে ফিরেছি আমরা	কিংবদন্তীকে কে হারিয়ে ফেলেছি আমরা	0.5240

Figure 8: The Top 5 Worst Predicted Outputs

There were certain words which were inadequately present in the dataset. Due to which the prediction scores tend to fall for such samples containing those words. We can see that the range of ROUGE-L score lies between 0.5240 and 0.7419 for the worst 5 predicted outputs.

Crosslingual Retrieval Augmented In-context Learning for Bangla

Xiaoqian Li[1] **Ercong Nie**[1,2] **Sheng Liang**[†1,2]
[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2] Munich Center for Machine Learning (MCML), Germany
Xiaoqian.Li@campus.lmu.de
{nie, shengliang}@cis.lmu.de

Abstract

The promise of Large Language Models (LLMs) in Natural Language Processing has often been overshadowed by their limited performance in low-resource languages such as Bangla. To address this, our paper presents a pioneering approach that utilizes cross-lingual retrieval augmented in-context learning. By strategically sourcing semantically similar prompts from high-resource language, we enable multilingual pretrained language models (MPLMs), especially the generative model BLOOMZ, to successfully boost performance on Bangla tasks. Our extensive evaluation highlights that the cross-lingual retrieval augmented prompts bring steady improvements to MPLMs over the zero-shot performance.

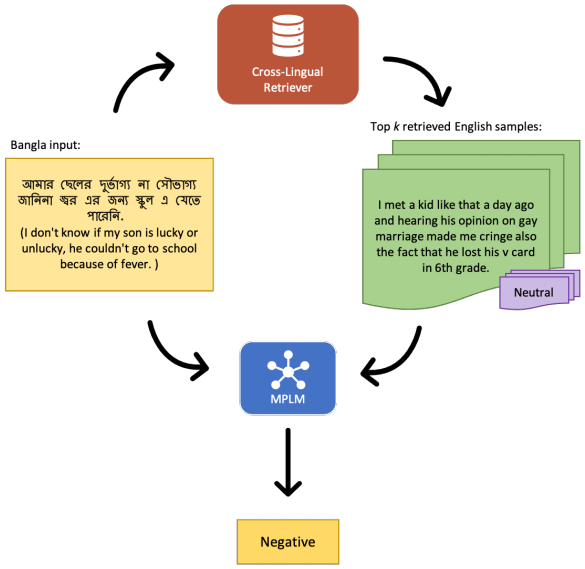


Figure 1: PARC pipeline using decoder-only Multilingual Pretrained Language Models.

1 Introduction

In recent years, the field of Natural Language Processing (NLP) has witnessed transformative advancements, especially with the advent of deep transformer techniques (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2019). The introduction of Large Language Models (LLMs), such as GPT-3 (Brown et al., 2020b) and GPT-4 (OpenAI, 2023), has further revolutionized the landscape. These models showcase unparalleled prowess in tasks like text classification and generation, unified under the umbrella of in-context learning, and cater to a plethora of applications across diverse languages (Conneau et al., 2020; Raffel et al., 2020; Radford et al., 2019). While comprehensive benchmarks like XTREME (Hu et al., 2020) and BUFFET (Asai et al., 2023) underscore their capabilities, languages such as English remain the primary beneficiaries. In stark contrast, several low-resource languages, Bangla being a prime example, grapple with challenges, notably the scarcity of pretraining corpora (Artetxe and Schwenk, 2019; Hangya et al., 2022; Sazzed, 2020).

Despite having a significant number of native speakers, Bangla remains underrepresented in the NLP arena due to linguistic intricacies, limited labeled datasets, and prevalent issues like data duplication (Das and Bandyopadhyay, 2010; Das and Gambäck, 2014). Although there have been commendable strides using conventional machine learning techniques in Bangla NLP tasks, the untapped potential of the latest LLMs is evident (Bhowmick and Jana, 2021; Wahid et al., 2019; Hoq et al., 2021).

In the evolving landscape of in-context learning with LLMs, the concept of retrieval augmentation, which emphasizes sourcing semantically rich prompts, has gained traction (Shi et al., 2023). However, when it comes to multilingual in-context learning, previous works like MEGA (Ahuja et al., 2023) often limit their scope to task instructions and lack deeper semantic insights due to their

† Corresponding author.

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 136–151

approach of random prompt selection. In contrast, strategies like PARC (Nie et al., 2023) pave the way for a more comprehensive methodology, fetching semantically aligned prompts from high-resource languages.

Our work draws inspiration from these methodologies but introduces novel perspectives. While MEGA offers task-level instructions, we infuse semantic understanding into our approach. Similar to PARC, our approach is cross-lingual, ensuring a broader application spectrum. Diverging from PARC's focus on masked language models like mBERT and XLMR, as shown in Figure 1, we venture into uncharted territories by employing larger, decoder-only multilingual pretrained language models (MPLMs) — BLOOM and BLOOMZ — to tackle Bangla NLP tasks in a generative style (Muennighoff et al., 2023; Scao et al., 2022).

In this paper, we explore the application of cross-lingual retrieval augmented in-context learning to Bangla text classification and summarization tasks. Our main contributions encompass:

- An extensive evaluation of cross-language retrieval augmented in-context learning methods in Bangla, achieving steady improvements over the zero-shot performance of MPLMs.
- A pioneering exploration to extend PARC to the generative models, BLOOM and BLOOMZ, providing insights for a unified pipeline of cross-lingual retrieval augmented in-context learning.

2 Related Work

Bangla Natural Language Processing Bangla is a morphologically rich language with various dialects that belongs to the Indo-Aryan branch of the Indo-European language family. With roughly 270 million speakers concentrating in Bangladesh and some regions of India, Bangla is ranked as the 7th most widely spoken language in the world[1]. However, Bangla is still considered as a low-resource language in the NLP research due to the scarcity of digital text resources and annotated corpora.

Research on Bangla NLP has covered a variety of common NLP subfields since 1990s, such as POS tagging (Dandapat et al., 2004; Ekbal and Bandyopadhyay, 2008b), stemming and lemmatization (Islam et al., 2007; Paik and Parui, 2008), named entity recognition (Ekbal and Bandyopadhyay, 2007, 2008a), sentiment analysis (Das and Bandyopadhyay, 2010; Wahid et al., 2019), news categorization (Mansur, 2006; Mandal and Sen, 2014), etc. However, the research in different areas of Bangla NLP still remains sparse. In the era of deep learning, further progress has been made in Bangla NLP, particularly in terms of the development datasets (Rahman and Kumar Dey, 2018; Islam et al., 2021, 2023) and models (Tripto and Ali, 2018; Ashik et al., 2019; Karim et al., 2020). Pretrained language models have achieved decent performance in a large variety of NLP downstream tasks through the fine-tuning. Under this background, Bhattacharjee et al. (2022) pretrained the BanglaBERT model, a BERT-based language understanding model pretrained on Bangla language corpora. With the advent of the large language models (LLMs), zero- and few-shot prompting methods have gradually gained prominence. Hasan et al. (2023) compared the zero- and few-shot prompting performance of LLMs with the finetuned models for the Bangla sentiment analysis task. Our work explores the application of the retrieval-augmented prompting method in Bangla violence detection and sentiment analysis tasks.

Multilingual In-context Learning Brown et al. (2020a) demonstrated that LLMs like GPT-3 can acquire task-solving abilities by incorporating input-output pairs as context. The in-context learning approach involves concatenating input with randomly selected examples from the training dataset, which is also called the prompting method. Recent research (Gao et al., 2021; Liu et al., 2022, 2023; Shi et al., 2023) has expanded on this idea by enhancing prompts for pretrained models through the inclusion of semantically similar examples. The effectiveness of prompting methods for English models extends to multilingual models in cross-lingual transfer learning as well. Zhao and Schütze (2021) and Huang et al. (2022) investigated the prompt-based learning with multilingual PLMs. Nie et al. (2023) incorporated augmented the prompt with cross-lingual retrieval samples in the multilingual understanding and proposed the PARC pipeline. Tanwar et al. (2023) augmented the prompt with not only cross-lingual semantic

[1] https://www.ethnologue.com/insights/ethnologue200/

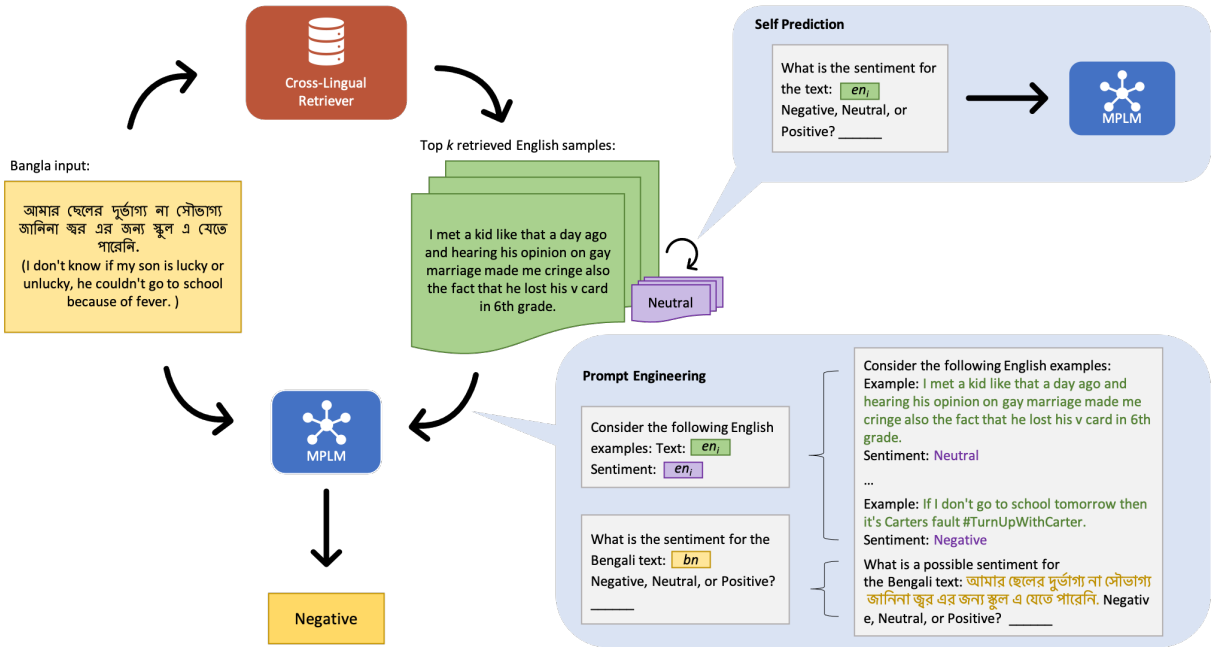


Figure 2: Detailed overview of the PARC pipeline for LRLs using cross-lingual retrieval: (a) An LRL input is used as a query for the cross-lingual retriever, which then retrieves the most semantically similar HRL sample from the HRL corpus. The associated label is either taken directly from the corpus (labeled setting) or determined by self-prediction (unlabeled setting). (b) Next, this HRL sample, its label, and the original input are combined to create a retrieval-enhanced prompt for MPLM prediction.

information but also additional task information. However, previous studies mainly concentrated on the multilingual encoder or encode-decoder models, while our work extend the PARC pipeline to the decoder-only multilingual LLMs.

Multilingual LLMs In the era of LLMs, BLOOMZ and mT0 (Muennighoff et al., 2023) are two representative newly emerging multilingual models. These two multilingual LLMs are finetuned on xP3, a multilingual multitask finetuning dataset, and based on the pretrained models BLOOM (Scao et al., 2022) and mT5 (Xue et al., 2021), respectively. Six different sizes of BLOOMZ models are released from 560M to 176B and 5 different sizes of mT0 models are released from 300M to 13B. These multilingual LLMs open up the possibility for conducting few- and zero-shot cross-lingual in-context learning, as demonstrated by recent benchmarking efforts, for example MEGA (Ahuja et al., 2023) and BUFFET (Asai et al., 2023).

3 Methodology

Our research extends the work of Nie et al. (2023) by focusing on improving multilingual pre-trained language models (MPLMs) for low-resource languages in a zero-shot setting, specifically using retrieved content from high-resource languages such as English.

The backbone of our research approach is a two-stage pipeline consisting of a cross-lingual retriever and a prompt engineering process as shown in Figure 2. This pipeline aims to build on the strengths of MPLMs while mitigating their limitations, especially when dealing with low-resource languages. The first stage of the pipeline uses a cross-lingual retriever that maps the input Bangla text q to a vector q_{embed} in a shared embedding space and uses it as a query. Using semantic similarities with q_{embed}, the retriever returns the most similar k examples from high-resource languages either with or without their labels:

$$R = \arg \max_{i \in \{1,\ldots,|d|\}}^{k} \cos(q_{embed}, d_i)$$

where d_i means each document in the high-resource language corpus and $|d|$ is the number of documents. If there's no label, it suggests a self-prediction step.

The second stage of the pipeline is the prompt engineering. The input Bangla text and the retrieved pattern are subjected to this process. A prefix prompt template P is used to reformulate the

input to facilitate the model's prediction **y**:

$$\mathbf{y} = MPLM(P(q, R))$$

Depending on the architecture of the chosen MPLM, for decoder-only models, the answer is generated by the model directly. For encoder models, the answer is obtained by first mapping each label to its predefined word using the *verbalizer* and then deducing the label word using mask token prediction.

By integrating cross-lingual content retrieval with prompt-guided prediction, we aim to improve the ability of MPLMs to handle low-resource languages. This synergy not only extracts rich linguistic insights from high-resource languages, but also uses them to improve performance on low-resource language tasks.

4 Experiments

In this study, we focused on the tasks of classification and summarization. We refer to our research approach, which uses k retrieved samples for cross-lingual augmented in-context learning methods, as the main method in the following sections.

4.1 Baselines

Zero-shot The template, when populated with the input sample, is fed directly into the MPLM for prediction. This process bypasses the use of cross-lingual context.

Lead64 The first 64 tokens of the input text are taken as a summary of the text (For summarization tasks only).

4.2 Tasks

4.2.1 Classification

Vio-Lens The Vio-Lens dataset (Saha et al., 2023) contains YouTube comments related to violent incidents in the Bengal region, with the goal of highlighting potential threats that could incite further violence. The prompt templates for both main method and zero-shot baseline are defined as follows:

- BLOOMZ-3b and BLOOM-3b:
  `Reflecting on the statement "{text}", which aggressive level does it resonate with: non-aggressive, slightly aggressive, or highly aggressive?`
- mBERT: `The underlying theme in {text} is [MASK].`
  with the verbalizer:
  $v(\textit{Direct Violence}) =$ `assaultive`,
  $v(\textit{Passive Violence}) =$ `indirect`,
  $v(\textit{Non-Violence}) =$ `peaceful`

The English Sentiment Analysis dataset (Rosenthal et al., 2017), which consists of tweets annotated for sentiment on 2-, 3-, and 5-point scales with labels positive, negative, and neutral, serves as the HRL corpora in our study. We use the labeled training set for our experimental sentence pool.

SentNoB Designed to capture the sentiment within text, SentNoB classifies content as positive, negative or neutral (Islam et al., 2021). The prompt templates for both main method and zero-shot baseline are defined as follows:

- BLOOMZ-3b and BLOOM-3b:
  `Text: {text} What is a possible sentiment for the text given the following options?`
- mBERT: `{text} Sentiment: [MASK]`
  with the verbalizer:
  $v(0) =$ `positive`, $v(1) =$ `neural`,
  $v(2) =$ `negative`

We use the ETHOS (onlinE haTe speecH detectiON dataSet) (Mollas et al., 2020) as sentence pool in our experiments. This repository provides a dataset designed to identify hate speech on social media. We use the binary variants of the dataset, which contains 998 comments, each labeled for the presence or absence of hate speech. Since the labels are inconsistent, we use the self-prediction method to predict the labels.

4.2.2 Summarization

XL-Sum is a large and varied dataset consisting of 1.35 million pairs of articles and their corresponding summaries (Hasan et al., 2021). These pairs have been expertly annotated by the BBC and meticulously extracted through a series of carefully designed heuristic methods. The dataset includes 45 languages, from low to high resource, many of which do not currently have publicly available datasets. The prompt template is defined for all models as follows:

- Main method:
  `{text} Generate a concise summary`

```
of the above text using the same
language as the original text
({target_lang}):
```

- Zero-shot baseline:
```
{text} Generate a concise summary
of the given text:
```

4.3 Models

BLOOM is an autoregressive Large Language Model trained on a diverse corpus to generate text based on prompts (Scao et al., 2022). It is capable of generating coherent text in 46 languages.

BLOOMZ takes a novel approach in the MPLM landscape by applying Bloom filters in the context of language models (Muennighoff et al., 2023). This allows the model to use high-resource languages to improve embeddings for low-resource languages, effectively bridging the gap between languages with different levels of available resources.

mBERT is an early MPLM that extends the original BERT model (Devlin et al., 2018). It is pretrained on a corpus of 104 languages, using shared WordPiece vocabularies and a unified architecture for all languages.

mT5 or Multilingual T5 (Xue et al., 2021), is an extension of the T5 (Text-to-Text Transfer Transformer) model (Raffel et al., 2020) designed specifically for multilingual capabilities. Pre-trained on mC4, a large multilingual dataset, mT5 demonstrates multilingual capabilities by transforming input text sequences into output sequences.

Cross-Lingual Retriever We followed Nie et al. (2023) to use the multilingual sentence transformer "*paraphrase-multilingual-mpnet-base-v2*" (Reimers and Gurevych, 2019). This transformer maps sentences and paragraphs into a 768-dimensional dense vector space. Such a high-dimensional embedding facilitates tasks such as clustering and semantic search. In our experiments, the number of retrieval samples k is 1 and 3 for classification task and 1 for summarization task.

5 Results

5.1 Results of classification tasks

Table 1 provides an overview of the results of classification. With the instructions of $k = 3$ retrieval augmented English prompts, we enhance the F1-scores of Bloomz-3b on the two tasks by 5% and 10% respectively. While Bloom-3b, without instruction tuning compared to Bloomz-3b, cannot generate any meaningful result, suggesting that instruction tuning has a strong impact on retrieval augmented in-context learning. The traditional masked MLM, mBERT, also gained improvement by 8% and 7%.

Vio-Lens	zero shot	k=1	k=3
bloomz-3b	0.19	0.2	0.24
bloom-3b	0.00	0.00	0.00
mbert	0.21	0.28	0.29
SentNoB	**zero shot**	**k=1**	**k=3**
bloomz-3b	0.34	0.44	0.44
bloom-3b	0.00	0.00	0.00
mbert	0.30	0.36	0.37

Table 1: F1-scores of the two classification tasks: Bangla zero-shot baseline and with k retrieval augmented prompts.

To facilitate a comprehensive understanding of the performance and discrepancies associated with each task, we present confusion matrices for analysis as follows. Given the confusion matrix in Table 2 , we find that:

1) With a general assessment across micro, macro, and weighted F1 scores, Bloomz-3b and mBERT gained improvement from the retrieval prompts. 2) Compare the two models, Bloomz-3b's zero-shot setting tends to misclassify "non-violence" and "Neutral", and has a reduced macro F1 compared to its weighted F1, while mBERT has a more balanced distribution of confusion between "non-violence" ("Neutral") and the other classes. This may indicate that for classification tasks, the text generation struggles more with minority classes compared to masked prediction.

5.2 Results of summarisation task

The Table 3 compares several models and methods for summarization task.

LEAD-64 As an extractive method, it performs well across all metrics. This indicates that in many cases the first few sentences or tokens of an article or document provide a fairly informative summary. As expected, LEAD-64 outperforms the mt5 base model in the zero-shot setting, but is outperformed by the Bloomz models in the same scenario.

	zero shot			k=1			k=3		
bloomz-3b	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
accuracy			0.33			0.35			**0.36**
macro avg	0.15	0.33	**0.20**	0.18	0.34	**0.20**	0.26	0.26	0.17
weighted avg	0.14	0.33	0.19	0.15	0.35	0.20	0.42	0.36	**0.24**
mbert	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
accuracy			0.22			0.32			**0.33**
macro avg	0.31	0.30	0.18	0.52	0.29	**0.21**	0.18	0.28	**0.21**
weighted avg	0.40	0.22	0.21	0.62	0.32	0.28	0.26	0.33	**0.29**

	zero shot			k=1			k=3		
bloomz-3b	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
accuracy			**0.61**			0.60			**0.61**
macro avg	0.31	0.37	0.34	0.48	0.48	**0.44**	0.47	0.48	**0.44**
weighted avg	0.51	0.61	**0.55**	0.53	0.60	0.54	0.53	0.61	0.54
mbert	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
accuracy			0.35			0.37			**0.39**
macro avg	0.38	0.34	0.30	0.40	0.38	0.36	0.42	0.39	**0.37**
weighted avg	0.43	0.35	0.34	0.47	0.37	0.39	0.48	0.39	**0.41**

Table 2: Confusion matrix of main method in Vio-Lens (top) and SentNoB (bottom) test set of BLOOMZ-3b and mBERT.

	R-1	R-2	R-L	R-LSum
LEAD-64	18.17	5.23	12.73	12.74
zero shot				
mt5-base	5.01	0.84	4.83	4.84
bloomz-1b1	22.08	7.11	18.43	18.44
bloomz-3b	22.36	7.88	18.60	18.58
k=1				
mt5-base	0.97	0.13	0.91	0.92
blommz-1b1	10.84	2.80	9.11	9.12
blommz-3b	6.61	1.52	5.56	5.55

Table 3: Rouge scores of Bangla summarization.

Zero-Shot Models mt5-base produces the lowest scores across all metrics, suggesting that it struggles to produce satisfactory summaries without domain-specific fine-tuning or data augmentation. Both bloomz-1b1 and bloomz-3b show significantly better performance, with bloomz-3b having a slight edge over bloomz-1b1, especially in bigram capture (R-2).

Retrieval augmentation with k=1 Retrieval augmentation seems to drastically affect the performance of mt5-base, reducing its score considerably. This could be due to noise introduced by the retrieved sample or ineffective use of the additional information. For the Bloomz models, bloomz-1b1 still retains decent performance, although there's a drop when compared to its zero-shot performance. Surprisingly, blommz-3b shows a sharper drop, suggesting that the additional retrieval data may be more of a distraction than an advantage for this model configuration in the summarization task.

5.3 Analysis and Discussion

When examining the performance of different models on different tasks, several key observations emerge that are related to linguistic nuances, the underlying language models, and resource allocation.

For classification tasks, it's clear that models with a strong grasp of complex sentence structure and deeper semantics, such as the Bloomz-3b, are more adept at distinguishing nuanced categories like "passive violence" or the more ambiguous "neutral" sentiment. This aptitude likely stems from their ability to understand context better than their simpler counterparts. In parallel, the critical role of zero-shot learning becomes apparent. The ability of a model to generalize a task without specific fine-tuning speaks volumes about its robustness. For example, in our studies, models such as the Bloomz-3b showed commendable performance in a zero-shot setting. Furthermore, as we played around with the variable k (representing the number of samples retrieved), it was instructive to see that a larger value didn't always translate into better performance. This underscores the nuanced ability of a model to sift through information

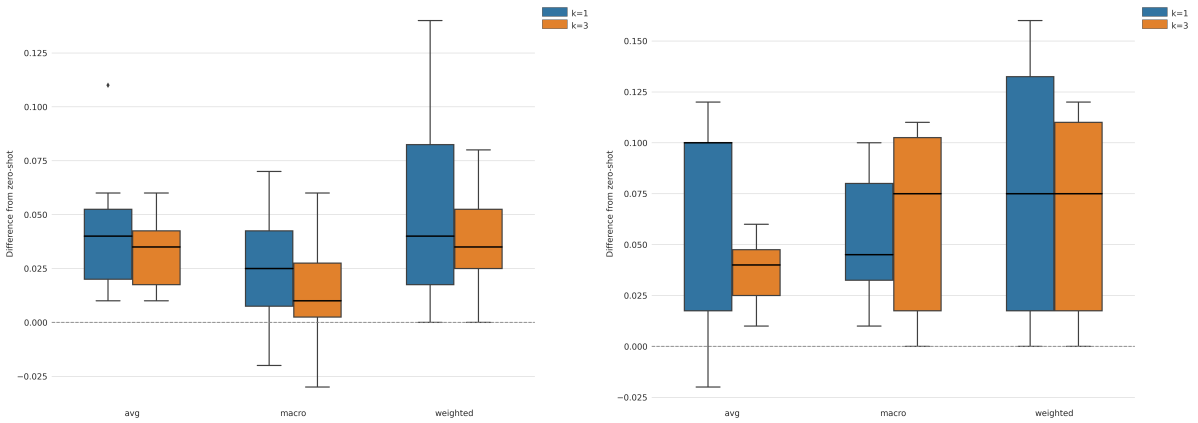


Figure 3: Model performance over differences between zero-shot (represented as '0' on the y-axis) and main method with k=1 and k=3 demonstrations for Vio-Lens test set using bloomz-3b (left) and mbert (right).The y-axis shows the deviations of the main method from the zero-shot values. The statistics are based on 8 and 6 templates, shown in Appendix Table 5 and Table 6, respectively.

and potentially eliminate noise.

Turning to the summarization task, coherence and relevance seem to be the pillars of excellence. Advanced models are more adept at weaving sentences that are not only structurally coherent, but also rich in information. This finesse is evident in the superior Rouge scores of the models. The dichotomy between generative and extractive approaches is also evident. While generative models, including mt5-base and Bloomz-1b1, outperformed the extractive model (LEAD-64) in a zero-shot framework, they seemed a bit sensitive when retrieval augmentation came into play.

Finally, when it comes to resource distribution, there's an undeniable correlation between performance and computational resources. The stellar performance of models like Bloomz-3b likely comes at the cost of intense computational demands. However, one must consider the cost-benefit ratio. In addition, the drop in performance of these models with retrieval augmentation at k=1 suggests a potential sensitivity to the balance or diversity of the dataset.

For the summarization task, an interesting observation is that more extensive models don't always outperform on all metrics, suggesting that we need to be more discriminating in our resource allocation. The significant performance drop with retrieval augmentation further supports this argument.

To conclude this analysis, while modern language models are capable of handling complex tasks, they require careful configuration and thoughtful resource distribution. Unraveling the complexity of these models can pave the way for optimized solutions in both classification and summarization.

6 Ablation Study

6.1 The Stability across Templates

In our experiment for Vio-Lens, we compared the performance of Bloomz-3b and mbert, in terms of their ability to classify text samples into categories. In order to assess the effectiveness of the retrieval augmented prompting method compared to the zero-shot baseline, we conduct a statistic across different templates.

For Bloomz-3b and mBERT, we test different prompt templates , and created a boxplot (Figure 3) to visualize the difference of F1 scores from our main method to the zero-shot baseline across templates. It's shown that with the retrieval augmented English prompts under different templates, both model achieved a stable improvement compared to the Bangla zeroshot baseline. Aso it's clear that mBERT, on average, shows greater improvements in F1 scores when transitioning from the zero-shot baseline to retrieval augmented prompting, compared to Bloomz-3b.

6.2 Impact of Bangla and Hindi Prompt Template

Instead of English, we further explore applying Bangla itself and its linguistically similar high-resource language Hindi as the language of the prompt template , as shown in Table 4.

Main method with English prompt: This configuration yields the highest macro average F1 score

	k=1			k=3		
	precision	recall	f1-score	precision	recall	f1-score
bangla prompt	"পাঠ্য: {text} নিম্নলিখিত বিকল্পগুলি দেওয়া পাঠ্যের জন্য সম্ভাব্য অনুভূতি কী ?"					
accuracy			0.14			0.45
macro avg	0.34	0.09	0.13	0.32	0.28	0.29
weighted avg	0.51	0.14	0.21	0.49	0.45	0.46
hindi prompt	"पाठ: {text} निम्नलिखित विकल्पों को देखते हुए पाठ के लिए संभावित भावना क्या है? "					
accuracy			0.39			0.54
macro avg	0.34	0.28	0.29	0.34	0.34	0.34
weighted avg	0.51	0.39	0.43	0.52	0.54	0.53

Table 4: Results of prompt template in bangla and hindi of main method in SentNoB test of bloomz-3b.

of all three prompt templates.

Hindi Prompt Template: While the Hindi prompt template leads to significant improvements in precision and recall for individual categories such as “Neutral”, the macro average F1 score is still lower than that of the main method with the English prompt.

Bangla prompt template: The Bangla prompt template, while showing some improvements in precision for specific categories such as “positive”, experiences a decrease in recall and overall accuracy. As a result, the macro average F1 score is the lowest of the three templates.

This means that while the Bangla prompt template may improve performance for specific categories, it has an overall negative impact on the model’s ability to generalize across all categories in the SentNoB test. Conversely, the Hindi prompt template’s improvements in precision and recall for individual categories don’t translate into a higher macro average F1 score compared to the main method with the English prompt.

In summary, the macro average F1-score results show that the main method with the English prompt template remains the most effective overall. However, the choice of prompt template can significantly affect performance for specific categories, as demonstrated by the Hindi and Bangla templates. This nuanced understanding underscores the need to balance category-specific and overall performance when selecting prompt templates in cross-lingual retrieval augmentation.

6.3 Impact of Hindi sentence pool

Comparing the results in Table 7 with the previous experiments, we observe that the Hindi retrieval dataset generally improves the model’s ability to retrieve “Neutral” content in the mBERT model. However, the model continues to struggle with the “Neutral” category, with low recall and F1 scores, regardless of the sentence pool used. This suggests that further refinements may be needed to improve retrieval accuracy for neutral sentiment sentences. The studies with Hindi retrieval data show that both bloomz-3b and mbert don’t show any improvements compared to the main method with the English prompt template. This suggests that while using alternative retrieval datasets can improve performance for specific sentiment categories, the choice of retrieval data may need to be carefully considered to maximize overall performance across categories in cross-lingual sentiment analysis tasks.

7 Conclusion

In this paper, we have introduced a novel approach to address the challenges of applying Large Language Models to low-resource languages, with a focus on Bangla. Our methodology employs cross-lingual retrieval-augmented in-context learning, thereby enriching the capabilities of MPLMs, specifically BLOOM and BLOOMZ. We have extensively tested our approach on two classification tasks and one summarization task.

Our experimental results demonstrate the effectiveness of our approach in achieve superior F1 scores for classification tasks.

Upon further analysis, the cross-lingual retrieval mechanism contributes significantly to the model’s performance.

This work lays the foundation for further studies on the application of cross-lingual retrieval and in-context learning methods in low-resource languages. Future work could extend this approach to even more underrepresented languages and potentially adapt it to more complex NLP tasks such as question answering or machine translation.

Limitations

While our study has yielded promising results, it is not without limitations. The effectiveness of retrieval augmentation is also tied to the model architecture, and its impact on different models remains largely unexplored. In addition, the availability of specific language datasets for sentence retrieval and resource constraints remain practical challenges. Further exploration of prompt design and consideration of external factors could improve our methodology. Acknowledging these limitations is essential for a full interpretation of our results and the direction of future research.

Acknowledgements

This work was supported by Leibniz Supercomputing Centre (LRZ), Munich Center for Machine Learning (MCML) and China Scholarship Council (CSC).

References

Kabir Ahuja, Rishav Hada, Millicent Ochieng, Prachi Jain, Harshita Diddee, Samuel Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, et al. 2023. Mega: Multilingual evaluation of generative ai. *arXiv preprint arXiv:2303.12528*.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Akari Asai, Sneha Kudugunta, Xinyan Velocity Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. 2023. Buffet: Benchmarking large language models for few-shot cross-lingual transfer. *arXiv preprint arXiv:2305.14857*.

Md Akhter-Uz-Zaman Ashik, Shahriar Shovon, and Summit Haque. 2019. Data set for sentiment analysis on bengali news comments and its baseline evaluation. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5. IEEE.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Anirban Bhowmick and Abhik Jana. 2021. Sentiment analysis for Bengali using transformer based models. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 481–486, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Sandipan Dandapat, Sudeshna Sarkar, and Anupam Basu. 2004. A hybrid model for part-of-speech tagging and its application to bengali. In *International conference on computational intelligence*, pages 169–172. Citeseer.

Amitava Das and Sivaji Bandyopadhyay. 2010. Phrase-level polarity identification for bangla. *Int. J. Comput. Linguistics Appl.*, 1(1-2):169–182.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387, Goa, India. NLP Association of India.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Asif Ekbal and Sivaji Bandyopadhyay. 2007. A hidden markov model based named entity recognition system: Bengali and hindi as case studies. In *Pattern Recognition and Machine Intelligence: Second International Conference, PReMI 2007, Kolkata, India, December 18-22, 2007. Proceedings 2*, pages 545–552. Springer.

Asif Ekbal and Sivaji Bandyopadhyay. 2008a. Development of bengali named entity tagged corpus and its use in ner systems. In *Proceedings of the 6th Workshop on Asian Language Resources*.

Asif Ekbal and Sivaji Bandyopadhyay. 2008b. A web-based bengali news corpus for named entity recognition. *Language Resources and Evaluation*, 42:173–182.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Viktor Hangya, Hossain Shaikh Saadi, and Alexander Fraser. 2022. Improving low-resource languages in pre-trained multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11993–12006, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Muntasir Hoq, Promila Haque, and Mohammed Nazim Uddin. 2021. Sentiment analysis of bangla language using deep learning approaches. In *COMS2*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *ArXiv*, abs/2003.11080.

Lianzhe Huang, Shuming Ma, Dongdong Zhang, Furu Wei, and Houfeng Wang. 2022. Zero-shot cross-lingual transfer of prompt-based tuning with a unified multilingual prompt. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11488–11497, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Md Ekramul Islam, Labib Chowdhury, Faisal Ahamed Khan, Shazzad Hossain, Md Sourave Hossain, Mohammad Mamun Or Rashid, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023. Sentigold: A large bangla gold standard multi-domain sentiment analysis dataset and its evaluation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4207–4218.

Md Zahurul Islam, Md Nizam Uddin, and Mumit Khan. 2007. A light weight stemmer for bengali and its use in spelling checker.

Md Rezaul Karim, Bharathi Raja Chakravarthi, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 390–399. IEEE.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yanchen Liu, Timo Schick, and Hinrich Schtze. 2023. Semantic-oriented unlabeled priming for large-scale language models. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 32–38, Toronto, Canada (Hybrid). Association for Computational Linguistics.

Ashis Kumar Mandal and Rikta Sen. 2014. Supervised learning methods for bangla web document categorization. *arXiv preprint arXiv:1410.2045*.

Munirul Mansur. 2006. *Analysis of n-gram based text categorization for bangla in a newspaper corpus*. Ph.D. thesis, BRAC University.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. 2023. Cross-lingual retrieval augmented prompt for low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8320–8340, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Jiaul H Paik and Swapan K Parui. 2008. A simple stemmer for inflectional languages. In *Forum for Information Retrieval Evaluation*. Citeseer.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Md Atikur Rahman and Emon Kumar Dey. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2):15.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Salim Sazzed. 2020. Cross-lingual sentiment classification in low-resource Bengali language. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 50–60, Online. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Elizabeth-Jane Pavlick, Suzana Ili'c, Daniel Hesslow, Roman Castagn'e, Alexandra Sasha Luccioni, Franccois Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Rose Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurenccon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa Etxabe, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris C. Emezue, Christopher Klamm, Colin Leong, Daniel Alexander van Strien, David Ifeoluwa Adelani, Dragomir R. Radev, Eduardo Gonz'alez Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady ElSahar, Hamza Benyamina, Hieu Trung Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jorg Frohberg, Josephine L. Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro von Werra, Leon Weber, Long Phan, Loubna Ben Allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, Mar'ia Grandury, Mario vSavsko, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad Ali Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla A. Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto L'opez, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, S. Longpre, Somaieh Nikpoor, S. Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-Shaibani, Matteo Manica, Nihal V. Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Févry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiang Tang, Zheng Xin Yong, Zhiqing Sun, Shaked Brody, Y Uri,

Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre Francois Lavall'ee, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aur'elie N'ev'eol, Charles Lovering, Daniel H Garrette, Deepak R. Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Xiangru Tang, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, S. Osher Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdenvek Kasner, Zdeněk Kasner, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ananda Santa Rosa Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Olusola Ajibade, Bharat Kumar Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David M. Lansky, Davis David, Douwe Kiela, Duong Anh Nguyen, Edward Tan, Emily Baylor, Ezinwanne Ozoani, Fatim Tahirah Mirza, Frankline Ononiwu, Habib Rezanejad, H.A. Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jan Passmore, Joshua Seltzer, Julio Bonis Sanz, Karen Fort, Lívia Macedo Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, M. K. K. Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nourhan Fahmy, Olanrewaju Samuel, Ran An, R. P. Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas L. Wang, Sourav Roy, Sylvain Viguier, Thanh-Cong Le, Tobi Oyebade, Trieu Nguyen Hai Le, Yoyo Yang, Zachary Kyle Nguyen, Abhinav Ramesh Kashyap, A. Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Kumar Singh, Benjamin Beilharz, Bo Wang, Caio Matheus Fonseca de Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel Le'on Perin'an, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Iman I.B. Bello, Isha Dash, Ji Soo Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthi Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, María Andrea Castillo, Marianna Nezhurina, Mario Sanger, Matthias Samwald, Michael Cullan, Michael Weinberg, M Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patricia Haller, R. Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo L. Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Pratap Bharati, T. A. Laud, Th'eo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yashasvi Bajaj, Y. Venkatraman, Yifan Xu, Ying Xu, Yu Xu, Zhee Xao Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2022. Bloom: A 176b-parameter open-access multilingual language model. *ArXiv*, abs/2211.05100.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. Multilingual LLMs are better cross-lingual in-context learners with alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6292–6307, Toronto, Canada. Association for Computational Linguistics.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6. IEEE.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Md Ferdous Wahid, Md Jahid Hasan, and Md Shahin Alom. 2019. Cricket sentiment analysis from bangla text using recurrent neural network with long short term memory model. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4. IEEE.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555,

Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

	zero-shot	k=1	k=3
prompt	{text} Direct Aggression, Indirect Aggression, or No Aggression?		
accuracy	0.53	0.54	0.54
macro avg	0.17	0.18	0.18
weighted avg	0.38	0.38	0.38
prompt	Evaluate the text: '{text}'. Would you categorize it as absence of aggression, mild aggression, or strong aggression?		
accuracy	0.18	0.23	0.23
macro avg	0.17	0.15	0.15
weighted avg	0.13	0.16	0.16
prompt	In the context of '{text}', which category best captures its aggression level: absence of aggression, mild aggression, or strong aggression?		
accuracy	0.12	0.15	0.16
macro avg	0.1	0.15	0.16
weighted avg	0.06	0.11	0.12
prompt	For the text: '{text}', ascertain its aggression scale: absence of aggression, mild aggression, or strong aggression?		
accuracy	0.19	0.21	0.21
macro avg	0.13	0.14	0.14
weighted avg	0.14	0.16	0.15
prompt	From the following choices, which resonates with the theme of '{text}'? Options: No Intensity, Low Intensity, High Intensity		
accuracy	0.13	0.24	0.19
macro avg	0.1	0.17	0.15
weighted avg	0.12	0.26	0.2
prompt	From the following choices, which resonates with the theme of '{text}'? Options: no intensity, low intensity, high intensity		
accuracy	0.23	0.28	0.27
macro avg	0.18	0.22	0.2
weighted avg	0.22	0.31	0.26
prompt	In the context of the text '{text}', which of the following best describes its tone? Options: No Intensity, Low Intensity, High Intensity		
accuracy	0.14	0.2	0.15
macro avg	0.11	0.15	0.12
weighted avg	0.1	0.18	0.13
prompt	Reflecting on the statement '{text}', which aggressive level does it resonate with: non-aggressive, slightly aggressive, or highly aggressive?		
accuracy	0.33	0.35	0.36
macro avg	0.2	0.2	0.17
weighted avg	0.19	0.2	0.24

Table 5: F1-score results with 8 prompt templates of Vio-Lens test using bloomz-3b model

	zero-shot	k=1	k=3
prompt	The text displays [MASK] aggression: {text}		
verbalizer	direct, indirect, none		
accuracy	0.36	0.35	0.36
macro avg	0.22	0.23	0.23
weighted avg	0.31	0.31	0.31
prompt	Considering aggressive tendencies, this is [MASK]: {text}		
verbalizer	overt, covert, absent		
accuracy	0.1	0.2	0.17
macro avg	0.07	0.17	0.14
weighted avg	0.03	0.19	0.15
prompt	From an aggression perspective, the text is [MASK]: {text}		
verbalizer	overt, covert, absent		
accuracy	0.12	0.22	0.2
macro avg	0.09	0.18	0.16
weighted avg	0.06	0.21	0.18
prompt	The described behavior in {text} is [MASK] aggression.		
verbalizer	explicit, implicit, neutral		
accuracy	0.24	0.36	0.35
macro avg	0.19	0.24	0.23
weighted avg	0.23	0.31	0.3
prompt	The underlying theme in {text} is [MASK] aggression.		
verbalizer	assaultive, indirect, peaceful		
accuracy	0.22	0.32	0.33
macro avg	0.18	0.21	0.21
weighted avg	0.21	0.28	0.29
prompt	{text} is interpreted as [MASK] aggression.		
verbalizer	assaultive, indirect, peaceful		
accuracy	0.51	0.49	0.51
macro avg	0.23	0.27	0.25
weighted avg	0.37	0.37	0.37

Table 6: F1-score results with 6 prompt templates of Vio-Lens test using mBert model

	k=1			k=3		
bloomz-3b	precision	recall	f1-score	precision	recall	f1-score
Negative	0.58	0.84	0.69	0.59	0.88	0.70
Neutral	0.09	0.00	0.00	0.08	0.00	0.00
Positive	0.55	0.49	0.52	0.58	0.47	0.52
accuracy			0.57			0.58
macro avg	0.41	0.44	0.40	0.42	0.45	0.41
weighted avg	0.48	0.57	0.51	0.49	0.58	0.51
mbert	precision	recall	f1-score	precision	recall	f1-score
Negative	0.48	0.24	0.32	0.48	0.33	0.39
Neutral	0.21	0.34	0.26	0.21	0.28	0.24
Positive	0.27	0.37	0.31	0.25	0.33	0.28
accuracy			0.30			0.32
macro avg	0.32	0.32	0.30	0.31	0.31	0.31
weighted avg	0.36	0.30	0.30	0.36	0.32	0.33

Table 7: Results in SentNoB test of BLOOMZ-3b and mBERT with hindi retrieval corpus.

Pseudo-Labeling for Domain-Agnostic Bangla Automatic Speech Recognition

Rabindra Nath Nandi[1], **Mehadi Hasan Menon**[1], **Tareq Al Muntasir**[1], **Sagor Sarker**[1], **Quazi Sarwar Muhtaseem**[1], **Md. Tariqul Islam**[1], **Shammur Absar Chowdhury**[2], **Firoj Alam**[2]
[1]Hishab Singapore Pte. Ltd, Singapore
[2]Qatar Computing Research Institute, HBKU, Doha, Qatar
rabindra.nandi@hishab.co, shchowdhury@hbku.edu.qa

Abstract

One of the major challenges for developing automatic speech recognition (ASR) for low-resource languages is the limited access to labeled data with domain-specific variations. In this study, we propose a pseudo-labeling approach to develop a large-scale domain-agnostic ASR dataset. With the proposed methodology, we developed a 20k+ hours labeled Bangla speech dataset covering diverse topics, speaking styles, dialects, noisy environments, and conversational scenarios. We then exploited the developed corpus to design a conformer-based ASR system. We benchmarked the trained ASR with publicly available datasets and compared it with other available models. To investigate the efficacy, we designed and developed a human-annotated domain-agnostic test set composed of news, telephony, and conversational data among others. Our results demonstrate the efficacy of the model trained on psuedo-label data for the designed test-set along with publicly-available Bangla datasets. The experimental resources will be publicly available.[1]

1 Introduction

Modern end-to-end automatic speech recognition (E2E-ASR) systems have made remarkable strides, performing well across various types of data (Li et al., 2020; Gulati et al., 2020; Chowdhury et al., 2021; Prabhavalkar et al., 2023). This success can be attributed to the advancement of deep learning techniques relying on different training strategies, highly dependent on large datasets. However, acquiring and maintaining these high-quality human transcriptions is both expensive and time-consuming, and hence hinders further progress for ASR especially in low-resource languages like Bangla.

To overcome these challenges, two dominant methods, leveraging unlabeled audio, are gaining popularity. These methods include: *(i)* pre-training via Self-supervised learning (SSL) (Baevski et al., 2020, 2022; Chung et al., 2021; Hsu et al., 2021); *(ii)* pseudo-labeling (PL) (Kahn et al., 2020; Xu et al., 2020b; Manohar et al., 2021; Zhu et al., 2023; Xu et al., 2020a; Higuchi et al., 2022). In the pre-training approach, the model is initially trained on raw unlabeled data and then fine-tuned using limited labeled data for some downstream ASR tasks. In pseudo-labeling, a pre-trained model generates labels for unlabeled data, which are then used alongside real labels for supervised ASR training. This paradigm is widely adopted due to its simplicity and effectiveness. Both SSL and PL have been shown to achieve competitive results with minimal labeled data, hence making these paradigms, especially PL, suitable for low-resource languages.

Despite being the 6^{th} most widely spoken language globally, Bangla still falls under low resource language family mainly due to the lack of accessible open datasets. To reduce this gap, we introduce a pseudo-labeling approach to develop an extensive, large-scale, and high-quality speech dataset of $\approx 20,000$ hours for developing domain-agnostic Bangla ASR. First, we curated and cleaned the largest collection of Bangla audio-video data from various Bangla TV channels on YouTube (YT) – varying domains, speaking styles, dialects, and communication channels among others. We then leverage the alignments from two ASR systems, to segment and automatically annotate the audio segments. We enrich the quality of pseudo-labels with our confidence and duration-based filtering method. We utilize the created dataset to design an end-to-end state-of-the-art Bangla ASR. Finally, we benchmark the ASR with widely used, domain-agnostic test sets and compare it with both publicly and commercially available Bangla ASR systems. To test domain-generalization capability, we also

[1]https://github.com/hishab-nlp/Pseudo-Labeling-for-Domain-Agnostic-Bangla-ASR

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 152–162

developed manually annotated test sets that include domain-diverse speech segments.
Our contributions are as follows:

- We develop and release **MegaBNSpeech** – the largest Bangla speech ($\approx 20,000$ hours) training corpus, alongside with its metadata;
- We introduce a robust data collection pipeline that systematically extracted audio segments from listed channels, ensuring wide coverage of speech samples;
- We developed and publicly released a domain-agnostic state-of-the-art Bangla ASR model;
- We developed two test sets comprising (a) diversified domain data from YT; and (b) real-life telephony conversational data, to test model generalizability across domains;
- We benchmark the proposed domain-agnostic Bangla ASR with publicly available test data and ASR models.

The rest of the paper is organized as follows: Section 2 presents previous work, Section 3 describes the dataset, Section 4 formulates our experiments, Section 5 discusses the evaluation results. Finally, Section 6 concludes and points to possible directions for future work.

2 Related Work

2.1 Speech Datasets Development

In the realm of speech corpus development, a variety of methods and techniques have been employed across multiple languages. For example, Wang et al. (2005) focused on Mandarin Chinese, creating a speech corpus from broadcast news and aligning the transcriptions. Similarly, Radeck-Arneth et al. (2015) curated data from diverse sources like audiobooks and web recordings to create a comprehensive speech corpus for German. In terms of automatic speech recognition datasets, Chui and Lai (2008) employed a method that constructs a Mandarin Chinese speech corpus using online videos and automated transcription. In a similar vein, Cho et al. (2021) harnessed web data and automatic alignment techniques to develop a Korean speech corpus geared toward speech recognition research.

Furthermore, current literature has also focused on specialized domains or applications. For instance, in the medical field, Cho et al. (2021) crafted a targeted speech corpus designed for medical dictation tasks, featuring recordings from healthcare professionals. Similarly, in the context of voice assistants, Gale et al. (2019) developed a corpus explicitly aimed at training and evaluating voice-controlled systems.

2.2 Speech datasets for Bangla

There have been several recent works for Bangla Speech Recognition. Sumit et al. (2018) proposed a deep learning based on approach and evaluated model on clean (Alam et al., 2010) and noisy speech datasets (Bills et al., 2016). Ahmed et al. (2020) developed a large annotated speech corpus comprising 960 hours, which are automatically curated from publicly accessible audio and text data. The data annotation primarily relies on publicly available audiobooks and TV news recordings from YouTube. It applies automated techniques such as format conversion, noise reduction, speaker diarization, and automatic gender detection. Transcriptions are generated iteratively using two speech recognition systems, with consensus determining accurate transcriptions. The resulting corpus, referred to as the 'Transcribed corpus', encompasses approximately 510 hours of data.

Similarly, Rakib et al. (2023a) created another extensive dataset with a focus on out-of-domain distribution generalization. The dataset is collected via crowdsourcing campaigns on the duration between Feb 2022 and Nov 2022 on the Mozilla Common Voice (MCV) platform. They followed two collection strategies: (i) scripted and (ii)spontaneous. The dataset contains 11.8k hours of training data curated from 22, 645 native Bangla speakers from South Asia. So far, this is the largest dataset available online for Bangla ASR Recognition. Kibria et al. (2022) also developed a speech corpus that includes 241 hours of both recorded and broadcast speech, featuring contributions from over 60 speakers.

Fleur's datasets are derived from the FLoRes-101 collection[2], which comprises 3,001 Wikipedia sentences. The authors translated development and training sentences from FLoRes-101 into 102 languages and annotated them for ASR applications. We extracted the Bangla test dataset, which includes 920 audio files totaling 3.43 hours. Fleur's dataset consists of 3,010 training, 920 testing, and

[2] https://huggingface.co/datasets/gsarti/flores_101

Datasets	Duration (Hours)	Source	Annotation
Fleurs (Conneau et al., 2023)	15.61	Wikipedia	Human
Common Voice13 (Ardila et al., 2020)	65.71	Open domain	Human
OpenSLR (Kjartansson et al., 2018)	229	Open domain	Human
Bengali Speech Corpus (Ahmed et al., 2020)	960	Youtube	Pseudo
OOD-Speech (Rakib et al., 2023a)	12K	Open domain	Human
MegaBNSpeech (Ours)	**19.8K**	YouTube	Pseudo

Table 1: A comparison of commonly used Bangla ASR datasets

402 validation audio files. We isolated the test files to evaluate them using our chosen models.

Common Voice is a comprehensive, multilingual ASR dataset. As of now, the dataset features 17,689 validated hours across 108 languages, with continual additions of new voices and languages (Ardila et al., 2020). The Common Voice 13 dataset includes 20.7k training, 9.23k testing, and 9.23k validation audio files. We also segregated the test files from this dataset for evaluation with our selected models.

The OpenSLR Bangla dataset, identified as OpenSLR-53, is a substantial ASR corpus sponsored by Google. It consists of a total of 232,537 recordings, amounting to 229 hours of audio data. For our evaluation purposes, we downloaded specific portions of this dataset and randomly selected 10,142 files, amounting to 10 hours of audio data.

Our introduced dataset surpasses all other available Bangla ASR datasets in terms of dataset size and annotation strategy, as outlined in Table 1. Compared to other methodologies, our data annotation pipeline is specialized in several crucial aspects. First, we focus on the manual curation of channels, allowing us to select content from reputable sources, thus enhancing both relevance and diversity. Second, our pipeline leverages both Hybrid ASR and Conformer ASR Models, which are potentially fine-tuned for Bangla, resulting in more accurate transcriptions. Finally, we have implemented a duplicate removal system to remove redundant content. These features make our data annotation process an excellent fit for applications that demand high-quality, domain-specific Bangla language resources.

3 Dataset

3.1 Data Collection

To develop a large-scale dataset focused on diverse domains, we selected YouTube as our data source due to its extensive coverage of Bangla speech. We gathered content from popular news channels such as ATN News, Banglavision News, ZEE 24 Ghanta, News18bangla, Republic Bangla, DD Bangla News, ABP Ananda, NTV News, DBC News, BBC News Bangla, Channel 24, mytvbd news, News24, and Channel I News, among others. Additionally, we included talk shows like RTV Talkshow and ATN Bangla Talk Show. We have also incorporated travel VLOGs into our dataset.

Crawler: To facilitate the collection of data from YouTube, we developed a web crawler that periodically collects videos using youtube-dl.[3] This crawler operates on a list of YouTube channels that we manually pre-select to ensure domain diversity. The crawler then lists all available videos from each channel and proceeds to download them. The download module within the crawler stores the downloaded videos in a Google Cloud Storage (GCS) bucket. The resulting collection consists of $\sim$53K hours with 42K number of videos.

Audio Extraction: We extracted audio from the videos, which were originally in Opus format. To ensure compatibility and standardization, we converted these Opus files to WAV format with a sampling rate of 16 kHz. The conversion process demanded the use of both high CPU and low memory resources. In Figure 1, we provide the data collection pipeline.

3.2 Pseudo Labeling

In Figure 2, we report the architecture of our proposed pseudo labeling approach for the ***MegaBNSpeech*** corpus development. The system takes audio files extracted from videos and passes them into two distinct in-house developed ASR systems:

- **Hybrid ASR (E_1):** Kaldi (Povey et al., 2011) based Factorized Time Delayed Neu-

[3] https://github.com/ytdl-org/youtube-dl

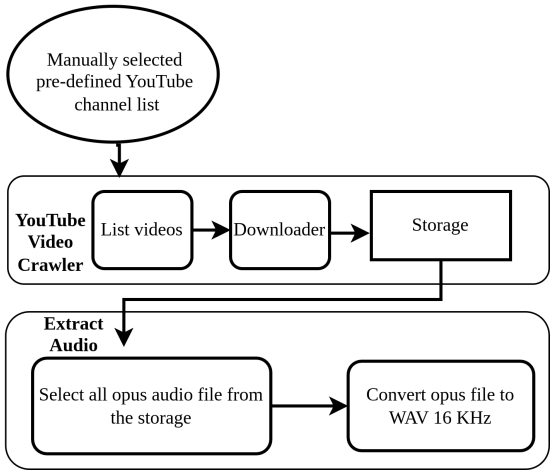


Figure 1: Data collection pipeline.

ral Network (TDNN) (Povey et al., 2018) model is used for training on 1.2K hours transcribed YouTube audio dataset which is manually collected. The model is called hybrid because firstly a Gaussian Mixture Model (GMM) is trained on speech acoustic features for phoneme level alignment and then DNN model is trained on the aligned features. During training, we use 15 factorized TDNN layers in model architecture and 4 epochs. The training recipe is available in the Kaldi Website.[4]

- **End-to-End Conformer ASR (E_2):** Nemo Tooklit based Conformer-CTC model (Gulati et al., 2020) is trained on 4k hours of transcribed YouTube data. A byte-pair encoding (BPE) tokenizer (Wang et al., 2005) is first built using the transcripts of the train set. At training time, pretrained weights of Nemo English ASR [5] are used for initializing weights of the encoder part only. The training parameters are epochs 16, batch size 32, sampling rate 16kHz, use_start_end_token TRUE, pin memory TRUE, number_of_workers 48, trim_silence False, max duration 18.5 and min duration 0.2. The training script is customized from the following the script. [6]

The objective was to leverage the capabilities of these ASR systems to generate transcription based on their decisions. We use the term expert to refer to these systems.

As part of our proposed pseudo-labeling approach, we consider them as expert systems. Based on the transcripts they generate, we take their decisions on segments that match, as depicted in Figure 2. To formally define this, we have two expert systems E_1 and E_2, each of which generates transcripts $T1$ and $T2$, respectively. We use a matching algorithm, Algorithm 1, that employs exact string matching to align the text of segments from the experts E_1 and E_2 ASR systems. The next step involves segmenting the audio based on matching text and removing the segments that do not match. For example, the words highlighted in red in Figure 2 indicate mismatched segments. We therefore remove these segments. The subsequent step is to filter out segments based on predefined criteria. These include: *(i)* confidence score of the ASR systems, *(ii)* minimum and maximum duration of the segments, *(iii)* the ratio of segment duration to the number of words, and *(iv)* the minimum number of words required in a segment. These steps resulted in the final MegaBNSpeech corpus.

Algorithm 1 Transcription matching algorithm.

$T_1 \leftarrow$ Kaldi model (E1)
$T_2 \leftarrow$ Conformer CTC model (E2)
for each (t_1, t_2) in zip (T_1, T_2) **do**
  $\mathcal{M} \leftarrow f(t_1, t_2)$
  for each m in $\mathcal{M}$ **do**
    $r_w \leftarrow$ word rate of m
    $d_a \leftarrow$ segment duration of m
    $c_t \leftarrow$ total characters in m
    $w_t \leftarrow$ total words in m
    if $r_w < r_{w,\text{min}}$ **or** $r_w > r_{w,\text{max}}$ **or** $d_a < d_{a,\text{min}}$ **or** $d_a > d_{a,\text{max}}$ **or** $c_t < c_{t,\text{min}}$ **or** $w_t < w_{t,\text{min}}$ **then**
      continue
    end if
    Write matched transcript and segment
  end for
end for

where $r_{w,\text{min}}$ and $r_{w,\text{max}}$ refers to minimum and maximum word rate; $d_{a,\text{min}}$ and $d_{a,\text{max}}$ refers to minimum and maximum segment duration; $c_{t,\text{min}}$ refers to minimum number of characters, and $w_{t,\text{min}}$ refers to minimum number of total words; $f(t_1, t_2)$ is the longest substring matching function.

[4] https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/local/chain/tuning/run_tdnn_1d.sh

[5] https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_en_conformer_ctc_medium

[6] https://github.com/NVIDIA/NeMo/blob/main/examples/asr/conf/conformer/conformer_ctc_bpe.yaml

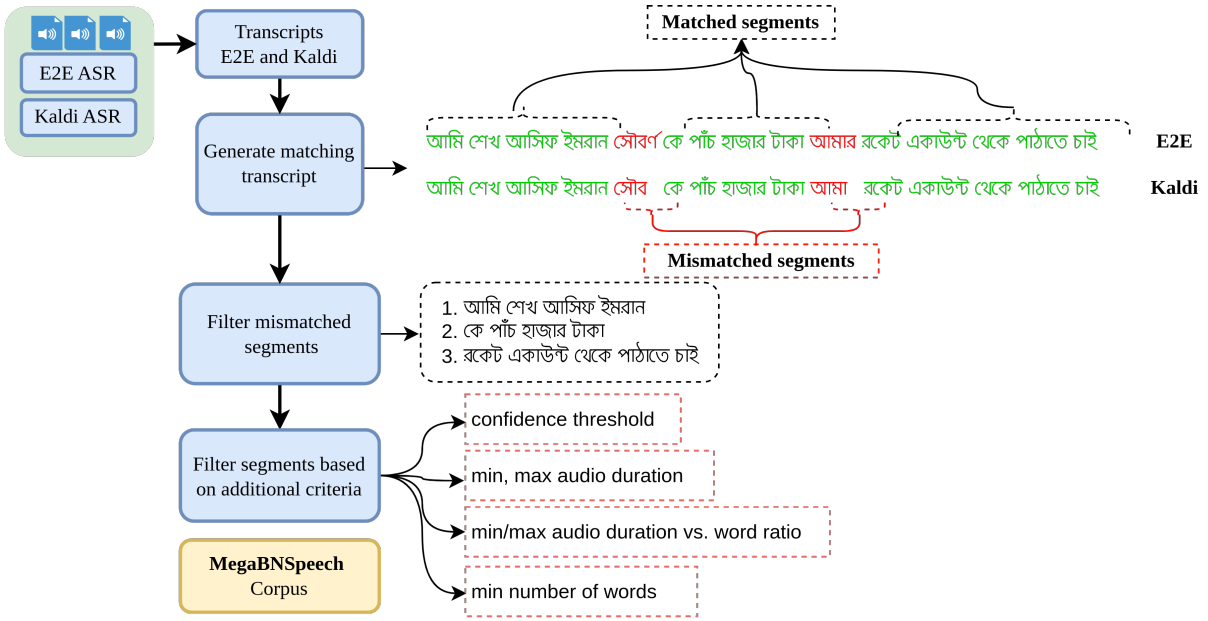


Figure 2: Architecture of the proposed pseudo labeling approach.

3.3 Metadata

To ensure both reproducibility and transferability, we store the metadata in JSON format. This metadata includes the following key elements: *(i)* audio_filepath, *(ii)* text, and *(iii)* duration. The audio_filepath field specifies the path to the audio file, with channel information embedded in the filename. The text field contains the data generated by the pseudo-labeling pipeline, while the duration field indicates the length of the audio in seconds. The audio files have a sampling rate of 16 kHz.

4 Experiments

4.1 Data splits

Training set For training the model, the dataset we selected comprises 17.64k hours of news channel content, 688.82 hours of talk shows, 0.02 hours of vlogs, and 4.08 hours of crime shows. Table 2 provides detailed information about each category and its corresponding duration in hours.

Channels Category	Hours
News	17,640.00
Talkshow	688.82
Vlog	0.02
Crime Show	4.08
Total	**18,332.92**

Table 2: Training data distribution according to channel category and hours

Development set To investigate the robustness of the pseudo-labeling approach, we randomly selected 10 hours of speech to create a development set.

Test set To evaluate the performance of the models, we used four test sets. Two of these were developed as part of the MegaBNSpeech corpus, while the remaining two (Fleurs and Common Voice) are commonly used test sets that are widely recognized by the speech community.

- **MegaBNSpeech-YT Test Set :** The test set has been prepared from a recent collection of YouTube videos, resulting in 8 hours of data. This set is manually transcribed for evaluation purposes. The domains of this set include News, Talkshow, Courses, Drama, Science, Waz (Islamic preaching), etc.
- **MegaBNSpeech-Tele Test Set:** To assess the model's generalization capabilities, we also included 1.9 hours of telephony conversations from our in-house dataset collection, which were subsequently manually transcribed. It involves telephone conversations covering various discussion topics, including online food orders, health services, online ticket bookings, and online banking. The calls were originally recorded using 8kHz sampling rate, which we then upsampled to 16kHz to match the ASR input.[7]

[7]The curated telephony dataset is open-ended conversa-

- **Fleurs:** Fleur's (Conneau et al., 2023) datasets are from FLoRes-101 datasets[8] which contain 3001 Wikipedia sentences. The authors translated dev and train sentences from FLoRes-101 to 102 languages and annotated them to develop ASR. We have separated the Bangla test datasets which contain 920 audio files with 3.43 hours of data. Fleurs contains a total of 3,010 train, 920 test, and 402 validation audio files. We separated the test datasets and evaluated them with our selected models.

- **Common Voice:** Common voice (Ardila et al., 2020) is a massively multilingual ASR dataset. The dataset currently consists of 17,689 validated hours in 108 languages, but more voices and languages are always added. Common Voice 13 contains a total of 20.7k train, 9.23k of test, and 9.23k of validation audio files. We separated the test datasets and evaluated them with our selected models.

4.2 Contemporary ASR Models

Google: Google speech-to-text[9] is a cloud-based ASR service that provides transcription from input Audio for several languages. It provides different domain-specific models for task-specific ASR services. We used the default model and settings and set the language to Bangla.

MMS: Massively Multilingual Speech (MMS (Pratap et al., 2023)) is a fine-tuned model developed by Meta. This model is based on the Wav2Vec2 (Baevski et al., 2020) architecture and makes use of adapter models to transcribe 1000+ languages. The model consists of 1 billion parameters and has been fine-tuned in 1,162 languages. The model checkpoint is published in the HuggingFace model hub.[10]

OOD-speech ASR: OOD-speech ASR is a Conformer-CTC-based model trained on ODD speech datasets (Rakib et al., 2023b). The model consists of 121 million parameters and is trained on 1,100+ hours of audio data which is crowdsourced from native Bangla speakers. The model was trained using NVIDIA NeMo[11] framework and published in Huggingface model hub.[12]

Parameter	Value
epoch	15
global_step	90,911
learning_rate	0.000073287
train_backward_timing	0.1630282
train_loss	11.203718
training_batch_wer	0.149231
val_loss	27.58967
val_wer	0.203385
validation_step_timing	0.089399

Table 3: Details of the hyperparameter settings.

4.3 MegaBNSpeech ASR

We trained the FastConformer model (Rekesh et al., 2023) using the full 18k MegaBNSpeech training sets. During the training phase, we employed a set of predefined parameters: a learning rate of 0.5, a weight decay of 0.001, a batch size of 32, AdamW optimizer, and a maximum audio duration of 15 seconds. We provide details of the hyperparameter settings in Table 3.

To optimize the performance of our model, we conducted experiments with various NVIDIA NeMo architectures and assessed their training accuracy. Specifically, we evaluated the Conformer-CTC, Conformer-Transducer, and Fast-Conformer models. Among these, the Conformer-CTC model exhibited the best performance, achieving a training loss of approximately 11.2%.

To accelerate the training process, we deployed a total of 16 $A100 - 40G$ GPUs to handle the entire dataset. Despite leveraging significant computational resources, the training still took approximately 112 hours to complete.

The model underwent training for 15 epochs, completing approximately 90,911 global steps. The chosen learning rate was relatively low, contributing to stable and incremental updates of the model's parameters. Although the training loss suggests potential for further improvement, it does indicate a narrowing gap between predicted and actual values during the training phase.

As for the WER the value indicates that our

tions with pre-defined topics and includes consent from the interlocutors.

[8] https://huggingface.co/datasets/gsarti/flores_101

[9] https://cloud.google.com/speech-to-text

[10] https://huggingface.co/facebook/mms-1b-all

[11] https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/stable/asr/examples/kinyarwanda_asr.html

[12] https://huggingface.co/bengaliAI/BanglaConformer

model performed with commendable accuracy. However, the validation loss remains somewhat elevated. These metrics offer valuable insights into the model's performance and serve as a road map for future optimization efforts.

4.4 Data Post-processing

During the evaluation of the test sets, we apply a set of post-processing on predicted transcription and human annotation to reduce unexpected symbols, confused words, and misleading alignment. We find that there are some typing issues during manual labeling. To resolve this, a typing error minimization function is applied. In addition, we added two common normalization rules including: (i) number-to-word conversation and (ii) punctuation removal.

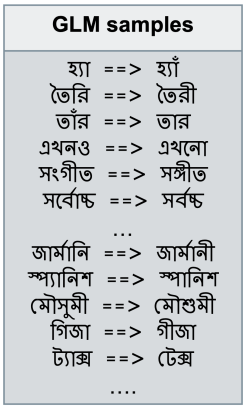


Figure 3: Sample of GLM entries.

Minimizing the confusion due to writing style An extensive analysis of transcriptions indicates many words have different forms of writing (as shown in Figure 3) based on different character combinations. In some cases, both words of confused pairs are acceptable as people annotated in different ways, especially for country names, along with borrowed or code-mixed words.

To minimize these differences, we created a simple Global Mapping File (GLM) that allows different variations of the word to be accepted during evaluation. The GLM file contains entries for different homophones, primarily those with spelling variations. We employed the most frequently occurring confusion patterns for the task, although this approach may not cover all possible variations.

4.5 Evaluation Metrics

To evaluate the performance of the models, we used widely accepted metrics such as Word Error Rate (WER) and Character Error Rate (CER). The reported WER values are presented using the GLM and postprocessing of the hypothesis and references.

5 Results and Discussion

5.1 Robustness of Pseudo-labelling

We first evaluate the robustness of our annotation process for unlabeled audio by utilizing our proposed pseudo-labeling approach. To investigate the quality of these annotations, we used the development set mentioned earlier. This set was subsequently annotated by a human annotator who had no prior knowledge of the ASR-generated pseudo-labels. We then computed the Word Error Rate (WER) and Character Error Rate (CER) between these pseudo-labels (serving as predictions) and the human annotations (acting as ground truth). We observed WER and CER rates of less than 3% (specifically, 2.89% for WER and 2.27% for CER), thereby increasing our confidence in the reliability of the pseudo-labeled datasets.

5.2 Effectiveness of MegaBNSpeech ASR

We initially assess the performance of MegaBNSpeech ASR, which is fully trained on a pseudo-labeled dataset, and compare its ASR performance against other systems such as Google, MMS, and OOD-speech ASRs. Utilizing our in-domain test set (MegaBNSpeech-YT), we noticed a significant performance gap; MegaBNSpeech ASR outperformed the commercial Google ASR, which itself was notably better than the rest (see Table 4).

One plausible explanation for MegaBNSpeech's high performance could be the nature of its training data, which is predominantly sourced from News and Talkshow segments, followed by Science content. These sources typically feature formal speaking styles and limited linguistic diversity, thereby contributing to improved performance. This hypothesis is further supported by the category-level performance data, especially within the 'News' category, as indicated in Table 5.

Across different categories: In Table 5, we report the WER for each category within the MegaBNSpeech-YT test set. From the table, it is evident that all the ASRs (except MMS) perform exceptionally well in the broadcast domain, specifically in News, with MegaBNSpeech achieving nearly 98% accuracy. In the case of talk shows – a

Category	Duration(hr)	MegaBNSpeech	Google	MMS	OOD-speech
MegaBNSpeech-YT	8.1	6.4/3.39	28.3/18.88	51.1/23.49	44.4/33.43
MegaBNSpeech-Tel	1.9	*40.7/24.38	*59/41.26	*76.8/39.36	*69.9/52.93
Fleurs	3.42	*36.1/8.43	24.6/8.54	*39.4/11.58	29.5/13.97
Common Voice	16.5	*42.3/11.44	23.6/ 8.31	*48/14.72	23.6/10.49

Table 4: Reported Word error rate (WER) /character error rate (CER) on four test sets using four ASR systems. * represent the training portion of the corresponding test set **was not** present in the ASR model.

Category	Duration(hr)	MegaBNSpeech ASR	Google ASR	MMS ASR	OOD-speech
News	1.21	2.5/1.21	18.9/10.46	52.2/21.65	32.3/20.71
Talkshow	1.39	6/3.29	28/18.71	48.8/21.5	45.8/34.59
Courses	3.81	6.8/3.79	30.8/21.64	50.2/23.52	46/35.99
Drama	0.03	10.3/7.47	37.3/27.43	64.3/32.74	53.6/45.14
Science	0.26	5/1.92	20.6/11.4	45.3/19.93	33.4/23.11
Vlog	0.18	11.3/6.69	33/22.9	57.9/27.18	49.3/37.22
Recipie	0.58	7.5/3.29	26.4/16.6	53.3/26.89	41.2/29.39
Waz	0.49	9.6/5.45	33.3/23.1	57.3/27.46	59.9/50.38
Movie	0.1	8/4.64	35.2/23.88	64.4/34.96	50.9/42.13

Table 5: Reported Word error rate (WER) /character error rate (CER) on different categories present in MegaBNSpeech - YT test set for four different ASR systems.

synchronized conversational setup – both MegaBNSpeech and Google significantly outperform MMS and OOD-speech. This trend is observed across almost all the categories.

5.3 Generalization Capability to unknown Dataset and Channel

Dataset: To understand how the model performs in unknown domains or datasets, we evaluated the four ASRs using the widely used Fleurs and Common Voice test sets. As seen in Table 4, MegaBNSpeech performs slightly better than MMS ASR on both Fleurs and Common Voice test sets, even though these two datasets are unfamiliar to both MMS and MegaBNSpeech ASR. On the other hand, Google and OOD-speech perform significantly well, with a Word Error Rate (WER) in the range of 23-29%. It should be noted, however, that OOD-speech ASR has been trained on Common Voice data – a crowdsourced dataset where the text prompts are randomly selected from Wikipedia, making it similar to Fleurs. Therefore, the content and style of these datasets are not entirely unknown to these models.

Telephony Channel: To assess how ASR models perform not just in unfamiliar domains but also across different communication channels,[13] we evaluated these four models using telephony conversational data, as shown in MegaBNSpeech-Tel Table 4. Our results indicate that MegaBNSpeech ASR significantly outperforms all other ASRs, with Google coming in second place. This level of performance is consistent with our earlier observations that MegaBNSpeech ASR excels in conversation-style categories like talk shows and vlogs.

5.4 Key Points: Psuedo-labelling based ASR vs Fully-supervised ASR

Traditional ASR training relies heavily on extensive labeled datasets, a requirement that becomes both challenging and expensive to meet for languages, dialects, and domains with limited resources. In contrast, pseudo-labeling not only enriches the training data but also diversifies domain-specific variations, as demonstrated in this study.

From our analysis, we found that MegaBNSpeech performs comparably to supervised out-of-domain (OOD) speech ASR systems, even when exposed to data or domains it has not previously encountered. This shows the efficacy of pseudo-labeling as well as the potential of both the MegaBNSpeech datasets and the model. In this study, we

[13]The collected data was upsampled from an 8K to a 16K sampling rate to match the input sampling rates of the models.

trained MegaBNSpeech exclusively with pseudo-labels to demonstrate the impact of this automated labeling technique. In practical applications, supplementing pseudo-labels with a small amount of manually annotated data can further enhance ASR performance while leveraging the model's strong generalization capabilities.

6 Conclusion and Future Work

This study offers a significant contribution in Bangla speech processing, in addition to the field of ASR particularly for low-resource language. The primary contribution of this paper lies in demonstrating that the model trained with pseudo-labeling only, offers comparable performance with supervised ASR systems. Specifically, the MegaBNSpeech model excels in their ability to generalize across multiple domains and channels as shown in the results.

Additionally, the developed train, development, and two test sets of MegaBNSpeech corpus of $\approx 20,000$ hours of data will serve as a valuable resource for the research community. The MegaBNSpeech corpus, especially the manually annotated YT and telephony test sets, can be used as a benchmark for future studies, enabling other researchers to build upon our work and potentially discover even more effective methods for designing low-resource ASR.

Acknowledgments

We are grateful to HISHAB[14] for providing us with all the necessary working facilities, computational resources, and an appropriate environment throughout our entire work.

7 Limitations

Our data collection originated from YouTube and in-house telephony conversations. Due to restrictions on sharing most of the YouTube content directly, we will instead release links to the YouTube videos along with their transcriptions.

References

Shafayat Ahmed, Nafis Sadeq, Sudipta Saha Shubha, Md Nahidul Islam, Muhammad Abdullah Adnan, and Mohammad Zuberul Islam. 2020. Preparation of bangla speech corpus from publicly available audio & text. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6586–6592.

Firoj Alam, S. M. Murtoza Habib, Dil Afroza Sultana, and Mumit Khan. 2010. Development of annotated bangla speech corpora. In *Spoken Language Technologies for Under-resourced languages (SLTU'10)*, volume 1, pages 35–41, Penang, Malaysia.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Aric Bills, Judith Bishop, Anne David, Eyal Dubinski, Jonathan G. Fiscus, Breanna Gillies, Mary Harper, Amy Jarrett, María Encarnación Pérez Molina, Anton Rytting Jessica Ray, Shelley Paget, Wade Shen, Ronnie Silber, Evelyne Tzoukermann, and Jamie Wong. 2016. IARPA babel bengali language pack iarpa-babel103b-v0.4b ldc2016s08.

Won Ik Cho, Seok Min Kim, Hyunchang Cho, and Nam Soo Kim. 2021. Kosp2e: Korean speech to english translation corpus. *arXiv preprint arXiv:2107.02875*.

S. Chowdhury, A. Hussein, A. Abdelali, and A. Ali. 2021. Towards one model to rule all: Multilingual strategy for dialectal code-switching Arabic Asr. *Interspeech 2021*.

Kawai Chui and Huei-ling Lai. 2008. The nccu corpus of spoken chinese: Mandarin, hakka, and southern min. *Taiwan Journal of Linguistics*, 6(2).

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE.

[14] https://hishab.co/

Robert Gale, Liu Chen, Jill Dolata, Jan Van Santen, and Meysam Asgari. 2019. Improving asr systems for children with autism and language impairment using domain-focused dnn transfer techniques. In *Interspeech*, volume 2019, page 11. NIH Public Access.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Yosuke Higuchi, Niko Moritz, Jonathan Le Roux, and Takaaki Hori. 2022. Momentum pseudo-labeling: Semi-supervised asr with continuously improving pseudo-labels. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1424–1438.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Jacob Kahn, Ann Lee, and Awni Hannun. 2020. Self-training for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7084–7088. IEEE.

Shafkat Kibria, Ahnaf Mozib Samin, M Humayon Kobir, M Shahidur Rahman, M Reza Selim, and M Zafar Iqbal. 2022. Bangladeshi bangla speech corpus for automatic speech recognition research. *Speech Communication*, 136:84–97.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. Crowd-sourced speech corpora for javanese, sundanese, sinhala, nepali, and bangladeshi bengali.

Jinyu Li, Rui Zhao, Zhong Meng, Yanqing Liu, Wenning Wei, Sarangarajan Parthasarathy, Vadim Mazalov, Zhenghao Wang, Lei He, Sheng Zhao, et al. 2020. Developing rnn-t models surpassing high-performance hybrid models with customization capability. *arXiv preprint arXiv:2007.15188*.

Vimal Manohar, Tatiana Likhomanenko, Qiantong Xu, Wei-Ning Hsu, Ronan Collobert, Yatharth Saraf, Geoffrey Zweig, and Abdelrahman Mohamed. 2021. Kaizen: Continuously improving teacher using exponential moving average for semi-supervised speech recognition. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 518–525. IEEE.

Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur. 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech*, pages 3743–3747.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.

Rohit Prabhavalkar, Takaaki Hori, Tara N Sainath, Ralf Schlüter, and Shinji Watanabe. 2023. End-to-end speech recognition: A survey. *arXiv preprint arXiv:2303.03329*.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2023. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

Stephan Radeck-Arneth, Benjamin Milde, Arvid Lange, Evandro Gouvêa, Stefan Radomski, Max Mühlhäuser, and Chris Biemann. 2015. Open source german distant speech recognition: Corpus and acoustic model. In *Text, Speech, and Dialogue: 18th International Conference, TSD 2015, Pilsen, Czech Republic, September 14-17, 2015, Proceedings 18*, pages 480–488. Springer.

Fazle Rabbi Rakib, Souhardya Saha Dip, Samiul Alam, Nazia Tasnim, Md Istiak Hossain Shihab, Md Nazmuddoha Ansary, Syed Mobassir Hossen, Marsia Haque Meghla, Mamunur Mamun, Farig Sadeque, et al. 2023a. Ood-speech: A large bengali speech recognition dataset for out-of-distribution benchmarking. *arXiv preprint arXiv:2305.09688*.

Mohammed Rakib, Md Ismail Hossain, Nabeel Mohammed, and Fuad Rahman. 2023b. Bangla-Wave: Improving bangla automatic speech recognition utilizing n-gram language models. In *Proceedings of the 2023 12th International Conference on Software and Computer Applications*, pages 297–301.

Dima Rekesh, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Juang, Oleksii Hrinchuk, Ankur Kumar, and Boris Ginsburg. 2023. Fast conformer with linearly scalable attention for efficient speech recognition. *arXiv preprint arXiv:2305.05084*.

Sakhawat Hosain Sumit, Tareq Al Muntasir, MM Arefin Zaman, Rabindra Nath Nandi, and Tanvir Sourov. 2018. Noise robust end-to-end speech recognition for bangla language. In *2018 international conference on bangla speech and language processing (ICBSLP)*, pages 1–5. IEEE.

Hsin-Min Wang, Berlin Chen, Jen-Wei Kuo, and Shih-Sian Cheng. 2005. Matbn: A mandarin chinese broadcast news corpus. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 10, Number 2, June 2005: Special Issue on Annotated Speech Corpora*, pages 219–236.

Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert. 2020a. Iterative pseudo-labeling for speech recognition. *arXiv preprint arXiv:2005.09267*.

Tatiana Likhomanenko Qiantong Xu, Jacob Kahn, and Gabriel Synnaeve Ronan Collobert. 2020b. Slimipl: Language-model-free iterative pseudo-labeling. *arXiv preprint arXiv:2010.11524*.

Han Zhu, Dongji Gao, Gaofeng Cheng, Daniel Povey, Pengyuan Zhang, and Yonghong Yan. 2023. Alternative pseudo-labeling for semi-supervised automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

BanglaNLP at BLP-2023 Task 1: Benchmarking different Transformer Models for Violence Inciting Text Detection in Bangla

Saumajit Saha
saha.saumajit@gmail.com

Albert Nanda
albert.nanda@gmail.com

Abstract

This paper presents the system that we have developed while solving this shared task on violence inciting text detection in Bangla. We explain both the traditional and the recent approaches that we have used to make our models learn. Our proposed system helps to classify if the given text contains any threat. We studied the impact of data augmentation when there is a limited dataset available. Our quantitative results show that finetuning a multilingual-e5-base model performed the best in our task compared to other transformer-based architectures. We obtained a macro F1 of 68.11% in the test set and our performance in this shared task is ranked at 23 in the leaderboard.

1 Introduction

Violence inciting text detection (VITD) is the task of identifying text that incites violence in the Bangla language. This is a challenging task due to the complexity of the Bangla language and the variety of ways in which violence can be incited. The VITD task is important for several reasons. First, it can help to prevent violence by identifying and removing inciting text before it can cause harm. Second, it can help to protect people from being targeted by violence. Third, it can help to build a more peaceful and tolerant society.

There are several challenges to VITD in Bangla, and one of them is the scarcity of annotated data, which is due to the limited number of datasets of Bangla text that have been annotated for violence. This makes it difficult to train machine learning models that can accurately detect violence inciting text. Another challenge is the complexity of the Bangla language. Bangla is a morphologically rich language, which means that words can have multiple meanings depending on their context. This can make it difficult to identify violence inciting text, as the same words can be used in both violent and non-violent contexts. Despite these challenges, there has been some progress in the development of VITD systems for Bangla. The VITD task is still a research area, and there is still much work to be done.

2 Related Works

Sharif et al. (2022) introduced a multilabel dataset in Bangla to do aggressive text classification with a hierarchical annotation scheme. (Jahan et al., 2022) created a new Bangla Hate dataset and proposed BanglaHateBERT for abusive language detection in Bangla. (Romim et al., 2022) introduced a manually labeled large hate speech dataset in Bangla.

3 System Description

This section describes our system which is developed to classify violence inciting text written in Bangla. This section starts with the shared task description, followed by the description of the dataset released by the shared task organizers, then our proposed architecture which has produced our team's standing on the leaderboard and finally the results achieved and observations made. All the codes and datasets used for performing the experiments are available in https://github.com/Saumajit/BanglaNLP/tree/main/Task_1.

3.1 Shared Task Description

The objective of this shared task[1] (Saha et al., 2023a) is to identify the threats associated with violence in a given text segment. Given a Bangla text segment as input, the output produced by the system should belong to one of the 3 classes - *Non-Violence*, *Passive Violence* and *Direct Violence*.

3.2 Dataset Description

The dataset (Saha et al., 2023b) comprises YouTube comments related to the top 9 violent incidents that have occurred in the Bengal region

[1] https://github.com/blp-workshop/blp_task1

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 163–167

Sentence	Label
একজন বাবা কতোটা অসহায় হলে এই কথা বলতে পারে আল্লাহ তুমি বিচার করো	Non-Violence
অসৎ এর বাচ্চারা তোরা কলেজ বিশ্ববিদ্যালয়ে এগুলো করার জন্যেই যাস, আর তোদের জন্য নীরিহ মানুষরা মারা যায়	Passive Violence
এই শালারে জন সম্মুখে আগুনে পুড়িয়ে মারা হউক, যাতে করে আর কোন অমানুষ এ রকম কাজ করতে সাহস না পায় ।	Direct Violence

Table 1: Sample dataset for each of the categories

Actual Sentence	Augmented Sentence	Label
হিজাবেই নারীর সৌন্দর্য ফুটে ওঠে।	হিজাবে নারীর সৌন্দর্য প্রতিফলিত হয়।	Non-Violence
ভোট চুরি করলে তো খুন হবেই।	যদি আপনি ভোট চুরি করেন, তাহলে আপনাকে হত্যা করা হবে।	Passive Violence
সকল ছাত্রদের এক হয়া উচিত এবং নিউমার্কেট কে বয়কট করা উচিত।	সকল ছাত্র-ছাত্রীকে একত্রিত হতে হবে এবং নিউমার্কেট বর্জন করতে হবে।	Direct Violence

Table 2: Comparison of actual and augmented data across different categories.

(Bangladesh and West Bengal) within the past 10 years. The dataset encompasses content in Bangla, with comment lengths of up to 600 words. *Non-Violence* refers to the category that pertains to non-violent subjects, such as discussions about social rights or general conversational topics that do not involve any form of violence. In *Passive Violence*, instances of violence are represented by the use of derogatory language, abusive remarks, or slang targeting individuals or communities. Additionally, any form of justification for violence is also classified under this category. *Direct Violence* refers to the category which encompasses explicit threats directed towards individuals or communities, including actions such as killing, rape, vandalism, deportation, desocialization (threats urging individuals or communities to abandon their religion, culture, or traditions), and resocialization (threats of forceful conversion). In Table 1, we can see a snippet of how the sentences in the dataset look like for each of the different categories. Table 3 highlights the distribution of different categories across train and development splits of the dataset.

Class Labels	Train	Dev
Non-Violence	1389	717
Passive Violence	922	417
Direct Violence	389	196

Table 3: Dataset distribution across train and development sets.

3.2.1 Data Augmentation

Finetuning deep learning models requires a lot of data for better performance on the desired task. However, we do not often have a large dataset available. We then require to increase the size of our dataset based on the limited dataset available to us. Feng et al. (2021) highlighted the different approaches available for doing data augmentation in NLP. From Table 3, we can understand that the amount of training data for every category is on the lower side. We therefore tried to augment data by using the Paraphrasing technique to generate text that will try to resemble actual data.

We used *bnaug*[2] library for augmenting data. We augmented 500 samples of each of the Non-Violence and Passive Violence categories. We augmented 389 samples of the Direct Violence category. We randomly chose samples from each category in the training set and then augmented one new sample for each original sample belonging to the training set. We had also tried to augment more number of samples for all categories to create a larger dataset but that led to inferior model performance. Table 2 shows a sample of augmented sentences corresponding to actual sentences for each of the categories.

3.3 Our Approaches

We performed several experiments to solve this task. We started with traditional machine learning algorithms like *Logistic Regression*, *Multinomial*

[2]https://github.com/sagorbrur/bnaug

Naive Bayes (Kibriya et al., 2005), *SGD Classifier*, *Majority Voting* (Lam and Suen, 1997) of earlier approaches and *Stacking* with *XGBoost* (Chen and Guestrin, 2016) as the final classifier. We used TF-IDF (Ramos, 2003) vectorization to convert words into vectors before feeding them to the machine learning algorithms. Table 4 highlights their performance on the development set. These experiments were performed on the actual data split provided, without doing any data augmentation.

Algorithms	Macro-F1
Logistic Regression	52.97%
SGD Classifier	44.8%
Multinomial Naive Bayes	52.13%
Majority Voting of above three	51.67%
Stacking	50.99%

Table 4: Performance of Traditional ML algorithms on the development set

Since we are solving a classification task where the contextual meaning of the sentence matters, we also experimented with several transformer (Vaswani et al., 2017) architectures to see how they perform in this task. We studied the impact data augmentation has when data are scarce and the model is unable to generalize well on unseen data. We used the AutoModelForSequenceClassification class from Hugging Face for finetuning all the models we discussed next.

We initially started with *BanglaBERT* (Sarker, 2020) which is nothing but base ELECTRA (Clark et al., 2020) model pre-trained with Replaced Token Detection objective. This model had been pretrained on the huge amount of web-crawled data and post-filtering to include only Bangla data. We finetuned BanglaBERT in this shared task's dataset using a learning rate of $5e-5$, batch size of 32, and number of epochs set to 4.

We then experimented with the multilingual version of Bert (Devlin et al., 2019), that is, *bert-base-multilingual*[3] which was pretrained using 104 languages. We used a learning rate of $5e-6$ and a batch size of 32, and the best model was obtained after finetuning for 3 epochs.

We also studied how the recently released and very popular multilingual models available in Hugging Face, *multilingual-e5-base*[4] and *multilingual-e5-large*[5] (Wang et al., 2022), perform in our task. Both these models were initialized from *xlm-roberta-base*[6] and *xlm-roberta-large*[7] respectively during pretraining. They undergo a two-stage training process - 1. Contrastive pretraining with unlabelled text pairs to gain a solid foundation on general-purpose embeddings, 2. Supervised training with labeled data so that human knowledge can be injected into the model and it is shown to boost performance. During our finetuning on the shared task's dataset, we used a learning rate of $5e-5$, batch size of 32, and number of epochs as 4. We also prepended a prompt (পাঠ্য অংশের অনুভূতি শ্রেণীবদ্ধ করুন:) to the input text during finetuning of both the variants of multilingual-e5.

3.4 Results and Findings

The evaluation metric for this shared task is Macro-F1. Macro-F1 calculates F1 for each label and finds their unweighted mean. This does not take label imbalance into account. Table 5 highlights the results obtained for different finetuned models with and without applying data augmentation during the development phase. We observed that data augmentation positively impacted model performance, providing significant gains in macro-F1 score. We also found that *multilingual-e5-base* with data augmentation performed the best out of all the experiments performed for this task. We thus chose this finetuned model for inference on the test set and obtained a macro F1 of 68.11% in the test set released during the evaluation phase of this task.

3.5 Error Analysis on Test set

This subsection dives deep into the performance of the model. It provides an analysis of the correct and incorrect predictions of the model on the test set during the evaluation phase. Table 6 highlights a few examples across different categories where the model makes incorrect predictions.

We analyzed the sentences that had been misclassified for each category individually. We looked at the *n-grams* present in those sentences and demonstrated a few of the most commonly oc-

[3] https://huggingface.co/bert-base-multilingual-cased

[4] https://huggingface.co/intfloat/multilingual-e5-base

[5] https://huggingface.co/intfloat/multilingual-e5-large

[6] https://huggingface.co/xlm-roberta-base

[7] https://huggingface.co/xlm-roberta-large

Models	Without Augmentation Macro-F1	With Augmentation Macro-F1	Change in Macro-F1
BanglaBERT	64.5%	69.4%	+4.9
Bert-base-multilingual-cased	67.2%	69.3%	+2.1
Multilingual-e5-base	71.57%	**74.6%**	+3.03
Multilingual-e5-large	60.48%	69.36%	+8.88

Table 5: Finetuning-based experimental Results on the development set

Sentence	Ground Truth	Prediction
দুই হাজার আটের পরে বাংলাদেশে কোনো বিচার হয়নি	Passive Violence	Non-Violence
ওরা রাজনিতি কে কাজে লাগায় বিচার হয় দলীয় ভাবে	Passive Violence	Non-Violence
এদের থেকে বাংলাদেশের নারীদের শিক্ষা নেওয়া উচিত	Non-Violence	Direct Violence
হিজাব বন্ধ করে ঠিকই করেছে।	Direct Violence	Non-Violence
হিন্দুদের কে ভাল করে সাইজ করা অচিত	Direct Violence	Non-Violence
ভারতের স্কুল খোলা অথচ বাংলাদেশে স্কুল বন্ধ। ভারতের মহামারী এখন বাংলাদেশে?	Non-Violence	Direct Violence

Table 6: Snippet of incorrect prediction on the test set.

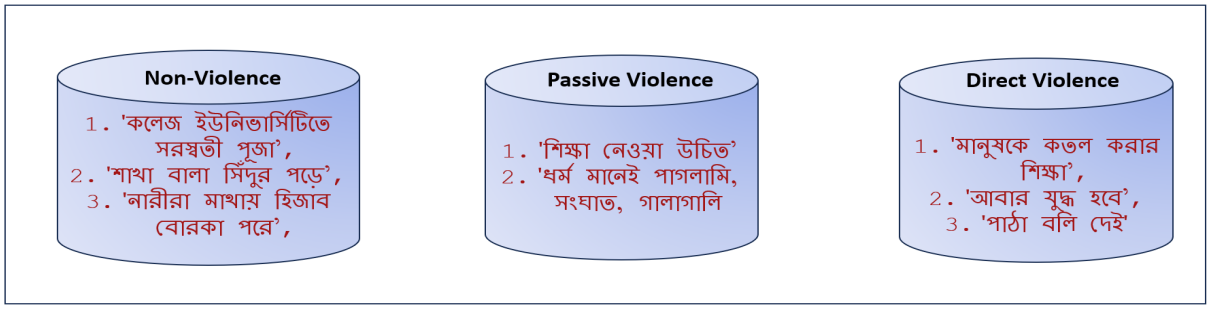


Figure 1: Snippet of phrases that the model has failed to capture correctly.

Figure 2: Confusion matrix on the test set. Non-Vio : Non-Violence, Pass Vio : Passive Violence, Dir Vio : Direct Violence.

curring *n-grams* in Figure 1. For example, when the ground truth is *Non-Violence*, Figure 1 shows that the presence of three phrases has confused the model to make the prediction incorrectly. Similarly, we also found examples of other phrases that may have confused the model for other labels.

Figure 2 highlights the confusion matrix our model's predictions produce on the test set. We observed that out of 201 sentences having *Direct Violence* as ground truth, 147 had been correctly predicted by the model. Similarly, 948 out of 1096 instances had been successfully predicted as *Non-Violence*, and 371 out of 719 instances had been correctly classified as *Passive Violence*. We therefore understand that our model is more accurate in understanding *Non-Violence* and *Direct Violence* categories and it needs to improve for *Passive Violence* category.

4 Conclusion

This paper reports the experiments we performed using the transformer-based models to solve this task. We show the impact that data augmentation has while dealing with smaller datasets. Future research direction can include exploring recently released large language models to solve similar tasks in a low-resource language like Bangla.

5 Limitations

The experiments reported in this paper have produced results in the particular setting of hyperparameters mentioned as well as in the dataset shared by the shared task organizer. We do not do exhaustive hyperparameter optimization for all the experiments reported because of compute constraints. We also do not use ChatGPT anywhere in our experimentation and data augmentation because of pricing constraints. All the experiments are run on Google Colab mostly using V100 and T4 GPUs.

References

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Md Saroar Jahan, Mainul Haque, Nabil Arhab, and Mourad Oussalah. 2022. BanglaHateBERT: BERT for abusive language detection in Bengali. In *Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis*, pages 8–15, Marseille, France. European Language Resources Association.

Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2005. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence*, pages 488–499, Berlin, Heidelberg. Springer Berlin Heidelberg.

L. Lam and S.Y. Suen. 1997. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27(5):553–568.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2022. Bd-shs: A benchmark dataset for learning to detect online bangla hate speech in different social contexts. In *Proceedings of the Language Resources and Evaluation Conference*, pages 5153–5162, Marseille, France. European Language Resources Association.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2022. M-BAD: A multilabel dataset for detecting aggressive texts and their targets. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 75–85, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training.

Team CentreBack at BLP-2023 Task 1: Analyzing performance of different machine-learning based methods for detecting violence-inciting texts in Bangla

Refaat Mohammad Alamgir
Independent Researcher
refaat.alamgir@gmail.com

Amira Haque
Independent Researcher
amirahaque1998@gmail.com

Abstract

Like all other things in the world, rapid growth of social media comes with its own merits and demerits. While it is providing a platform for the world to easily communicate with each other, on the other hand the room it has opened for hate speech has led to a significant impact on the well-being of the users. These types of texts have the potential to result in violence as people with similar sentiments may be inspired to commit violent acts after coming across such comments. Hence, the need for a system to detect and filter such texts is increasing drastically with time. This paper summarizes our experimental results and findings for the shared task on The First Bangla Language Processing Workshop at EMNLP 2023 - Singapore. We participated in the shared task 1 : Violence Inciting Text Detection (VITD). The objective was to build a system that classifies the given comments as either non-violence, passive violence or direct violence. We tried out different techniques, such as fine-tuning language models, few-shot learning with SBERT and a 2 stage training where we performed binary violence/non-violence classification first, then did a fine-grained classification of direct/passive violence. We found that the best macro-F1 score of 69.39 was yielded by fine-tuning the BanglaBERT language model and we attained a position of 21 among 27 teams in the final leaderboard. After the competition ended, we found that with some preprocessing of the dataset, we can get the score up to 71.68.

1 Introduction

With the rise of the Internet, it has become easy to post and comment on multiple social media platforms. Ease of access means that people have the power to influence others to commit violent acts. Early detection and removal of these type of content is necessary to avoid regrettable events such as killing, rape or mass murder. To this end, the Violence Inciting Text Detection (VITD) shared task (Saha et al., 2023a) was introduced at the Bangla Language Processing Workshop at EMNLP 2023. To the best of our knowledge, the shared task and its accompanying dataset (Saha et al., 2023b) are the first of its kind. While previous work explored similar tasks such as hate speech detection in Bangla (Ishmam and Sharmin, 2019), (Romim et al., 2021), this task is the first of its kind to call for systems that can classify a given text as likely to incite violence or not. It has a further fine-grained classification label for violence-inciting texts - namely passive and direct.

In this paper, we discuss our submitted systems to the shared task. We present our analysis related to the dataset and also the models that were submitted.[1]. The paper is organized as follows : first, we analyze the dataset, then we describe the experiments performed both before and after the competition ended. Finally, we analyze the systems submitted and discuss their shortcomings, along with possible directions for future work. The accompanying code for our experiments and analysis is publicly made available.[2]

2 Dataset Overview

The dataset was created compiling YouTube comments in Bangla associated with the top 9 violent incidents that have occurred in the Bengal region (Bangladesh and West Bengal) within the past 10 years.

Comments which stated facts or referred to any kind of social discussion were classified under the Non-Violence category. For comments which contained opinions in a derogatory language and statements which attempted to justify violence were classified under the Passive Violence category. Fi-

[1] https://github.com/blp-workshop/blp_task1

[2] https://github.com/refaat31/team-centreback-blp-task-1

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 168–173

Text	Label	Category
অসংখ্য ধন্যবাদ জানাই ঢাকা কলেজ শিক্ষার্থীদের	0	Non-Violence
ভোট চুরি করলে তো খুন হবেই	1	Passive Violence
তোদের উপর আল্লাহর গজব পড়বে	2	Direct Violence

Table 1: Example comments from each category

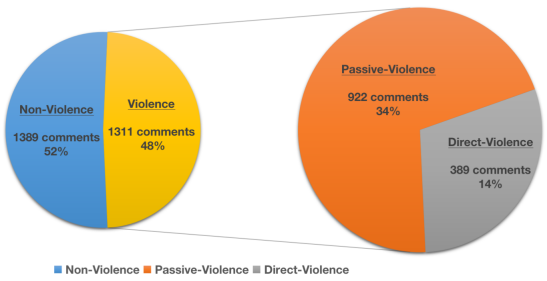


Figure 1: Extensive dataset view showing actual percentages of comments under each category

nally, the comments which presented an unjustified demand, order, actions and any kind of threats were classified under the Direct Violence category. Example comments for each category and an outline of the dataset is given in Table 1.

3 Dataset Analysis

The comments in the dataset have a maximum comment length of up to 600 words. It covers a wide range by considering comments from over the past 10 years.

However, it lacks equality in the amount of data it has for the three different categories. The ratio of Non-Violence to Violence does seem like a roughly equal distribution from a high level. In spite of that, if we dive deep we see that the ratio of Passive Violence to Direct Violence is roughly 2:1. The complete and extensive view of the dataset showing the actual percentages of the comments under each category is given in Figure 1.

Although we do have a total of around 1350 comments for both Non-Violence and Violence categories, if we only consider the Violence categories then we do not have the Direct and Passive Violence categories equally distributed. On one hand for Direct Violence category we have a deficit of around 300 and on the other hand we have a surplus of 300 for the Passive Violence category. Hence, we can conclude that we have a significantly low amount of data for the direct violence category and overall have an imbalance of data in the dataset. Besides, confusion arises

Text	Label	Our Confusion
ওবায়দুল কাদের সরাত হবে একদিন মনে রাখেন ক্ষমতার জন্য আল্লাহ তায়ালার কোরআন এর সাথে বেয়াদবি করছেন এর বিচার খুবই তারাতাড়ি হবে	2	Here we are not seeing any direct threat or action. Instead this is more of an opinion made about an individual so we think this should be label 1 instead of 2.
আওয়ামীলীগের চেতনা	0	It looks like a sarcastic and derogatory remark about a political party so we think this should be category 1.
আল্লাহ কোন দেশে বসবাস করি।আওয়ামীলীগের উপর গজব নাজিল করো।	2	These sentences are similar to one another. Yet both have different labels. We think both of these comments should have label 2 since here a urge towards wishing misfortune for a group of individuals is potrayed.
আল্লাহ তুমি এদের উপর গজব নাজিল করো।	1	
আমরা সনাতনলম্বীরা একটু শান্তি চাই আমাদের সাথে হওয়ার অন্যায়ইয়ের বিচার চাই আর এই নাস্তিকগুলোর ফাঁসি চাই 🙏🙏🙏😡😡😡😡😡😡😡 সব নাস্তিক দের শাস্তি চাই 🙏🙏🙏🙏😡😡😡😡😡	0	This comment is expressing hate and also calling for punishment of a specific community. We think it should have a label of 2 rather than 0.
ওরে বাটপার 😀😀	0	This can have different meanings depending on context.

Table 2: Confusing comment-label combinations

for multiple comments where according to the description it should have been identified as one category but it was denoted as another category. Examples are shown in Table 2.

Furthermore, throughout the dataset instead of portraying religion in a non-discriminative way a biasness was observed towards Islam or Hinduism. Religion was displayed in more of a negative tone which was further strengthened by the lower number of neutral religious comments across the whole dataset. This in turn can lead to a potential misuse and give rise to some ethical concerns. To add to it, the dataset consists of only Bengal regions (Bangladesh and West Bengal). Also, since the only source for the dataset was YouTube comments, a wide range of other diverse origins like educational sites such as Encyclopedias, Blogs, E-books and factual sites like news channel websites were not taken into account. Thus, considering a worldwide perspective a major portion of variety was missing. To summarize, there was an absence of impartiality in terms of religion and regarding the scope a more narrowed down sense was noticed throughout the whole dataset.

Moreover, lots of repetition and redundancy was seen on all parts of the dataset. Not only are there recurrences of comments but also loads of spelling mistakes at frequent intervals. Example comments which show repetition in comments are shown in Table 3. To add to it, throughout the dataset we have seen multiple comments which had missing spacing between independent words. This might have also contributed to the number of misclassifications that we had for the test set. Example of such comments are shown in Table 4.

Also, comments should normally have a variety in the speaking tone. However, no such steps were

Text	Label
ইসলামের বিরুদ্ধে লিখলে ই বিজ্ঞান মনস্ক হয়ে যাবে নাকি?কেন অভিজিৎ কে মারা হলো তা একবারও বলল না।	1
অভিজিৎকে কিসের জন্য মারা হইছে এটা বলা হইলো না হায়রে সাংবাদিক তোরা ও ইসলামের বিরুদ্ধে কথা বলোছ।	1

Table 3: Examples of repetitive comments

Text	Label
ধর্ম যার যারউৎসব সবার,,,,তাহলে , বিয়ে করবে করেছে আসাদুজ্জামান খান কামাল বউ সবার , পুকুর কামালের ,মাছ সবার , কামালের টাক মাথা ঢোল বাজাবে সবাই, গাড়ি কামালের চড়বে সবাই , মেয়ে কামালের সেই মেয়ের সাথে সেক্স করার অধিকার সবার , ইত্যাদি ইত্যাদি ইত্যাদি ইত্যাদি	1
আজ যদি শেখ মুজিবুর রহমান বেঁচে থাকতেনতাহলে হয়তো উনি বলতেন72 এর যুদ্ধ করেছিলাম বাংলার জন্য...এ আমার সোনার বাংলার মানুষ জন্য	0

Table 4: Example comments showing no spacing between independent words

taken to consider the different tones of speaking to further normalize the dataset as a whole. Examples for these kind of words in comments found in the dataset are shown in Table 5.

Additionally, similar token length values were used for all the categories. However, the value taken was comparatively a very small value. The mean value was around 19.6 and the standard deviation value was around 16.6. Hence, the deviation from the mean was very high. In ideal scenarios we would expect the standard deviation to be as low as possible. Standard deviation and mean token length values are shown in Table 6.

Finally, it is worth noting that the emojis have a significance while denoting the category labels. For example, for the comment shown in Table 7, if we consider that emoji has a significance, then the category (1) which it has been given seems justified. However, leaving aside the emoji, it seems like this is simply a normal statement and does not imply a violent tone. Hence, across the whole dataset the emojis played a vital role while classifying the

Text	Word
এই বার ইন্ডিয়ার বিরুদ্দে কথা বলা দরকার কেন জানেন তারা কি জন্য সেনা পাটাবে দেশে সেটা ক কিচু বলচেন না আপনারা কি জন্য বয় করেন আমরা ত দেকতেচি শুনতেচি তারা কি বাজে মনত্তব করতেচে	বলচেন
জ্বি জাই। মনের কথা বলসেন। এত খারাপ পরিস্থিতিতেও মানুষ শপিং এ যাই? চিন্তা করেন	বলসেন

Table 5: Example words in comments showing different speaking tones

	Non-Violence	Passive Violence	Direct Violence
Mean	18.6	21.2	19.2
Standard Deviation	15.9	17.9	16.0

Table 6: Category wise mean and standard deviation token length values

Text	Category
এটা গুজব নয় এইটা সত্যি সত্যিই ঘটেছে 😡😡😡😡	1

Table 7: Example comment showing significance of emoji

comments.

4 System Overview

For all the experiments, we have used either Nvidia Tesla V100 GPU or T4 GPU provided by Google Colaboratory, depending on the availability. During our submission for the competition we did not consider any preprocessing for the dataset and focused fully on the methodology of the model. After the competition ended, we performed some preprocessing to remove punctuations completely from the dataset.

It is important to note that, in the competition, there were two phases - in the first round, we were provided a test set with the ground truth labels, while in the second phase, we had a hidden test set, whose labels were provided after the competition ended. Thus, we have reported our results (in Table 8) on both the first and second round of the competition, as well as the result obtained from experiments performed after the competition ended.

In this competition, the evaluation metric was the macro-F1, which takes the arithmetic mean of the per-class F1 scores. The F1-score is calculated for each class in the following way [3] - $2 * \frac{precision * recall}{precision + recall}$, where precision tells us what fraction of the positive predictions are correct, and recall tells us what fraction of the positive labels have been correctly identified. Here, positive label means that the comment belongs to the class, for which F1-score is being calculated.

4.1 Fine-tuning Language Models

This task can be thought of as a sequence classification task, since we are assigning each comment a category : non-violence, direct violence or passive

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

Model name	macro-F1 (first round)	macro-F1 (second round)
XLM-ROBERTa (base)	68.97	65.43
DistilBERT (base-multilingual)	64.26	63.50
Few-shot learning with SBERT	38.22	33.87
Two-stage (BanglaBERT + catboost with SBERT embeddings)	71.30	69.20
BanglaBERT (50 epochs)	74.81	68.92
BanglaBERT (20 epochs)	76.50	69.39
BanglaBERT with preprocessing done after the competition ended (10 epochs)	77.38	71.68

Table 8: Macro-F1 for different models on the first and second test set provided by organizers

violence. We have fine-tuned a few language models for this purpose by adding a classification head at the end. The training dataset is composed of 2700 Bangla comments and we have used an 80:20 train-test split for our training.

First, we used BanglaBERT (Bhattacharjee et al., 2022) as it was pre-trained on Bangla text and has been shown to have good scores for sentiment analysis task, which is a form of sequence classification. We also used XLM-ROBERTa (Conneau et al., 2019) which is an advanced version of BERT (Devlin et al., 2019), trained on multilingual data. Finally we used the multilingual version of DistilBERT (Sanh et al., 2019). We have trained each of these models for 50 epochs and have noticed that all of the models overfit quite quickly after a certain number of epochs. This can be attributed to the relatively small amount of training data. Finally, BanglaBERT trained for 20 epochs gave the best macro-F1 score of 76.50 for the first test phase set.

4.2 Two stage approach

This was our second best model. The reason for using different models in the two stages is that the models learn unique things in different ways when put under contrasting scenarios. This would result in the model being more versatile and adaptable to any circumstances.

For this method in stage 1, we have done a 80:20 train-test split on the provided original dataset of 2700 comments. Conversely, for stage 2 we have only considered the violence section (both passive and direct violence classes) of the dataset and hence performed a 80:20 train-test split on only the 1311 violence comments.

The process we followed for both the stages are as follows - first we tried to perform one kind of binary classification to categorize comments as either violence or non-violence. Then we did a fine-grained classification of the comments labeled as violent by further classifying them as either direct or passive violence. For the two stages, we used BanglaBERT for the first stage and catboost (Dorogush et al., 2018) with SBERT (Reimers and Gurevych, 2020) embeddings for the second stage. Initially we tried out BanglaBERT for both the stages. In stage 1 it was used for classifying between violence and non-violence and in stage 2 similarly it was used for classifying between direct violence and passive violence. It was seen that it performed better in stage 1 and hence for our two stage approach it was chosen for the first stage. On the other hand, for the second stage both catboost and BanglaBERT gave similar scores hence we thought of going with something non-identical compared to stage 1 and choose catboost as opposed to BanglaBERT. Finally, we believe that two stage training is a possible future direction.

4.3 Few-shot learning with SBERT

As the number of training examples for the category "direct violence" is significantly less compared to the other two categories, we wanted to see if few-shot learning would yield good results. Furthermore, considering the structure of the comments, where they consist of one or more sentences, we encoded them using SBERT. Since SBERT has been shown to perform excellent results on measuring semantic text similarity (STS), we converted our dataset into a suitable format for fine-tuning an STS model. We randomly sampled 100 examples from each class first. Then, if sentence 1 and sentence 2 are of the same class, we gave the sentence pair a label of 1, else we gave it a label of 0.

Finally, for inference, we computed the semantic textual similarity for each sentence embedding as follows: we computed its cosine similarity with every training example, and took the maximum. The class for which we got the highest score, we assigned that class to the test example. The training was done for 50 epochs, and this yielded a poor result as shown in Table 8.

5 Error Analysis

After the competition ended, we performed further analysis to determine the reason for the comparatively low macro-F1 scores. In the test set, we noticed that among the misclassifications done by our best model, a portion of those comments had a lot of repetitive punctuations as shown in Table 9. Furthermore, comments consist of either single sentences or multiple sentences. In order to ensure the

	Predicted	Ground Truth
কারা যেনো বলছিলো ঢাকা কলেজ এর ছাত্ররা নির্দোষ ,,,,,,,, ভাই আপনাদের মুখটা একটু দেখতে চাই	0	1
এই হলো আওয়ামী সংস্কৃতি!!! মন্ডপে কেন প্রতিমার উপর কেন কোরআন শরিফ রাখা হলো??? এটার বিচার আগে করেন কাউয়া কাদের	0	1

Table 9: Example comments showing repetitive punctuations

model does not treat comments of variable length sentences differently, we determined that the complete removal of punctuations was necessary. We achieved this with the bnlp toolkit (Sarker, 2021), which is an excellent library for preprocessing text in Bangla. After this was done, it improved our score from 76.5 to 77.38 for test set 1 and 69.39 to 71.68 for test set 2 . We also noticed that training for 10 epochs seems to give us the best score for the final test set. This is in line with our previous observation that the model overfits quite quickly due to the relatively small amount of training data. Thus, the best model is actually BanglaBERT trained for 10 epochs which gives a score of 71.68 on comments that have punctuations completely removed.

6 Future Works

Although we tried out different methods but our system did not take into account a number of things. Firstly, the spelling mistakes and missing spaces between two independent words in both training and inference stages. Secondly, the significance of emojis was also not taken into consideration. Furthermore, additional knowledge bases for fine-tuning could have also been used to see if it solves the issue with the limited dataset. Lastly, the repetition of similar comments throughout the whole dataset was also not taken into account.

The points mentioned above can be considered for future work for improving violence inciting text detection. In addition, the performance of large language models can also be investigated in this task, as they have been recently shown to perform well on different NLP tasks. (Liu et al., 2023)

7 Conclusion

In this paper, we have presented our experiments and findings for the BLP Shared Task 1 : Violence Inciting Text Detection. Initially, we provide a detailed analysis of the dataset, showing statistics and discussing problems with the dataset. We have found that BanglaBERT fine-tuned for 20 epochs gives us the best macro-F1 score of 69.39. After the competition ended, we analyzed the possible reasons for misclassifications. To further explore and overcome some of those causes, we conducted different experiments that led to a further improvement, taking the macro-F1 score to 71.68. Finally, we discussed the shortcomings of our system and the various possible directions for future work that can improve the detection of violence-inciting texts.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Alvi Md Ishmam and Sadia Sharmin. 2019. Hateful speech detection in public facebook pages for the bengali language. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pages 555–560. IEEE.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023. Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam. 2021. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*, pages 457–468. Springer.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Sagor Sarker. 2021. Bnlp: Natural language processing toolkit for bengali language. *arXiv preprint arXiv:2102.00405*.

EmptyMind at BLP-2023 Task 1: A Transformer-based Hierarchical-BERT Model for Bangla Violence-Inciting Text Detection

Udoy Das, Karnis Fatema, Md Ayon Mia, Mahshar Yahan, Md Sajidul Mowla, MD Fayez Ullah, Arpita Sarker, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
u1804{109, 052, 128, 007, 100, 094, 099}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

The availability of the internet has made it easier for people to share information via social media. People with ill intent can use this widespread availability of the internet to share violent content easily. A significant portion of social media users prefer using their regional language which makes it quite difficult to detect violence-inciting text. The objective of our research work is to detect Bangla violence-inciting text from social media content. A shared task on Bangla violence-inciting text detection has been organized by the First Bangla Language Processing Workshop (BLP) co-located with EMNLP, where the organizer has provided a dataset named VITD with three categories: nonviolence, passive violence, and direct violence text. To accomplish this task, we have implemented three machine learning models (RF, SVM, XGBoost), two deep learning models (LSTM, BiLSTM), and two transformer-based models (BanglaBERT, Hierarchical-BERT). We have conducted a comparative study among different models by training and evaluating each model on the VITD dataset. We have found that Hierarchical-BERT has provided the best result with an F1 score of 0.73797 on the test set and ranked 9^{th} position among all participants in the shared task 1 of the BLP Workshop co-located with EMNLP 2023.

1 Introduction

The presence of violent language on social media has significantly increased in recent times which may lead to bigger crime in real life. Violent texts often result in cyberbullying in online communication. Government authorities and social media companies are very much concerned about such a critical issue. A significant amount of previous studies have been conducted on hate speech and toxic, and abusive text detection. The majority of related research works have been done in high-resource languages, like English (Lee et al., 2018). However, little has been done for low-resource languages such as Bangla. Since many social media users prefer using their regional languages, such as Bangla, it becomes a greater challenge to identify violent text content. Difficult lexemes and no specific pattern for tokens make Bangla violent word detection so hard.

Rule-based machine learning methods (Jia et al., 2019, Khalafat et al., 2021) for detecting violent text are considered insufficient nowadays. Therefore, applying rule-based lexical analyzers or parsing methods provides poor performance. Deep learning-based (Castorena et al., 2021) and transformer-based (Arellano et al., 2022,Ta et al., 2022) approaches provide better performance compared to traditional rule-based machine learning methods violence-inciting text detection. Transformer-based approaches have not been utilized for violent text detection in the Bangla language.

The primary objective of this paper is to detect violence-inciting text in Bangla on social media using a hierarchical transformer. The First Bangla Language Processing Workshop (BLP), co-located with EMNLP, has arranged a shared task, introducing a novel dataset called VITD, categorized into nonviolence, passive violence, and direct violence text, for the purpose of detecting Bangla violence-inciting text (Saha et al., 2023a,b).

To achieve this objective, we have employed a diverse range of models, including three machine learning models (RF, SVM, XGBoost), two deep learning models (LSTM, BiLSTM), and two transformer-based models (BanglaBERT, Hierarchical-BERT). We have conducted a comparative analysis by training and evaluating each model on the VITD dataset, ultimately determining that the Hierarchical-BERT model has outperformed the others with an impressive F1 score of 0.73797 on the test set. In the hierarchical-based transformer model, the first BERT model is em-

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 174–178

ployed to differentiate between violence and non-violence text while the second BERT is used to classify direct and passive violence text.

The core contributions of our research work are as follows-

- We have developed a hierarchical transformer-based technique for detecting violent text.
- We have conducted a series of experiments on the dataset and provided a comprehensive analysis of their performance outcomes.

The implementation details have been provided in the following GitHub repository - https://github.com/ML-EmptyMind/blp-task1.

2 Related Work

The previous studies on Violence Inciting Text Detection (VITD) can be categorized under machine learning, deep learning, and transformer-based approaches.

Traditional machine learning (ML) techniques have been applied for violence text detection in online social media platforms (Khalafat et al., 2021). Machine learning algorithms such as Support Vector Machine (SVM), Naive Bayes (NB), and K-Nearest Neighbours (KNN) have been utilized where SVM provides the best result for violence-inciting text detection. In another previous work, a lexicon-based method has been used to retrieve violence-related microblogs and then a similarity-based method has been applied to extract sentiment words to detect violent text (Jia et al., 2019) which outperforms the previous SVM methods. However, hierarchically structured categories in text categorization have been used (Krendzelak and Jakab, 2015). They have emphasized the importance of considering the impact of hierarchy on machine learning approaches for improved text classification efficiency.

Compared to the traditional methods used for detecting violence-inciting text (VITD), deep learning (DL) based approaches are less dependent on explicitly defined features. Instead, these models learn patterns and features automatically. A deep learning neural network has been capitalized to detect gender-based violence (GBV) in Mexican tweets (Castorena et al., 2021). They have used techniques like CountVectorizer and a multilayer perceptron to design the model architectures.
A fine-tuned transformer named DistilBETO has been applied to detect aggressive and violent incidents from social media in Spanish (Arellano et al., 2022). Another approach utilizes GAN-BERT to detect violent text in the same dataset (Ta et al., 2022).

3 Dataset

We have utilized the violence detection dataset provided under shared task 1 (VITD) of the BLP Workshop @ EMNLP 2023 (Saha et al., 2023b). This dataset contains three categories Non-Violence, Passive Violence, and Direct Violence. The dataset is divided into three sets train, dev, and test with 2700, 1330, and 2016 samples. Each split contains 16-18 words on average. Each category contains 14-20 words on average. Table 1 shows that the provided dataset is imbalanced. The number of samples under the nonviolence category is considerably high (1389) whereas the number of samples under the direct violence category is significantly low (389).

Split	Nonviolence	Passive Violence	Direct Violence
Train	1389	922	389
Dev	717	417	196
Test	1096	719	201

Table 1: Category-wise distribution in the dataset

As this corpus has been built using YouTube comments, the input text contains several emojis and repeated punctuation. During the training and evaluation phase, several preprocessing steps have been performed on the dataset. We have removed all emojis, punctuation, extra spaces, URLs, ZWNJ, and ZWJ from the input text. However, as the numeric text is vital for semantic analysis, we do not remove the numeric text. At the final step of the preprocessing, we have normalized the text using a popular Bangla text normalizer library (Hasan et al., 2020).

4 Methodology

In this section, we provide an overview of the methods and techniques used on the dataset explained before. Initially, we have extracted features using different extraction techniques and applied various ML and DL algorithms. Moreover, different transformer models have been applied to develop the system shown in Figure 1.

Machine learning based approaches for detecting violence inciting text, we have applied traditional ML-based methods such as Random Forest and Support Vector Machine. We also have used XGBoost as an ensemble classifier to improve the performance. Here, we have used NLTKTokenizer to tokenize the dataset and applied Word2Vec to extract features from the dataset. For SVM, we have chosen the parameter C value of 1 for a soft margin in a hyperplane. For the ensemble method, we have specified the boosting rounds or number of decision trees to n_estimators = 100.

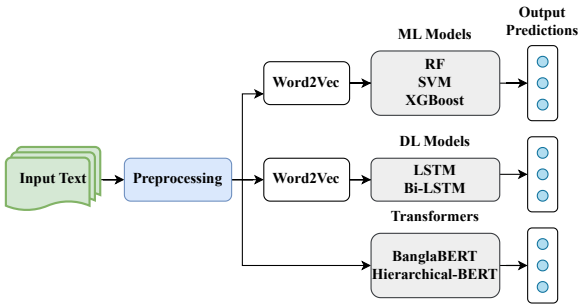


Figure 1: Abstract process of violence text detection

Deep learning-based approaches have been utilized for detecting violence-inciting text. We have implemented two LSTM-based models. In the first model, we have applied three bidirectional LSTM layers with different numbers of LSTM cells. The three directional layers consist of 32, 16, and 8 bidirectional LSTM cells respectively. In the second model, we have used four LSTM layers which consist of 32, 32, 16, and 8 LSTM cells respectively. Both models have been trained up to 10 epochs. We have used categorical cross-entropy as the loss function, and callback method to monitor validation loss during training to select the best model.

Transformer-based approaches are being used very widely in many aspects nowadays. We have employed BanglaBERT(Bhattacharjee et al., 2022) to address this task. As the dataset is imbalanced, we have used the hierarchical approach shown in Figure 2. In the hierarchical approach, we first classify the violence and non-violence text, then further classify the violence text into direct violence and passive violence. For both classification tasks, we have finetuned two BanglaBERT models.

At first, we have divided the dataset into two groups, violence, and nonviolence. Under the violence category, we have assigned the remaining two categories- direct and passive violence. We have finetuned one BanglaBERT model named

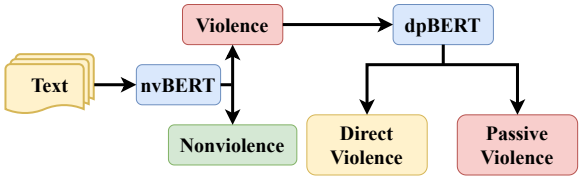


Figure 2: Hierarchical-BERT (HBERT)

nvBERT to distinguish between violence and non-violence texts. Then, we have finetuned another BanglaBERT model named *dpBERT* using violence-categorized text which further classifies the violent text into direct and passive violence. At the time of inference, when nvBERT recognizes a text as violence then we give that text as input to dpBERT to determine whether it is direct or passive violence.

Passive and direct violence texts are significantly different from non-violence texts. In passive or direct violence texts, we find the presence of violent words which is not the case for non-violence texts. That is the reason we have selected the combination where first nvBERT finds out the non-violence text and dpBERT finds out the passive and direct violence text.

5 Results and Analysis

In this section, we present performance comparisons among various machine learning, deep learning, and transformer-based approaches.

5.1 Parameter Setting

Table 2 shows parameter settings for different models.

Model	lr	optim	bs	wd	wr
nvBERT	$2e^{-5}$	adafactor	16	0.01	0.1
dpBERT	$2e^{-5}$	adafactor	16	0.01	0.1
BBERT	$6e^{-5}$	adafactor	16	0.01	-
LSTM	$1e^{-3}$	Adam	32	-	-
BiLSTM	$1e^{-3}$	Adam	32	-	-

Table 2: Parameter settings for different models

In Table 2, *lr*, *optim*, *bs*, *wd*, and *wr* represents *learning_rate*, *optimizer*, *batch_size*, *weight_decay*, and *warmup_ratio* respectively. Also, model name BBERT represents BanglaBERT.

Categories	SVM	RF	XGBoost	LSTM	BiLSTM	BanglaBERT	Hierarchical BERT
Nonviolence	0.72	0.72	0.74	0.72	0.78	0.84	**0.85**
Passive Violence	0.43	0.39	0.45	0.48	0.61	0.70	**0.71**
Direct Violence	0.13	0.11	0.30	0.30	0.52	0.65	**0.65**

Table 3: Category wise F1-Score based performance of various systems on test set

5.2 Evaluation Metrics

The performance of various models has been evaluated by calculating the precision (P), recall (R), and F1-Score on the test set.

5.3 Comparative Analysis

We have found that among the machine learning models, the XGBoost has achieved the highest F1 score (0.5). We have trained different deep learning-based models, where the stacked BiLSTM model has provided the best F1-score of 0.633. BanglaBERT has achieved the highest precision of 0.73 whereas Hierarchical-BERT has provided the highest F1-score of 0.73797 respectively. Table 3 highlights the classwise F1-score. Table 4 shows the performance of nvBERT and dpBERT on the test set. The nvBERT model performs slightly better than the dpBERT model in terms of classification. With a high margin compared to ML and DL-based approaches, Hierarchical-BERT performed better and slightly better than BanglaBERT securing 9^{th} rank in the leaderboard.

Classifier	Macro Average		
	P	**R**	**F1**
nvBERT	0.83	0.832	0.8310
dpBERT	0.81	0.893	0.8306

Table 4: Performance matrix for both BERTs of Hierarchical BERT on the test set. Here P, R, F1, TF denotes to Precision, Recall, F1-Score, Transfomrer.

	Classifier	Macro Average		
		P	**R**	**F1**
ML	RF	0.63	0.41	0.40
	SVM	0.63	0.63	0.43
	XGBoost	0.63	0.50	0.50
DL	BiLSTM	0.63	0.68	0.633
	LSTM	0.39	0.42	0.40
TF	BBERT	**0.75**	0.71	0.730
	HBERT	0.73	**0.79**	**0.738**

Table 5: Performance of various systems on test set

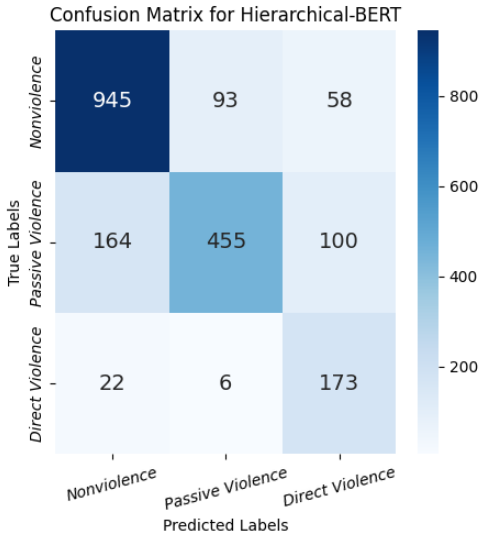


Figure 3: Confusion matrix of Hierarchical-BERT model

5.4 Error Analysis

Table 5 shows that the Hierarchical-BERT model has outperformed all models in terms of classifying violence-inciting text. To get further insights about the system, a confusion matrix (Figure 3) is used. We notice that the model achieves the highest True Positive Rate (TPR) of 86.22% for the nonviolence category and 86.07% for the direct violence category. However, the model provides the lowest TPR of 63.28% for the passive violence category.

Our model has misclassified text of passive violence category as nonviolence or direct violence. Non-violence text does not contain any direct violence words. Passive violence is treated as non-violence words because often passive violence does not contain any violent words. Thus nvBERT treats passive violence as non-violence text due to the lack of direct violent words. Therefore, it leads to misclassification between non-violence and passive. Table 1 indicates imbalances between three classes and therefore leads to misclassification.

6 Conclusion

In this research work, we have conducted a comparative study among different machine learning, deep learning, and transformer-based models for Bangla violence-inciting text detection in social media content. During the training and evaluation of the different models, we utilized the VITD dataset provided in a shared task. We have found that the Hierarchical-BERT model has outperformed all other models with an F1-score score of 0.73797. The error analysis shows that our trained models become biased toward the majority class. In the future, we will address the issue by incorporating different strategies to address the class imbalance in the VITD dataset.

Limitations

Several limitations can be noted in our work. First, the provided dataset is quite small and highly imbalanced. The impact of the dataset on model development is visible in the result and analysis section. Secondly, our employed model shows limitations in efficiently detecting the category of passive violence text. Future work should explore advanced techniques and the robustness of passive violence text classification.

Ethics Statement

In this study, the tools and technologies used to perform data analysis and development of the model have been ethically and responsively employed. The aim of our work is to develop a system that detects violence-inciting text for the greater good of our society and culture. As per our belief, knowledge should be shared and we are committed to sharing our findings and contributing to the development of violence-inciting text detection in the Bangla language.

References

Luis Joaquín Arellano, Hugo Jair Escalante, Luis Villaseñor Pineda, Manuel Montes y Gómez, and Fernando Sanchez-Vega. 2022. Overview of da-vincis at iberlef 2022: Detection of aggressive and violent incidents from social media in spanish.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Carlos M Castorena, Itzel M Abundez, Roberto Alejo, Everardo E Granda-Gutiérrez, Eréndira Rendón, and Octavio Villegas. 2021. Deep neural network for gender-based violence detection on twitter messages. *Mathematics*, 9(8):807.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623, Online. Association for Computational Linguistics.

Yun-Fei Jia, Shan Li, and Renbiao Wu. 2019. Incorporating background checks with sentiment analysis to identify violence risky chinese microblogs. *Future Internet*, 11(9):200.

Monther Khalafat, S Alqatawna Ja'far, Rizik Al-Sayyed, Mohammad Eshtay, and Thaeer Kobbaey. 2021. Violence detection over online social networks: An arabic sentiment analysis approach. *iJIM*, 15(14):91.

M. Krendzelak and F. Jakab. 2015. Text categorization with machine learning and hierarchical structures. In *2015 13th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 1–5.

Ho-Suk Lee, Hong-Rae Lee, Jun-U Park, and Yo-Sub Han. 2018. An abusive text detection system based on enhanced abusive and non-abusive word lists. *Decision Support Systems*, 113:22–31.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Hoang Thang Ta, Abu Bakar Siddiqur Rahman, Lotfollah Najjar, and Alexander Gelbukh. 2022. Gan-bert: Adversarial learning for detection of aggressive and violent incidents from social media. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022), CEUR Workshop Proceedings*.

nlpBDpatriots at BLP-2023 Task 1: A Two-Step Classification for Violence Inciting Text Detection in Bangla

Md Nishat Raihan*, **Dhiman Goswami***, **Sadiya Sayara Chowdhury Puspo***, **Marcos Zampieri**
George Mason University
{dgoswam|mraihan2|spuspo|mzampier}@gmu.edu

Abstract

In this paper, we discuss the nlpBDpatriots entry to the shared task on Violence Inciting Text Detection (VITD) organized as part of the first workshop on Bangla Language Processing (BLP) co-located with EMNLP. The aim of this task is to identify and classify the violent threats, that provoke further unlawful violent acts. Our best-performing approach for the task is two-step classification using back translation and multilinguality which ranked 6^{th} out of 27 teams with a macro F1 score of 0.74.

1 Introduction

In an era dominated by social media platforms such as Facebook, Instagram, and TikTok, billions of individuals have found themselves connected like never before, enabling them to swiftly share their thoughts and viewpoints. The growth of social networks provides people all over the world with unprecedented levels of connectedness and enriched communication. However, social media posts often abound with comments containing varying degrees of violence, whether expressed overtly or covertly (Kumar et al., 2018, 2020). To combat this worrisome trend, social media platforms established community guidelines and standards that users are expected to adhere to.[1,2].Violations of these rules may result in the removal of offensive content or even the suspension of user accounts. Given the vast amount of user-generated content on these platforms, manually scrutinizing and filtering potential violence is a very challenging task. This moderation approach is limited by moderators' capacity to keep pace, comprehend evolving slang and language nuances, and navigate the complexity of multilingual content (Das et al., 2022). To address this issue, several social media platforms turn to AI and NLP models capable of detecting inappropriate content across a range of categories such as aggression and violence, hate speech, and general offensive language (Zia et al., 2022; Weerasooriya et al., 2023).

The shared task on Violence Inciting Text Detection (VITD) (Saha et al., 2023a) aims to categorize and discern various forms of communal violence, aiming to shed light on mitigating this complex phenomenon for the Bangla speakers. For this task, we carry out various experiments presented in this paper. We employ various models and data augmentation techniques for violent text identification in Bangla.

2 Related Work

Violence Identification in Bangla Several works have been done on building datasets similar to this task and training models on those data. Such datasets include the works of (Remon et al., 2022; Das et al., 2022), which mostly gather data by social media mining. However, most of the datasets are comparatively small in size. One of the larger datasets is prepared by Romim et al. (2022), which consists of 30,000 user comments from YouTube and Facebook, annotated using crowdsourcing.

While most works focus primarily on the datasets, they also present some experimental analysis. Das et al. (2022) evaluates transformer-based models like m-BERT, XLM-RoBERTa, IndicBERT, and MuRIL. XLM-RoBERTa excels with ample training and MuRIL performs well in joint training, while m-BERT and IndicBERT show proficiency in zero-shot scenarios. However, the most notable work here is done by Jahan et al. (2022) who introduces BanglaHateBERT, a re-trained BERT model for abusive language detection in Bangla. It is trained on a large-scale Bangla offensive, abusive, and hateful corpus. The authors collect and annotate a balanced Bangla hate speech dataset and use

*These three authors contributed equally to this work.

[1] https://transparency.fb.com/policies/community-standards/hate-speech

[2] https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 179–184

it to pretrain BanglaBERT. The proposed model, BanglaHateBERT, outperforms other BERT models and CNN-based models in detecting hate speech on benchmark datasets.

Related Shared Tasks Zampieri et al. (2019, 2020) organized OffensEval, a series of shared tasks identifying and categorizing offensive language in tweets organized at SemEval 2019 and 2020. At OffensEval, participants trained a variety of models ranging from machine learning to deep learning approaches. While BERT and other transformed dominated the leaderboard in 2020, systems' performance in 2019 was more varied with traditional ML classifiers and ensemble-based approaches achieving competition performance along with deep learning approaches. Another shared task, MEX-A3T track at IberLEF 2019 (Aragon et al., 2019), focused on author profiling and aggressiveness detection in Mexican Spanish tweets. Additionally, Modha et al. (2021) presents an overview of the HASOC track at FIRE 2021 for hate speech and offensive content detection in English, Hindi, and Marathi, where the highest accuracy is achieved on the Marathi dataset.

3 Dataset

The VITD shared task (Saha et al., 2023b) provides the participants with a Bangla dataset including 2700 instances for training and 1330 instances for development. The blind test set contains 2016 instances. The dataset (Saha et al., 2023a) has been annotated using three labels: Non-Violence, Direct-Violence, and Passive-Violence. This three-class annotated dataset differs from similar datasets where a binary annotation is used (Romim et al., 2022; Wadud et al., 2021). The data distribution per label is shown in Table 1.

Label	Train	Dev	Test
Non-Violence	51%	54%	54%
Passive-Violence	34%	31%	36%
Direct-Violence	15%	15%	10%

Table 1: Label-wise data distribution across training, development, and test datasets.

4 Methodologies

4.1 Models

Statistical ML Classifiers In our experiments, we use statistical machine learning models like Logistic Regression and Support Vector Machine using TF-IDF vectors.

Transformers We test multiple transformer models pre-trained on Bangla. Our initial experiments include Bangla-BERT (Kowsher et al., 2022) which is only pre-trained on Bangla corpus. We fine-tune the model on the train set and evaluate it on the dev set with empirical hyperparameter tuning. We then use multilingual transformer models like multilingual-BERT (Devlin et al., 2019) and xlm-roBERTa (Conneau et al., 2020), which are pre-trained on 104 and 100 different languages respectively, including Bangla. We also do the same hyperparameter tuning with both models. Lastly, we use MuRIL (Khanuja et al., 2021), another transformer pre-trained in 17 Indian languages including Bangla.

Task Fine-tuned Models We use BanglaHateBERT (Jahan et al., 2022) as a task fine-tuned model which is developed on existing pre-trained BanglaBERT (Kowsher et al., 2022) model and retrained with 1.5 million offensive posts.

Prompting We prompt gpt-3.5-turbo model (OpenAI, 2023) from OpenAI for this classification task. We use the API to prompt the model, while providing a few examples for each label and ask the model to label the dev and test set.

4.2 Data Augmentation

Given the relatively small size of the VITD dataset, we implement a few data augmentation strategies to expand its size. First, we use Google's Translator API (Google, 2021) to translate the train and dev set to 3 other languages that are very similar to Bangla (Hindi, Urdu, and Tamil). Bangla, Hindi, Urdu belong to Indo-Aryan language branch and Tamil from Dravidian language brach, though, all of these languages have cultural interaction in south-east asian region. The native speakers of these languages live in closer geographic proximity. Moreover, these languages have similar morphosyntactic features. So, translating Bangla text to those languages do not hamper structural and grammatical integrity of the sentences. Therefore, we combine these new synthetic datasets with the original train dataset and finetune the multilingual transformer models on them.

The second approach to augment the dataset is back translation. We again use the Translator API to translate the original train and dev set to a few

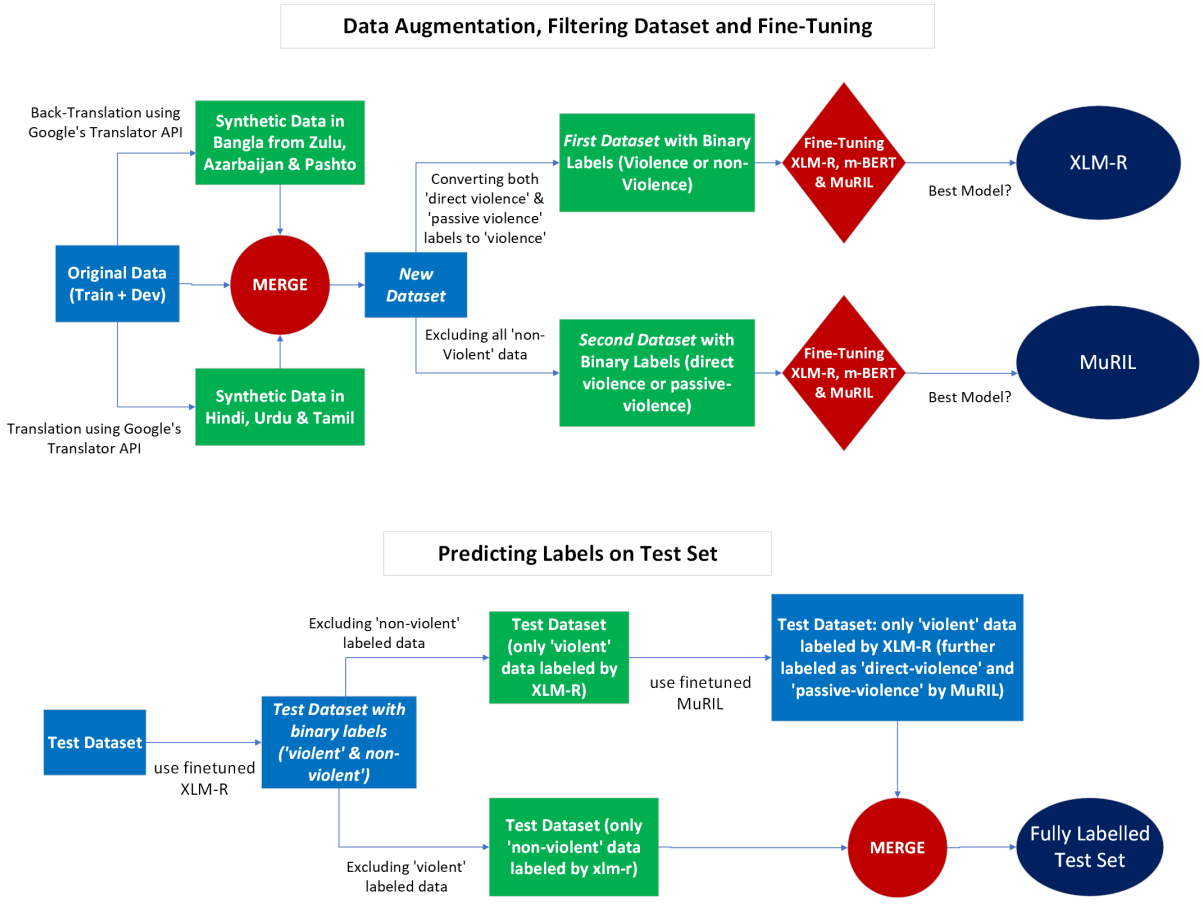


Figure 1: Two-step Classification with Data Augmentation

different low-resource languages like Zulu, Pashto, and Azarbijani as the intermediary language for back translation, in order to add more context. Zulu is from Niger-Congo, Pashto is Indo-Iranian and Azabijani is from Turkic language family. As these languages does not have any cultural interaction with Bangla, back translating from these languages will make three additional version of same sentences with versatility. Then we combine these data with the original dataset. We observe that xlm-roBERTa produces a better macro F1 than the first approach, but still the same as it was on the original data, 0.73.

4.3 Two-step Classification with Data Augmentation

Finally, we combine the two dataset augmentation techniques discussed previously. After combining the synthetic data with the original train set, we have a *New Dataset* that is 7 times the size of the original train set. We generate two different datasets using this *New Dataset*. For the *First Dataset*, we convert all the labels in the *New Dataset* to either Violent (1) or non-Violent (0). And for the *Second Dataset*, we only keep the violent data (both Direct and Passive) from the *New Dataset*.

We finetune mBERT, MuRIL and xlm-roBERTa on both binary labeled *First Datatset* and *Second Dataset* and save their model weights. xlm-roBERTa outperforms the other two when fine-tuned the *First Dataset* and *MuRIL* outperforms the other two when fine-tuned on the *Second Dataset*. For the test set, we first use the finetuned xlm-roBERTa to label the whole dataset as either violent or non-violent data. We then separate all the data from the test set that are labeled as 'violent' by the finetuned xlm-roBERTa model and use the fine-tuned MuRIL model to predict the 'active violence' and 'passive violence' labels. Finally, we merge this with all the 'non-violent' labeled datasets from the first step. Thus, we get all the predicted labels for the test set using 2-step classification by two fine-tuned models. The whole procedure is demonstrated in Figure 1.

5 Results and Analysis

5.1 Results

At the start of the shared task, three baseline macro F1 scores have been provided by the organizers. For BanglaBERT, XLM-R and mBERT, the provided baselines are 0.79, 0.72, and 0.68 respectively. The results of our experiments are shown in Table 2.

Models	Dev	Test
Logistic Regression	0.55	0.56
Support Vector Machine	0.61	0.63
BanglaBERT	0.66	0.67
mBERT	0.71	0.67
MuRIL	0.81	0.72
XLM-R	0.79	0.73
BanglaHateBERT	0.59	0.60
GPT 3.5 Turbo	0.46	0.43
XLM-R (Self-transfer Learning)	0.79	0.72
XLM-R (Multilinguality)	0.78	0.72
XLM-R (Back Translation)	0.77	0.73
XLM-R, MuRIL (Two-step)	**0.84**	**0.74**

Table 2: Dev and test macro F-1 score for all evaluated models and procedures.

Among the statistical machine learning models, we use logistic regression and support vector machine. For logistic regression, we achieve a macro F1 score of 0.56 and for the support vector machine the F1 is 0.63. For transformer-based models, we use BanglaBERT, mBERT, MuRIL and XLM-R where we get the best F1 score of 0.73 by XLM-R. Task fine-tuned model BanglaHateBERT scores 0.60 macro F1.

A few shot learning procedure is used by using GPT3.5 Turbo. We give a few instances of each label as prompt and got 0.43 F1 which is significantly lower than our other attempted approaches. This is because GPT3.5 is still not enough efficient for any downstream classification problem in Bangla like this shared task.

We also perform some customization in our approach instead of directly using the existing models. We use transfer learning. Instead of using the basic idea of transfer learning by fine-tuning a model with a larger dataset of the same label, we translate the train set to English with Google Translator API and used XLM-R on that data. Then we use that finetune model and perform the same procedure over the actual Bangla train set. We refer this procedure as *self-transfer learning* and the F1 score from this procedure is 0.72.

Introducing multilinguality to many downstream tasks proves to be effective. So we also opt for this procedure by translating the train data using Google Translator API to Hindi, Urdu, and Tamil as they are grammatically less diverse and vocabulary is close in contact among the native speakers of these languages. That is how we make the size of our train set three times higher than the original one and got a 0.72 F1 score.

On the other hand, we use Zulu, Azerbaijan, and Pashto - 3 very diverse languages from Bangla for back translation. So, we also get the size of our train set three times higher than the original Bangla one with significantly different translations for each instance. And we get a 0.73 F1 score for that.

Moreover, we use a two-step classification with the data achieved by multilinguality and back translation. Along with these data, we also merge our original Bangla train set. Then, we perform two separate streams of classification. At first, instead of direct and passive violence, we convert them as violence and finetune by XLM-R, mBERT, and MuRIL to classify violence and non-violence where XLM-R performs the best. Then we use the same procedure with the same models to classify direct and passive violence from the merged labels of violence where MuRIL performs the best. Following this procedure, we achieve our best macro F1 score of 0.74 for this shared task.

5.2 Analysis

In terms of text length, the model attains a perfect macro F1 score of 1.000 for texts of 10 words or fewer but struggles with longer texts, evidenced by a macro F1 of only 0.329 for texts of 500-1000 words (Figure 2, Table 3). Though, it maintains respectable F1 scores for text lengths commonly encountered in the dataset, future work should focus on enhancing F1 score for texts with direct violence content.

Text Length	Macro F1	Count	Percentage
(0, 10]	1.000	1	0.050
(10, 20]	0.836	34	1.687
(20, 50]	0.820	528	26.190
(50, 100]	0.736	632	31.349
(100, 200]	0.673	571	28.323
(200, 300]	0.606	156	7.738
(300, 500]	0.627	80	3.968
(500, 1000]	0.329	14	0.694

Table 3: Performance analysis based on text length.

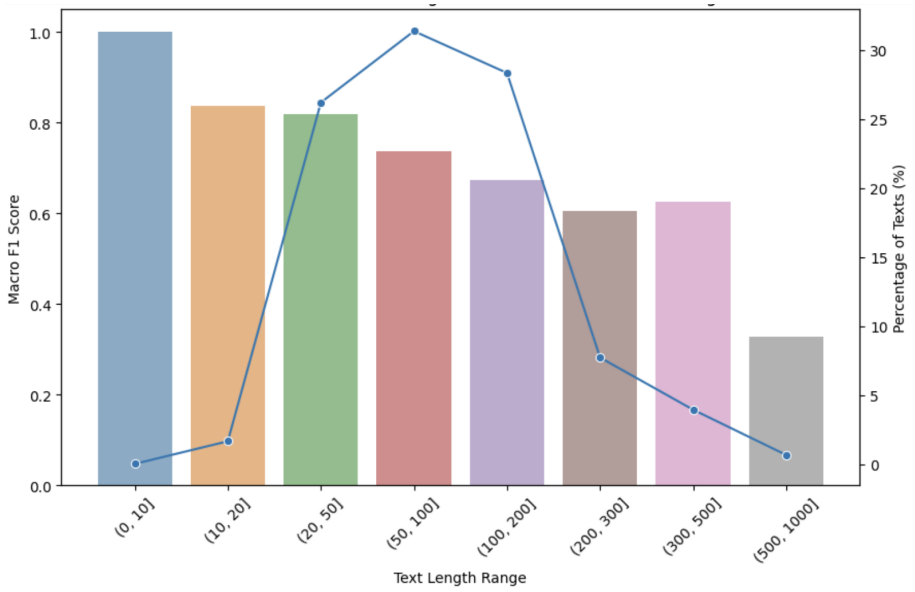


Figure 2: Performance analysis based on text length.

Our model is tasked with categorizing text into one of three labels: non-offensive, direct violence, and passive violence. The confusion matrix, displayed in Figure 3, depicts the performance of the model across these categories. It's pivotal to recognize that in our task, an ideal model would demonstrate high precision and recall across all three labels.

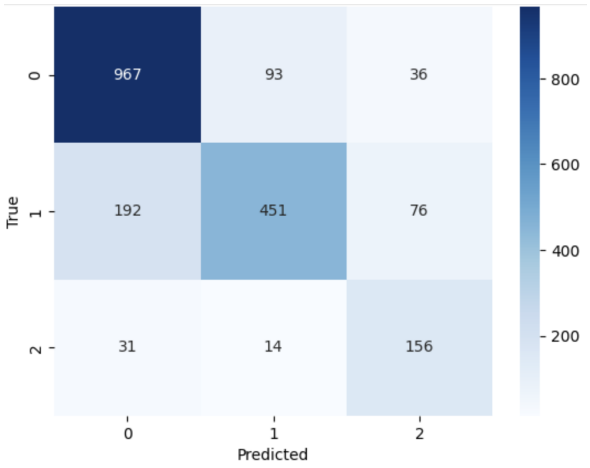


Figure 3: Confusion Matrix

The model categorizes text into non-violence (label 0), passive violence (label 1), and direct violence (label 2) with an overall macro F1 score of 0.74. It particularly excels in identifying non-violence texts. It also demonstrates aptitude in recognizing passive violence texts. However, it faces challenges in the realm of direct violence.

6 Conclusion

In this paper we described the nlpBDpatriots approach to the VITD shared task. We evaluated various models on the data provided by the shared task organizers, namely statistical machine learning models, transformer-based models, few shot prompting, and some customization with transformer-based models with multilinguality, back translation, and two-step classification. We show that the two-step classification procedure with multilinguality and back translation is the most successful approach achieving a macro F1 score of 0.74.

Our two-step approach towards solving the problem presented for this shared task shows promising results. However, the relatively small size of the dataset made it difficult for the other pre-trained models to learn informative features that would help them perform classification. Also, the dataset contains three imbalanced labels making it easy for the models to overfit. Our approach with data augmentation and two-step classification generates good results, but it is still below one of the three baseline results announced by the organizers prior to the start of the competition.

Acknowledgment

We would like to thank the VITD shared task organizing for proposing this interesting shared task. We further thank the anonymous reviewers for their valuable feedback.

References

Mario Aragon, Miguel Angel Carmona, Manuel Montes, Hugo Jair Escalante, Luis Villaseñor-Pineda, and Daniela Moctezuma. 2019. Overview of mex-a3t at iberlef 2019: Authorship and aggressiveness analysis in mexican spanish tweets. In *Proceedings of IberLEF*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Mithun Das, Somnath Banerjee, Punyajoy Saha, and Animesh Mukherjee. 2022. Hate speech and offensive language detection in bengali. In *Proceedings of AACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Google. 2021. Google cloud translation api documentation. Accessed: 2023-08-28.

Md Saroar Jahan, Mainul Haque, Nabil Arhab, and Mourad Oussalah. 2022. BanglaHateBERT: BERT for abusive language detection in Bengali. In *Proceedings ResTUP*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

M Kowsher, Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, 10:91855–91870.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of TRAC*.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2020. Evaluating aggression identification in social media. In *Proceedings of TRAC*.

Sandip Modha, Thomas Mandl, Gautam Kishore Shahi, Hiren Madhu, Shrey Satapara, Tharindu Ranasinghe, and Marcos Zampieri. 2021. Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages and conversational hate speech. In *Proceedings of FIRE*.

OpenAI. 2023. Gpt-3.5 turbo fine-tuning and api updates. Accessed: 2023-08-28.

Nasif Istiak Remon, Nafisa Hasan Tuli, and Ranit Debnath Akash. 2022. Bengali hate speech detection in public facebook pages. In *Proceedings of ICISET*.

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2022. Bd-shs: A benchmark dataset for learning to detect online bangla hate speech in different social contexts. In *Proceedings of LREC*.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of BLP*.

Md Anwar Hussen Wadud, Md Abdul Hamid, Muhammad Mostafa Monowar, and Atif Alamri. 2021. L-boost: Identifying offensive texts from social media post in bengali. *Ieee Access*, 9:164681–164699.

Tharindu Cyril Weerasooriya, Sujan Dutta, Tharindu Ranasinghe, Marcos Zampieri, Christopher M Homan, and Ashiqur R KhudaBukhsh. 2023. Vicarious offense and noise audit of offensive speech classifiers: Unifying human and machine disagreement on what is offensive. In *Proceedings of EMNLP*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of SemEval*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

Haris Bin Zia, Ignacio Castro, Arkaitz Zubiaga, and Gareth Tyson. 2022. Improving zero-shot cross-lingual hate speech detection with pseudo-label fine-tuning of transformer language models. In *Proceedings of ICWSM*.

Score_IsAll_You_Need at BLP-2023 Task 1: A Hierarchical Classification Approach to Detect Violence Inciting Text using Transformers

Kawsar Ahmed, Md Osama, Md Sirajul Islam, Md Taosiful Islam, Avishek Das
and **Mohammed Moshiul Hoque**
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology, Chattogram-4349, Bangladesh
{u1804017, u1804039, u1804116, u1804041}@student.cuet.ac.bd
{avishek, moshiul_240}@cuet.ac.bd

Abstract

Violence-inciting text detection has become critical due to its significance in social media monitoring, online security, and the prevention of violent content. Developing an automatic text classification model for identifying violence in languages with limited resources, like Bangla, poses significant challenges due to the scarcity of resources and complex morphological structures. This work presents a transformer-based method that can classify Bangla texts into three violence classes: direct, passive, and non-violence. We leveraged transformer models, including BanglaBERT, XLM-R, and m-BERT, to develop a hierarchical classification model for the downstream task. In the first step, the BanglaBERT is employed to identify the presence of violence in the text. In the next step, the model classifies stem texts that incite violence as either direct or passive. The developed system scored 72.37 and ranked 14th among the participants.

1 Introduction

Social media and the internet have become crucial components of daily interactions. They can quickly spread information to millions of people. Thus, identifying and categorizing aggressive texts on social media is paramount in maintaining online safety, fostering positive digital interactions, and preventing dissemination of harmful or offensive content (Sharif and Hoque, 2022). Real-world problems like relational anger or even violence are significant problems since threats and insults made online can occasionally result in actual hurt. To keep us secure and calm, we must address this issue because it can make an impact in the short term and also in the long term on the victims (Ta et al., 2022). Different regions' governments try to prevent violations and ensure the safety of the nation's citizens due to social media (Kumar et al., 2021).

The BLP Shared Task 1, Violence Inciting Text Detection (VITD), was launched to address this problem (Saha et al., 2023a). This work presents us with the challenge of devising effective methods to identify diverse types of violent content within the text. The primary objective is to detect and avoid violence from internet remarks. The data used for this task was gathered from YouTube comments on violent incidents that have taken place in the Bengal region (Bangladesh and West Bengal) over the past ten years. We have tried to solve this problem of violence-inciting text detection with two significant contributions.

- Employed a hierarchical classification approach for detecting and classifying violent texts using transformer-based models.
- Explored the model's efficacy in detecting and categorizing violence-inciting texts through the developed model.

2 Related Work

Detecting violence-inciting text has become increasingly crucial in natural language processing. Numerous studies have already focused on identifying hate speech and aggression on social media comments (Badjatiya et al., 2017). Mustakim et al. (2022) employed classify emotions in Tamil text XLM-R model obtained the highest macro f1-score of 0.33. Riza and Charibaldi (2021) detected emotions in Twitter text using the LSTM and achieved an accuracy of 73.15% with both Word2Vec and FastText embeddings. This corpus was used in the DA-VINCIS (Ta et al., 2022) for detecting aggressive and violent incidents on Spanish social media. To train users' tweets on their text embeddings from previously learned transformer models, they employed a multi-task learning network and achieved the best f_1 of 74.80%. Plaza-Del-Arco et al. (2021) applied the transformer-based model to identify hate speech in Spanish tweets. Sharif et al. (2020) proposed a machine learning-based model that classifies Bangla texts into non-suspicious and

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 185–189

suspicious categories. This work attained the highest accuracy (84.57%) for the SGD classifier with TF-IDF features.

Sharif and Hoque (2020) developed a corpus containing 2000 texts and used several machine learning techniques (such as LR, NB, SVM, KNN, and DT) to classify the suspicious Bangla texts, where LR gained the best performance (accuracy=92%). Hossain et al. (2022) proposed a dataset (MUTE) containing 4158 memes with Bangla captions for identifying hateful memes, and they obtained the maximum f_1-score of 0.672 with the VGG16+Bangla-BERT model. Sharif et al. (2022) introduced a Bangla aggressive text dataset (M-BAD). Using a transformer-based technique (Bangla-BERT), they achieved top scores of 92% in identifying aggressive texts. A recent study (Sharif and Hoque, 2022) introduced a Bangla aggressive text dataset (BAD). Using a weighted ensemble of m-BERT, distil-BERT, Bangla-BERT, and XLM-R, they achieved top scores of 93.43% (coarse-grained) and 93.11% (fine-grained) in identifying and categorizing aggressive Bangla texts. As far as we are concerned, the research has yet to be conducted on identifying and classifying violence-inciting texts in Bangla. This work exploited a transformer-based model to detect violence-inciting texts and classify them into direct, passive, and non-violence targets.

3 Task and Dataset Descriptions

The task organizer developed a benchmark corpus for the shared task 1 (Saha et al., 2023a). To perform the violence-inciting text classification, this task developed a dataset called *Violence Inciting Text Detection (VITD)* corpus[1] consisting of 6046 texts and 20199 unique words. This task focuses on classifying Bangla texts inciting violence into three categories: direct (DVio), passive (PVio), and non-violence (NVio). The definition of each class is illustrated in the following:

- **Direct violence (DVio):** This category encompasses texts that explicitly convey threats, thereby falling under the umbrella of direct violence.
- **Passive violence (PVio):** This violence pertains to texts that use abusive or derogatory language.
- **Non-violence (NVio):** This class is characterized by discussions conducted through texts that do not involve any form of violence in their content.

The VITD dataset (Saha et al., 2023b) was divided into training (2700 texts), validation (1330 texts), and test sets (2016 texts) for training and evaluation purposes. Table 1 shows the summary of the dataset statistics.

Table 1: Distribution of the dataset, where $\mathbf{W}_T$ denotes the total words.

Classes	Train	Valid	Test	$\mathbf{W}_T$
DVio	389	196	201	13071
PVio	922	417	719	38959
NVio	1389	717	1096	53838
Total	2700	1330	2016	105868

The dataset contains uneven distribution among the classes. The direct (contained 786 texts) and the passive (2058 texts) classes have fewer samples than the non-violence class (3202 samples). The maximum length of the data is 110 words, whereas the minimum and average data length are one and 18 words, respectively.

4 Methodology

This work exploited three pre-trained transformer-based models, XLM-R, BanglaBERT, and m-BERT, for classifying violence inciting text in Bangla. Specifically, we have used the 'xlm-roberta-base' (Conneau et al., 2019), 'csebuetnlp/banglabert' (Bhattacharjee et al., 2022) and 'bert-base-multilingual-cased' (Devlin et al., 2018) from Huggingface transformers[2] library and fine-tuned on the dataset. Figure 1 illustrates the schematic process of the proposed system.

4.1 Training

Instead of using the direct ternary classification method, we have used a hierarchical classification approach. In the first step, we split the dataset into two classes: 'violence' and 'non-violence.' The 'violence' class included text related to 'DVio' and 'PVio.' We finetuned 'Model 1' to differentiate between 'violence' and 'non-violence' classes. In the second step, we used the samples related to 'Direct violence' and 'Passive violence' to finetune 'Model 2'. All model's hyperparameters are tuned with the

[1] https://github.com/blp-workshop/blp_task1

[2] https://huggingface.co/docs/transformers/index

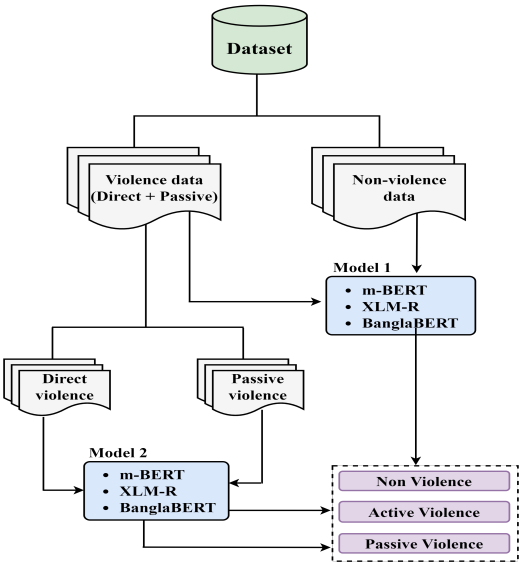


Figure 1: Schematic process of the proposed system.

training dataset. Table 2 shows the tuned hyperparameters of employed models. For BanglaBERT, we set a learning rate of 5e-05 for both Models 1 and 2. We executed training for Model 1 over eight epochs with a batch size of 32, while for Model 2 used five epochs with a batch size of 8.

4.2 Testing and Prediction

If Model 1 classified a text as a 'violence' category, then Model 2 determines whether it is DVio or PVio. This hierarchical process helps us understand different aspects of violent text more effectively. Finally, the results from these two steps have been merged to get the final evaluation score. The predictions of Models 1 and 2 can be expressed by Eqs.1-2.

$$Y_{i_{logits}} = BERT(x) \tag{1}$$

$$M_{i_x} = \frac{e^{Y_{i_{logits_x}}}}{\sum_{p=1}^{p+1} e^{Y_{i_{logits_x}}}} \tag{2}$$

$$if\ M_{1_{(X=Vio)}}:$$
$$Prediction(X) := M_{2_{(X=DVio\ or\ PVio)}}$$

$$else:$$
$$Prediction(X) := NVio$$

The BERT model analyzes the input text x, yielding a result called Y_{logits}. We used a classification head that classifies the Y_{logits} using the softmax activation function into violence (Vio) and non-violence (NVio) classes. M1 (Model 1) represents the probability of violence (Vio) or non-violence (NVio). Subsequently, M2 (Model 2) assesses the likelihood of direct violence (DVio) and passive violence (PVio) within the subset categorized by M1 as violence (Vio). This two-step process helps refine the classification of violence in the text.

5 Results

The assessment of the models' performance relies on the macro F1-score (MF1) as a primary metric. In addition, we incorporated precision (P) and recall (R) metrics for analysis. Table 3 represents the performance of the employed models.

The evaluation encompassed BanglaBERT, XLM-R, and mBERT models in single-step multiclass classification. Among these models, the BanglaBERT achieved the highest macro f_1 (MF1) score, reaching 56.45. BanglaBERT emerged as the top-performing model in the hierarchical framework, surpassing all others with an impressive MF1 score of 72.37. Additionally, it is worth highlighting that the results in the hierarchical approach demonstrated a remarkable improvement of almost 28% over the single-step method.

5.1 Error Analysis

An extensive error analysis has been conducted, offering both quantitative and qualitative assessments. This in-depth examination furnishes valuable insights into the operational efficacy of the proposed model. We conducted a comprehensive quantitative error analysis on the proposed model, employing the confusion matrix depicted in Figure 2.

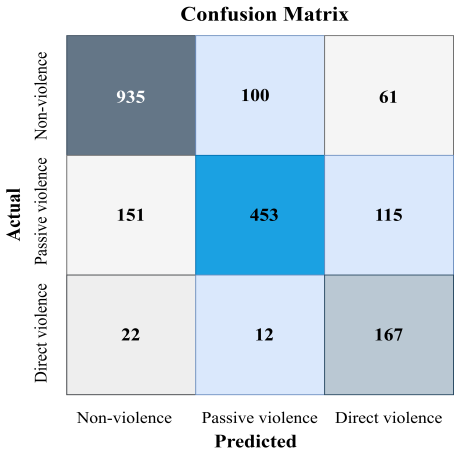


Figure 2: Confusion matrix of the top-performing model (BanglaBERT).

The proposed model misclassified 151 instances

Table 2: Summary of tuned hyper-parameters

Hyperparameters	XLM-R		BanglaBERT		m-BERT	
	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
LR scheduler	Linear	Linear	Linear	Linear	Linear	Linear
Learning rate	2e-05	3.20e-05	5e-05	5e-05	1e-05	1e-05
Epochs	10	4	8	5	5	5
Batch size	16	32	32	8	16	16

Table 3: Performance comparison of various models on the test set.

Approach	Classifier	P	R	MF1
Single-step	m-BERT	61.51	54.36	56.11
	XLM-R	45.03	48.92	46.65
	BanglaBERT	78.12	55.81	**56.45**
Hierarchical	m-BERT	61.52	65.73	61.90
	XLM-R	66.60	69.83	67.62
	BanglaBERT	71.08	77.13	**72.37**

of PVio as NVio and 115 instances of PVio as DVio. Additionally, the model erroneously labeled 100 NVio texts as PVio. These findings shed light on a notable difficulty faced by the model in distinguishing between PVio and NVio. We posit that the primary contributing factor to this challenge could be the class imbalance nature within the dataset.

Figure 3 illustrates a few predictions by the proposed model.

Text Sample	Predicted	Actual
Sample1: মাইজদী - চৌমুহনী - ফেনী মন্দিরে হামলা নিয়ে রিপোর্ট করুন। (Report on the attack on Maizdi - Chaumuhuni- Feni temple.)	NVio	NVio
Sample2: বিবিসি হলো সত্য কে বিনষ্টকারী আর মিথ্যা কে গ্রহণকারী।(The BBC is the destroyer of truth and the acceptor of falsehood.)	PVio	PVio
Sample3: বুধবার কি তোরা মারা গেছিলি বিবিসি বাংলা!(Did you die on Wednesday BBC Bangla!)	NVio	PVio
Sample4: আমরা হিন্দু রা কুরআন পূজা করি না। এটা সম্পূর্ণ চক্রান্ত, সঠিক বিচার চাই (We Hindus do not worship the Quran. This is a complete conspiracy, we want a fair trial.)	PVio	NVio
Sample5: শিক্ষা প্রতিষ্ঠান এ হিজাব নিশিদ্ধ হোক (Hijab should be banned in educational institutions)	DVio	DVio

Figure 3: Few instances of the predicted results generated by the proposed model.

Notably, the proposed model accurately forecasts text samples 1, 2, and 5, aligning with their labels. In contrast, text samples 3 and 4 are challenging as they are not accurately classified. Text sample 3 is erroneously categorized as NVio when its actual class is PVio, while text sample 4 is misclassified as PVio instead of its actual class, NVio. These prediction disparities may be attributed to class imbalance concerns, mainly stemming from the limited number of DVio instances, totaling just 201 samples within the dataset.

6 Conclusion

This paper developed a transformer-based model to address the task of identifying and classifying violence-inciting texts in Bangla. The experimental investigation demonstrated that the BanglaBERT model outperformed the other transformer models (XLM-R and mBERT) by obtaining the highest macro f_1-score (0.72). We plan to investigate the task with the advanced transformer-based model (such as GPT). Additionally, we aim to explore various ensemble techniques of transformers to enhance the model's performance.

Limitations

Model 1 should better identify violence-inciting texts in the proposed two-step hierarchical approach. The success of the entire system hinges directly on the performance of Model 1. If Model 1 fails to deliver accurate results, it will inevitably lead to subparity of the overall system performance. This dependency on Model 1 underscores the critical nature of achieving optimal performance at the initial classification stage, as any shortcomings will adversely affect the outcomes of the developed approach. This limitation emphasizes the need for continuous refinement and enhancement of Model 1 to ensure the effectiveness of the suggested hierarchical system. A fundamental weakness of the proposed solution stems from the imbalanced dataset, with relatively small instances of direct violence (DVio). This imbalance may have influenced the prediction disparities. Additionally, variations

in vocabulary and context within DVio texts, compared to the majority class (NVio), could have contributed to these prediction anomalies. It is worth noting that the dataset's limited size is another constraint.

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 759–760.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. Mute: A multimodal dataset for detecting hateful memes. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 32–39.

Ritesh Kumar, Bornini Lahiri, and Atul Kr Ojha. 2021. Aggressive and offensive language identification in hindi, bangla, and english: A comparative study. *SN Computer Science*, 2(1):26.

Nasehatul Mustakim, Rabeya Rabu, Golam Md. Mursalin, Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. CUET-NLP@TamilNLP-ACL2022: Multi-class textual emotion detection from social media using transformer. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 199–206, Dublin, Ireland. Association for Computational Linguistics.

Flor Miriam Plaza-Del-Arco, M Dolores Molina-González, L Alfonso Ureña-López, and María Teresa Martín-Valdivia. 2021. A multi-task learning approach to hate speech detection leveraging sentiment analysis. *IEEE Access*, 9:112478–112489.

M Alfa Riza and Novrido Charibaldi. 2021. Emotion detection in twitter social media using long short-term memory (lstm) and fast text. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, 3(1):15–26.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Omar Sharif and Mohammed Moshiul Hoque. 2020. Automatic detection of suspicious bangla text using logistic regression. In *Intelligent Computing and Optimization*, pages 581–590, Cham. Springer International Publishing.

Omar Sharif and Mohammed Moshiul Hoque. 2022. Tackling cyber-aggression: Identification and fine-grained categorization of aggressive texts on social media using weighted ensemble of transformers. *Neurocomputing*, 490:462–481.

Omar Sharif, Mohammed Moshiul Hoque, A. S. M. Kayes, Raza Nowrozy, and Iqbal H. Sarker. 2020. Detecting suspicious texts using machine learning techniques. *Applied Sciences*, 10(18).

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2022. M-BAD: A multilabel dataset for detecting aggressive texts and their targets. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 75–85, Dublin, Ireland. Association for Computational Linguistics.

Hoang Thang Ta, Abu Bakar Siddiqur Rahman, Lotfollah Najjar, and AF Gelbukh. 2022. Multi-task learning for detection of aggressive and violent incidents from social media. In *Proceedings of the 2022 Iberian Languages Evaluation Forum, IberLEF*.

Mavericks at BLP-2023 Task 1: Ensemble-based Approach Using Language Models for Violence Inciting Text Detection

Saurabh Page*, **Sudeep Mangalvedhekar***, **Kshitij Deshpande***, **Tanmay Chavan*** and **Sheetal Sonawane***
Pune Institute of Computer Technology, Pune
{saurabhpage1,sudeepm117,kshitij.deshpande7,chavantanmay1402}@gmail.com,
sssonawane@pict.edu

Abstract

This paper presents our work for the Violence Inciting Text Detection shared task in the First Workshop on Bangla Language Processing. Social media has accelerated the propagation of hate and violence-inciting speech in society. It is essential to develop efficient mechanisms to detect and curb the propagation of such texts. The problem of detecting violence-inciting texts is further exacerbated in low-resource settings due to sparse research and less data. The data provided in the shared task consists of texts in the Bangla language, where each example is classified into one of the three categories defined based on the types of violence-inciting texts. We try and evaluate several BERT-based models, and then use an ensemble of the models as our final submission. Our submission is ranked 10th in the final leaderboard of the shared task with a macro F1 score of 0.737.

1 Introduction

In today's digital age, numerous social platforms play an important role in connecting individuals around the world. However, certain malicious elements resort to using these platforms to instigate riots, protests, and disturbances that lead to violence. The online posts and comments involve direct threats pertaining to resocialization, vandalism, and deportation while indirect threats involve derogatory language and abusive remarks. These texts which are thought to be a potential reason for instigating violence are called violence-inciting texts. Classifying them has become a major challenge and various techniques are used to implement it. These applications can be used to monitor social media websites and take precautions to avoid any mishaps. Thus the task boils down to text classification wherein we need to label such texts into predefined categories.

The shared tasks involve performing sentiment analysis and text classification. The BLP Workshop offers two shared tasks namely, Violence Inciting Text Detection (VITD) (Saha et al., 2023a) and Sentiment Analysis of Bangla Social Media Posts (Hasan et al., 2023). Our team, under the name *Mavericks* contested in the VITD task under the Codalab username *kshitij*. Our paper illustrates work on the VITD task where we have to classify text into predefined categories of violence. The dataset consists of text in the Bangla language with a length of up to 600 words.

Transformer-based models (Vaswani et al. (2023)) such as BERT (Devlin et al. (2019)) have brought revolution in NLP-related tasks and have proved their worth by attaining state-of-the-art (SOTA) results on several benchmarks (Lan et al., 2020). Large Language Models (LLMs) are increasingly used for text classification tasks (Liu et al., 2019). We use several transformer-based pre-trained models to achieve higher performance. Furthermore, we use ensembling techniques to produce better results. We present our results after experimenting with several models and ensembling techniques.

2 Related Work

Pang et al. (2002) considers classifying documents by overall sentiment and not just by topic. The three machine learning methods - Naive Bayes, Maximum Entropy Classification, and Support Vector Machines did not perform well on sentiment analysis. Warner and Hirschberg (2012) describes the definition of hate speech as the collection and annotation of hate speech corpus along with a mechanism for detecting some commonly used methods of evading common "dirty word" filters. Hammer (2014) automatically detects threats related to violence using machine learning methods. 24,840 sentences obtained from YouTube comments were manually annotated and were used to

*Equal contribution

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 190–195

train and test the machine learning model. They suggest that the features that combine main words and the distance between those in the sentence attain the best results.

Hassan et al. (2016) provides a textual dataset in Bangla and Romanized Bangla language which can be directly used for sentiment analysis. The dataset was tested using Long Short Term Memory (LSTM) a type of Deep Recurrent Model. Two types of loss functions are used- binary cross-entropy and categorical cross-entropy. Emon et al. (2019) used Linear Support Vector Classifier (LinearSVC), Logistic Regression (Logit), Random Forest (RF), Artificial Neural Network (ANN), and Recurrent Neural Network with an LSTM cell. A Deep-learning-based algorithm using RNN beats all other algorithms by gaining the highest accuracy 82.20%.

In 2017, "Attention is all you need"(Vaswani et al. (2023)) introduces the concept of Transformers which transformed the Natural Language Processing (NLP) landscape. The paper introduced the concept of self attention. In 2019, a new language model called BERT (Bidirectional Encoder Representations from Transformers) was put forward by Devlin et al. (2019). BERT is designed to pre-train deep bidirectional representations from the unlabelled text by joint conditioning on both the left and right context in all layers. Pre-trained BERT can be used for numerous tasks including text classification by fine-tuning it.

Nuryani et al. (2023) proposes a BERT-based method for Aspect-based Sentiment Analysis that can identify and handle conflicting opinions. The method achieves better results on three-class and four-class classification tasks. Sarker et al. (2022) performs sentiment analysis of book reviews in Bangla. A dataset consisting of 5189 reviews was produced by crawling data. An investigation of several deep neural network models and three transformer models is performed. XLM-R outperforms all models, achieving a weighted F1-score of 88.95% on the test data. Anan et al. (2023) performs sarcasm detection using BERT and achieved 99.60% accuracy. A new dataset "BanglaSarc", consisting of comments from Facebook and YouTube was used. Prottasha et al. (2022) utilizes a deep integrated model "CNN-BiLSTM" for enhanced performance of decision-making in text classification.

Dataset	Number of Samples
Training	2700
Development	1300
Testing	2016

Table 1: Dataset statistics.

3 Data

We use the *Vio-Lens* dataset provided by Saha et al. (2023b) for the task. The dataset consists of YouTube comments related to nine violent incidents in the Bengal region (Bangladesh and West Bengal) within the past ten years. The comments are in the Bangla language with a length of up to 600 words. The dataset consists of two attributes: text, and label. The "text" column contains comments while the "label" column contains three values 0, 1, and 2 representing Non-Violence, Passive Violence, and Direct Violence respectively. The training dataset consists of 2700 samples out of which approximately 15% depict direct violence, 34% portray passive violence and the remaining 51% represent non-violent instances. The development dataset consists of 1330 samples out of which approximately 15% illustrate direct violence, 31% depict passive violence and the remaining 54% represent non-violent instances. The test dataset provided at the time of evaluation consists of 2016 samples as seen in Table 1.

4 System

This shared task discusses the problem of Violence Inciting Text Detection. This issue falls under the category of classification, for which transformer-based models have seen extensive application and have demonstrated outstanding performance. As a result, we use and experiment with a variety of such models and ensembling techniques in our research. In the section below, the approaches have been briefly discussed.

4.1 BERT-based Models

Khanuja et al. (2021) discusses how even the state-of-the-art models do not perform satisfactorily well in Indian languages and summarises the gaps found. To mitigate these gaps, they propose their model "MuRIL"[1] which is trained in 16 different Indian languages and English. As we deal with the Bangla

[1]Model link: https://huggingface.co/google/muril-base-cased

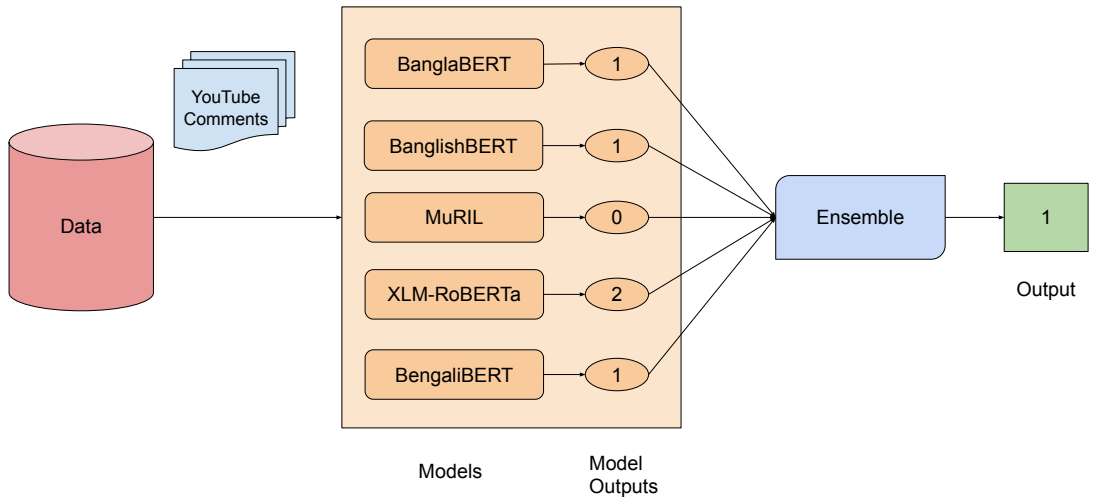


Figure 1: System Architecture

language in this task, MuRIL is specifically relevant. It is trained on two learning objectives, first - Masked Language Modeling, and second - Translation Language Modeling. The model has 236M parameters and a vocabulary of 197285.

Joshi (2022) states that even though multilingual BERT models are suitable for very low-resource languages, models trained on a single language outperform it when sufficient resources for a language are available. Based on this assertion, they propose several models for different languages. BengaliBERT[2] is of specific interest to us. Existing multilingual models are fine-tuned on the Bangla language corpus to create this model.

Conneau et al. (2019) demonstrates how cross-lingual understanding can be improved by pre-training multilingual models on a large scale. XLM-RoBERTa[3] is pre-trained on 2.5TB of filtered CommonCrawl data, which included 100 different languages. It is trained with the multilingual Masked Language Modeling objective. We use the base-sized model in our experiments, XLM-RoBERTa-base which has 270M parameters.

4.2 ELECTRA-based Models

In Bhattacharjee et al. (2022), authors propose training Large Language Models on a dataset specifically tailored for pre-training transformer models useful for Natural Language Processing tasks in the Bangla language. The authors observe that instead of the Masked Language Modeling(MLM) pre-training approach used to train BERT-based models, using ELECTRA and its Replaced Token Detection (RTD) objective provides significant performance improvements while at the same time using significantly less compute power for pre-training. Two Large Language Models are pre-trained namely BanglaBERT[4] and BanglishBERT[5].

The dataset used for pre-training these models was collected by crawling 110 Bangla websites. The total size of the dataset is 27.5GB consisting of 5.25 million documents.

BanglaBERT, introduced in Bhattacharjee et al. (2022), is trained using the ELECTRA pre-training approach consisting of a 12 layer Transformer encoder with 768 embedding size and 12 attention heads. The batch size used is 256 and it is trained for a total of 2.5M steps.

BanglishBERT introduced in Bhattacharjee et al. (2022), is a bilingual model trained on Bangla and English data. It acts as the generator model in the pre-training phase of the ELECTRA approach. BERT pretraining corpus is used along with Bangla data which is upsampled to have equal participation of both languages.

BanglaBERT outperforms other multilingual models such as mBERT and XLM-R (base) on

[2]Model link: https://huggingface.co/l3cube-pune/bengali-bert

[3]Model link: https://huggingface.co/xlm-roberta-base

[4]Model link: https://huggingface.co/csebuetnlp/banglabert

[5]Model link: https://huggingface.co/csebuetnlp/banglishbert

Model	Pre-Training Approach	Macro F1-Score
BanglaBERT	**ELECTRA (RTD)**	**0.791**
BanglishBERT	ELECTRA (RTD)	0.742
MuRIL	MLM	0.753
XLM-RoBERTa	MLM	0.743
BengaliBERT	MLM	0.739
Ensemble - Hard Voting	**-**	**0.782**

Table 2: Results on the development dataset.

Model	Pre-Training Approach	Macro F1-Score
BanglaBERT	**ELECTRA (RTD)**	**0.733**
BanglishBERT	ELECTRA (RTD)	0.662
MuRIL	MLM	0.720
XLM-RoBERTa	MLM	0.705
BengaliBERT	MLM	0.690
Ensemble - Hard Voting	**-**	**0.737**
Weighted Ensemble	**-**	**0.745**

Table 3: Results on the test dataset.

a Bangla-specific benchmark introduced by the authors - Bangla Language Understanding Benchmark (BLUB). BanglaBERT achieves impressive results while having better convergence and thus being more compute-efficient than other previously pre-trained multilingual models.

The batch size used for training all the models is 16. The learning rate used is $1e$-5. We use the AdamW optimizer and the Cross-Entropy Loss. We train the models for 10 epochs. All of the models we use in the experiments are freely available on HuggingFace. We have tagged the models with their respective HuggingFace model links in the footnotes. We use the tokenizers recommended by the model developers provided along with the HuggingFace models.

5 Ensembling

Ensembling is a technique that combines the results of various models to generate the system's eventual result. Statistical as well as non-statistical methods are used for this purpose. Ensembling is useful as it helps generate results that are better than the results given by the individual models.
Amongst several methods leveraged for ensembling, we use the "hard voting" ensemble technique. In hard voting, the majority vote or the "mode" of all the predictions is selected as the final prediction. It helps improve the robustness of the system and minimizes the variance in the results. The ensembling mechanism is illustrated in figure 1.

In the post-evaluation phase, we experiment with the weighted ensemble keeping in mind the varied performances of the underlying models. We give higher weights to the models which perform better. We experiment with different weights for models and choose the weights which provide the best results. We also explore different subsets of the 5 mentioned models and form an ensemble of the models to generate predictions. However, the ensembles of the subsets did not provide improvements to our system's predictions.

6 Results

This section discusses the findings of our experiments. Table 3 contains our results for the models and ensembles. The macro F1 score is the shared task's official score statistic for the Violence Inciting Text Detection task.

BanglaBERT achieves the best result with a macro F1 score of 0.733 among the individual models as seen in table 3. This performance can be attributed to the fact that BanglaBERT is trained on a carefully curated dataset of the Bangla language, unlike other multi-lingual models such as MuRIL and XLM-RoBERTa whose training corpus consists of numerous other languages. It also uses the ELECTRA approach for pre-training which involves using the Replaced Token Detection (RTD) objective instead of the Masked Language Modeling (MLM) objective used in other multilingual BERT models; this allows BanglaBERT to achieve a better performance whilst also converging faster. The performance of MuRIL and XLM-RoBERTa is limited by the quantity and quality of Bangla text they used in pre-training, although it is worth noting that the models will perform much better in a multilingual setting.

BanglaBERT performs marginally better on the

development dataset as seen in Table 2, than the ensemble of the five mentioned models but underperforms on the test dataset. We can attribute this slight difference to variations in the performance of individual models on different data samples and the ensemble's stable and high performance across different data samples. We chose ensembling as the final approach for our final submission owing to its better generalizability and low variance in its predictions. Our final submission to the task using the hard voting ensembling mechanism achieves a macro F1 score of 0.737.

Our post-evaluation phase experiments yielded better results with the weighted ensembling technique. The weighted ensemble achieves a macro F1 score of 0.745 on the test dataset, thus outperforming the hard voting-based ensembling approach.

7 Conclusion

We present our approach for the shared task in the First Workshop on Bangla Language Processing through this paper. We experiment with several BERT and ELECTRA-based models as a part of our efforts. We observed that the ELECTRA-based BanglaBERT model has the best performance, followed by MuRIL. We can see that the ELECTRAbSased models have similar performances compared to their BERT-based counterparts, despite being smaller in size. Our final submission consists of predictions generated by ensembling the evaluated models and has a macro F1 score of 0.737, placing us tenth on the shared task leaderboard. Our experiments have shed light on several further avenues for improvement. Larger pre-training datasets are required for better low-resource models. More sophisticated ensembling techniques can better utilize the performance of individual models and need to be researched further.

Acknowledgment

We thank the Pune Institute of Computer Technology's Computational Linguistics Research Lab for providing us guidance and support necessary for the work. We are grateful for their help.

Limitations

The models that have been utilized are compute-intensive and thus can pose a challenge in real-world applications. Also, it must be considered that the pre-training and evaluation datasets, although of high quality, might possess certain implicit biases and thus might not fully model real-world situations.

References

Ramisa Anan, Tasnim Sakib Apon, Zeba Tahsin Hossain, Elizabeth Antora Modhu, Sudipta Mondal, and MD. Golam Rabiul Alam. 2023. Interpretable bangla sarcasm detection using bert and explainable ai.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Mubasshir, Md. Saiful Islam, Wasi Ahmad Uddin, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Lagnuage model pretraining and benchmarks for low-resource language understanding evaluation in bangla. In *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL 2022*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Estiak Ahmed Emon, Shihab Rahman, Joti Banarjee, Amit Kumar Das, and Tanni Mittra. 2019. A deep learning approach to detect abusive bengali text. In *2019 7th International Conference on Smart Computing Communications (ICSCC)*, pages 1–5.

Hugo Lewi Hammer. 2014. Detecting threats of violence in online discussions using bigrams of important words. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 319–319.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023. Blp 2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

A. Hassan, M. R. Amin, N. Mohammed, and A. K. A. Azad. 2016. Sentiment analysis on bangla and romanized bangla text (brbt) using deep recurrent models.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Nuryani Nuryani, Ayu Purwarianti, and Dwi Hendratmo Widyantoro. 2023. Identification of conflict opinion in aspect-based sentiment analysis using bert-based method. In *Proceedings of the 2022 International Conference on Computer, Control, Informatics and Its Applications*, IC3INA '22, page 276–280, New York, NY, USA. Association for Computing Machinery.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques.

Nusrat Jahan Prottasha, Abdullah As Sami, Md Kowsher, Saydul Akbar Murad, Anupam Kumar Bairagi, Mehedi Masud, and Mohammed Baz. 2022. Transfer learning for sentiment analysis using bert based supervised fine-tuning. *Sensors*, 22(11).

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Gobinda Chandra Sarker, Kazi Md Sadat, and Aditya Das. 2022. Book review sentiment classification in bangla using deep learning and transformer model. In *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pages 1–6.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26, Montréal, Canada. Association for Computational Linguistics.

VacLM at BLP-2023 Task 1: Leveraging BERT models for Violence detection in Bangla

Shilpa Chatterjee*
IIT Kanpur
shilpa20@iitk.ac.in

P J Leo Evenss*
IIT Kanpur
leoevenss20@iitk.ac.in

Pramit Bhattacharyya*
IIT Kanpur
pramitb@cse.iitk.ac.in

Abstract

This study introduces the system submitted to the BLP Shared Task 1: Violence Inciting Text Detection (VITD) by the VacLM team. In this work, we analyzed the impact of various transformer-based models for detecting violence in texts. BanglaBERT outperforms all the other competing models. We also observed that the transformer-based models are not adept at classifying Passive Violence and Direct Violence class but can better detect violence in texts, which was the task's primary objective. On the shared task, we secured a rank of *12* with macro F1-score of *72.656%*.

1 Introduction

In the age of digital empowerment, microblogging sites and social media have ushered in a new era of unfettered expression, providing a global stage for individual voices to be heard like never before. However, this newfound freedom of speech has a darker side, one characterized by the rampant spread of hate speech, cyberbullying, and the toxic dissemination of prejudice across various online platforms. As the digital landscape evolves, so too does the challenge of striking a balance between enabling free expression and curbing the rising tide of online hostility. In this digital dichotomy, the need for innovative solutions to detect and combat hate speech in multiple languages has never been more pressing.

While significant progress has been made in identifying hate speech in languages with more resources, Bangla, despite being spoken by nearly 230 million people across the globe and characterized by its linguistic richness and diversity, faces a substantial shortage of computational resources, language models, annotated datasets and efficient methodologies needed for effective natural language processing(NLP) tasks. Transformer-based models that provide state-of-the-art results in various downstream tasks in European languages lag for Bangla (Bhattacharyya et al., 2023). In this paper, we tried to analyze the impact of transformer-based models on detecting violent inciting text (Saha et al., 2023b) (Saha et al., 2023a), specifically aiming to categorize communal violence on social media platforms in the Bangla language worldwide. BanglaBERT outperforms all the other competing models with a macro F1-score of 72.65% which helped us to secure a rank of *12* on the shared task. We observed that the transformer-based models misclassify Passive Violence as Direct Violence but there performance enhances in detecting violence in texts.

2 Related Works

Numerous methods have been proposed to effectively detect offensive and hateful statements across various platforms, primarily relying on traditional machine learning (ML) techniques, which heavily depend on manual feature engineering. However, ML-based approaches exhibit lower accuracy and also need to improve on scalability issues (Karim et al., 2020). In contrast, methods based on neural networks, particularly deep neural networks (DNNs), have the capability to learn more abstract features directly from raw text.

Prominent DNN architectures, including convolutional neural networks (CNN), long short-term memory (LSTM)(Staudemeyer and Morris, 2019), and gated recurrent unit (GRU) (Zhang et al., 2018), have their advantages. Some approaches have amalgamated CNN and LSTM into a unified network known as convolutional LSTM (ConvLSTM) (Karim et al., 2020). These hybrid models (Karim et al., 2020) have demonstrated superior classification accuracy compared to only neural networks. Additionally, pre-trained word embeddings, such as fastText (Grave et al., 2018) and Word2Vec (Mikolov et al., 2013), have been employed in conjunction with CNN, LSTM, or GRU

*These authors contributed equally to this work

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 196–200

in recent years (Zhang et al., 2018). It's important to note that the majority of these methods have primarily been designed for well-resourced languages like English. Consequently, research in NLP for many underresourced languages, such as Bangla, is still in its early days.

In recent times, language models based on transformers, such as Bidirectional Encoder Representations from Transformers (BERT) built on attention mechanism and the Robustly Optimized Pretraining Approach (RoBERTa) (Liu et al., 2019), have achieved remarkable success in a multitude of natural language processing (NLP) tasks emerging as a natural and highly effective option for addressing the challenges in low-resource languages like Bangla. Other transformer-based language models such as GPT (Brown et al., 2020), DistilBERT (Sanh et al., 2019), ALBERT (Lan et al., 2019), ELECTRA (Clark et al., 2020) has also been proposed for Bangla. For Indian languages, including Bangla, multilingual BERT such as XLM-R (Conneau et al., 2020), multilingual BERT (mBERT) (Pires et al., 2019), IndicBERT (Kakwani et al., 2020) and MuRIL (Khanuja et al., 2021) are available. BanglaBERT (Bhattacharjee et al., 2022), BanglishBERT (Bhattacharjee et al., 2022), sahajBERT (Diskin et al., 2021) are BERT models made specifically for Bangla. BanglaBERT outperforms all other transformer-based models. BanglaBERT was trained on an extensive 40GB dataset derived from various internet sources, such as news articles, web discussions, blogs, government publications, TED Talks, subtitles, newspapers, and articles by crawling data from the web.

This naturally led us to choose BERT over other techniques, and we anticipated that incorporating additional training data could further enhance our approach.

3 DataSet

The dataset provided for BLP Shared Task 1 (Saha et al., 2023b) comprises YouTube comments primarily from social media discussions related to the nine most significant violent incidents in the Bengal region (encompassing Bangladesh and West Bengal) within the past decade. This dataset is characterized by its content being in the Bangla language, with individual comments extending up to 600 words. The dataset is categorized into three classes. They are:

1. **Direct Violence:** Explicit threats directed towards individuals or communities, including actions such as killing, rape, vandalism, deportation, desocialization (threats urging individuals or communities to abandon their religion, culture, or traditions), and resocialization (threats of forceful conversion) falls under this category. Earliest detection of direct violence is crucial because of its potential to yield severe consequences in future.

2. **Passive Violence:** Derogatory language, abusive remarks, slang targeting individuals or communities and justification for violence fall under this category.

3. **Non-Violence:** General conversational topic not involving any form of violence falls under this category.

The training dataset comprises 2,700 samples, with an allocation of around 15% for direct violence, 34% for passive violence, and the remaining 51% for non-violence instances. In the development dataset, which includes 1,330 samples, 15% pertain to direct violence, 31% to passive violence, and 54% to non-violence occurrences.

4 Dataset Preparation

We used the pre-trained models to train on the given dataset and then evaluate the test data provided. Since the training dataset only had around 2.7K sentences, we augmented the training dataset by integrating an additional dataset obtained from (Karim et al., 2020), consisting of 30,000 examples, with 10,000 categorised as violence. Our approach involved annotating these 10,000 hate speech examples into direct and passive violence categories. This annotation was done manually based on our observation from the original dataset, where sentences containing slang were classified as direct violence. We first cleaned different unicode characters to prepare the dataset and removed punctuations from the sentences (Bhattacharyya et al., 2023). Our next task involved identifying the top 200 words that contributed significantly to the direct violence class (directList) and the passive violence class (passiveList) from this original dataset.To form the top 200 word list, we first removed stop words from the original dataset and created a word dictionary consisting of each word and its count of occurrence,one word dictionary

for each of the classes - direct and passive violence. We then selected the top 200 words that contributed to each of the classes. Subsequently, we compiled a corpus comprising slang words in the Bangla language. For each hate speech example within the new additional dataset, we assessed its likelihood of belonging to either the direct violence or passive violence class. If the sentence contained any word from the slang word corpus, it was immediately classified as direct violence. Otherwise, we evaluated each word in the sentence against the lists directList and passiveList. If a word was found in either of these lists, the corresponding score for direct violence or passive violence was incremented by 1. In cases where a word appeared in both directList and passiveList sets, the scores for both classes were incremented by 1.

The final classification was determined based on which class had the higher score. In instances of a tie, we labelled the example as passive violence. In this way, all of the 10,000 hate speech examples were categorised into direct and passive violence.

To maintain the same class proportions as the original dataset, with non-violence at 51%, passive violence at 34%, and direct violence at 15%, we selected an appropriate number of samples from each class in the newly annotated dataset.

5 Baseline Systems

We have used several pretrained models, for the BLP workshop Task 1 (Saha et al., 2023a).

5.1 MuRIL

Multilingual Representations for Indian Languages(MuRIL) supports 16 Indian languages and English and have shown significant gain over mBERT. So we selected MuRIL as our first baseline. We used pretrained MuRIL from Hugging Face.

5.2 IndicBert

IndicBert from Ai4bharat(Doddapaneni et al., 2022) was another choice for a baseline system. IndicBert supports 23 indic languages and english. It is a vanilla BERT which has been trained on IndicCorp with the MLM objective.

5.3 BanglishBert

BanglishBERT (Bhattacharjee et al., 2022) achieves state-of-the-art zero-shot cross-lingual transfer results in many of the NLP tasks in Bangla. It is an ELECTRA discriminator model which has been pretrained with the Replaced Token Detection (RTD) objective on large amounts of Bangla and English corpora.

5.4 BanglaBert

Our next system uses a pretrained BERT model which has been trained specifically on Bangla dataset, which fits perfect for the task in hand.BanglaBert(Bhattacharjee et al., 2022) can be used for a variety of tasks like sentiment classification, Named Entity Recognition,Natural Language Inference etc. and thus served perfect for our Violence Inciting Text Detection (VITD) task.

5.5 Results

We first finetuned the pre-trained models – MuRIL, IndicBert, BanglishBert and BanglaBert on the training dataset provided and evaluated it on the test set. We cross-validated the hyperparameters and found that the best for a batch of 16 with Adam optimizer cross-entropy loss works the best for the task. The learning rate was set at $5 * 10^{-}5$. In addition, we combined the additional dataset to the train set, finetuned the same set of models, and evaluated them using the same metric. Results of these experiments are shown in Table 1. BanglaBERT outperforms all the other models for the task on the original dataset. It is also observed from Table 1 that adding new training points confused models more between Passive and Direct Violence classes, thereby degrading the F1-score.

Dataset Type	Model	F1-Score
Original	MuRIL	0.7026
Original	IndicBert MLM	0.7172
Original	BanglishBert	0.7239
Original	BanglaBert	**0.7265**
Augmented	MuRIL	0.6916
Augmented	IndicBert	0.6723
Augmented	BanglishBert	0.6939
Augmented	BanglaBert	0.7065

Table 1: Macro F1-Score of the models used on Test Data

On analysing the results, we observed multiple instances where the same words were used in various classes. For example, we observed that most sentences where the word “gajaba” was used denoted Passive or Direct Violence in the train set,

however, it denoted Non-Violence in the development set. We also observed that there were occurrences of similar-meaning sentences labelled differently. "mithyā kathā āra kata balabē" and "ēi mēyētā mithyā kathā balachē" are similar meaning sentences but the former is labelled as "Non-Violence" whereas the later is labelled as "Passive Violence". These ambiguous words and sentences resulted in misclassification by the models, degrading the F1-score.

Our analysis also revealed that the models misclassified the Passive Violence class as the Direct Violence class. To confirm this claim, we further conducted an experiment that merged both direct and passive violence into a violence class, mapping it to a binary class classification problem. It was observed that the F1 scores of the models improved significantly. Table 2 reports the macro F1-score of different models on the binary classification task. It can thus be concluded that the models are good at detecting violence, which was the primary objective of the task.

Model	Macro F1-Score
IndicBERT MLM	74.26%
MuRIL	76.35%
BanglaBERT	81.86%

Table 2: Performance of models in detecting violence and non-violence texts.

6 Conclusion

We tried to leverage transformer-based models for violence detection in Bangla for the BLP shared task 1. Our analysis shows that the transformer-based models are not adept at segregating Direct Violence from Passive Violence but are good at detecting violence-inciting text. We would like to develop models that can accurately classify Passive violence in the future.

7 Limitations

Our approach suffers from the lack of a large number of data points essential for transformer-based models. Even after incorporating additional data, we acknowledge that this dataset is relatively small for such a vast language base. A substantial challenge arises from the need for suitable word embeddings for Bangla as used in social media, as the language used in social media significantly diverges from print media, featuring a multitude of misspellings, grammatical errors, and more. Furthermore, a significant portion of users frequently mix both Bangla and English in various contexts. The performance of transformer-based models on such data points lags for a low-resource language like Bangla.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad Uddin, Kazi Mubasshir, Md. Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla. In *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL 2022*.

Pramit Bhattacharyya, Joydeep Mondal, Subhadip Maji, and Arnab Bhattacharya. 2023. Vacaspati: A diverse corpus of bangla literature.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Michael Diskin, Alexey Bukhtiyarov, Max Ryabinin, Lucile Saulnier, quentin lhoest, Anton Sinitsin, Dmitry Popov, Dmitry V. Pyrkin, Maxim Kashirin, Alexander Borzunov, Albert Villanova del Moral, Denis Mazur, Ilia Kobelev, Yacine Jernite, Thomas Wolf, and Gennady Pekhimenko. 2021. Distributed Deep Learning In Open Collaborations. In *Advances in Neural Information Processing Systems*,

volume 34, pages 7879–7897. Curran Associates, Inc.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreyansh Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2022. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. *ArXiv*, abs/2212.05409.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Md. Rezaul Karim, Bharathi Raja Chakravarti, John P. McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In *7th IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA,2020)*. IEEE.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual Representations for Indian Languages.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A Lite BERT for Self-supervised Learning of Language Representations.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Nabeel Mohammed, Sudipta Kar, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter.

Ralf C. Staudemeyer and Eric Rothstein Morris. 2019. Understanding lstm – a tutorial into long short-term memory recurrent neural networks.

Ziqi Zhang, D. Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network.

Aambela at BLP-2023 Task 1: Focus on UNK tokens: Analyzing Violence Inciting Bangla Text with Adding Dataset Specific New Word Tokens

Md Fahim
Center for Computational & Data Sciences
Independent University, Bangladesh
Dhaka-1229, Bangladesh
fahimcse381@gmail.com

Abstract

The BLP-2023 Task 1 aims to develop a Natural Language Inference system tailored for detecting and analyzing threats from Bangla YouTube comments. Bangla language models like BanglaBERT have demonstrated remarkable performance in various Bangla natural language processing tasks across different domains. We utilized BanglaBERT for the violence detection task, employing three different classification heads. As BanglaBERT's vocabulary lacks certain crucial words, our model incorporates some of them as new special tokens, based on their frequency in the dataset, and their embeddings are learned during training. The model achieved the **2nd position** on the leaderboard, boasting an impressive macro-F1 Score of 76.04% on the official test set. With the addition of new tokens, we achieved a 76.90% macro-F1 score, surpassing the top score (76.044%) on the test set.

1 Introduction

In recent times, transformer models have gained popularity through pretraining on diverse text data (Zhang et al., 2022). Pretraining imparts context awareness, linguistic patterns, and word knowledge (Yenicelik et al., 2020). Combining it with fine-tuning vastly outperforms traditional models. These models have a vocabulary from pretraining, representing known words. During fine-tuning the pretrained in a task, there might be some words not in the vocabulary. Tokenizer furhtermore tries to split the word into multiple subwords. Still, if a word is missing or unsegmentable, they use a *[UNK]* token (Nayak et al., 2020).

BanglaBERT (Bhattacharjee et al., 2022) model is one of the most popular pretrained language models in Bangla text classification which is trained on a large corpus of Bangla text. It faces trouble because of *[UNK]* tokens. This problem arises because the pretrianed BERT model sometimes can't give representation to the words that are very important to understand the context. Table 1 represents a small sample taken from the dataset. Here, it can be seen that the significant words which are important for the contextual understanding, assigned as *[UNK]* token in token representation.

Sentence	Tokens
সুন্দর সিদ্ধান্ত নেওয়া হয়েছে	['সুন্দর', 'সিদ্ধান্ত', '[UNK]', '[UNK]']
বয়কট নিউমার্কেট	['[UNK]', 'নিউমার্কেট']
ঢাকা কলেজ মানেই অন্যায়ের বিরুদ্ধে প্রতিবাদ।	['ঢাকা', 'কলেজ', 'মানেই', '[UNK]' 'বিরুদ্ধে', 'প্রতিবাদ','।']

Table 1: The table depicts word-wise tokenization for some example sentences using BanglaBERT tokenizer. Here [UNK] means unknown token

In this study, the main focus was to identify the most frequent words for which *[UNK]* tokens are assigned and add these words to the pretrained vocabulary as a special token. It is shown in the study that this approach improved performance for text classification. To further improve the model's performance, three different classification heads were used. These heads improved the model's prediction by focusing on different words. Three classification heads, along with the proposed approach, achieved better performance than the previous approach.

2 Background

2.1 Task and Dataset Description

The preliminary task of shared task 1 (Saha et al., 2023b) is to detect violence-inciting text (VITD), particularly focusing on identifying threats that could incite further violence. The dataset (Saha et al., 2023a), comprised of Bangla-language YouTube comments, is centered around the top nine violent incidents in the Bengal region over the past decade. This task involves three categories: Direct violence, encompassing explicit

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 201–207

threats to individuals or communities; Passive violence, representing violence through derogatory language, abusive remarks, or slang; and Non-violence, which pertains to content unrelated to violence, including discussions on social rights and general topics. The primary objective is to develop models for automated detection, contributing to online safety and, foster responsible and constructive discourse.

Data Splits	Total Samples	Label wise Samples		
		Label 0	Label 1	Label 2
Train	2700	1389	922	389
Dev	1330	717	417	196
Test	2016	1096	719	201

Table 2: Dataset Statistics for Shared Task 1 (VITD).

The dataset,**Vio-lens**, for shared task 1 (Saha et al., 2023a) comprises texts explicitly associated with violence, each annotated with a corresponding label. Labels are assigned as follows: direct violence is labeled as 2, passive violence is labeled as 1, and non-violence is designated as 1. The dataset statistics are given in Table 2, with mentioning label wise sample sizes for different splits of the dataset.

The dataset (for all splits) contains a significant number of emojis, and these emojis exhibit a notable influence on class dependencies. For instance, violent texts often feature angry emojis.

2.2 Related Work and Baselines

BanglaHateBERT (Jahan et al., 2022), a BERT model for Bangla abusive language detection, was trained on a large-scale offensive text corpus. They also provide a 15K manually annotated Bangla hate speech dataset to the research community. By retraining BanglaBERT (Bhattacharjee et al., 2022) with 1.5 million offensive posts, BanglaHateBERT consistently outperforms the generic pre-trained language model in all datasets. (Mridha et al., 2021) address the rise of offensive Bangla and Banglish texts in online communication. They propose an offensive message detection mechanism using BanglaBERT (Sarker, 2020) combining AdaBoost (Hastie et al., 2009) and LSTM (Hochreiter and Schmidhuber, 1997) models. This proposed **L-Boost** model outperforms baseline classifiers.

Vio-lens dataset provided by the organizer for this shared task introduces a novel dataset related to violence detection tasks in Bangla consists of different forms of violence. The organizer also provided the baselines for this tasks, where fine-tuned model of BERT multilingual base (Devlin et al., 2019) gets 68.19% macro-F1 score where as fine-tuned model of XLM-Roberta (Conneau et al., 2020) gets 72.92% and the fine-tuned model of BanglaBERT (Bhattacharjee et al., 2022) gets 78.79% in the validation test dataset.

3 Method Description

3.1 Adding New Tokens to Vocabulary

Table 1 provides insights into the behavior of the BanglaBERT tokenizer. Notably, the tokenizer occasionally represents highly informative words as *[UNK]* tokens. These words are pivotal for context comprehension, and their conversion to *[UNK]* tokens can pose challenges for the model's predictive capabilities. Identifying the words that result in *[UNK]* tokens from the tokenizer presents a notable challenge. This complexity arises from the tokenizer's utilization of subword tokenization techniques, wherein token lengths may not align with the number of words in a sentence.

To address this challenge effectively, we restrict our analysis to samples without subwords in their tokenization. Within this subset, we extract the specific words that are tokenized as *[UNK]* by the BanglaBERT tokenizer. These words are then ranked by their frequency of occurrence, and we select the top p words to be introduced as new tokens, precisely as special tokens, into the pre-trained vocabulary.

$$Vocab_{new} = Vocab_{original} + \{w_1, w_2, \ldots, w_p\}$$

where $w_1, w_2, \ldots, w_p$ denotes the those frequent words. For a given sentence S, the original tokenization process as:

$$S_{\text{original}} = \{t_1, t_2, \ldots, t_n\}$$

Here, t_i represents the i-th token obtained using the BanglaBERT pretrained tokenizer. While considering new vocabulary for tokenization, the tokenization process becomes:

$$S_{\text{updated}} = \{t_1, t_2, \ldots, t_l\}$$

During the fine-tuning process of the BanglaBERT model, it adapts its internal representations to consider new tokens as valid tokens. This enables the model to encode the

contextual information of words. As a result of fine-tuning, the BanglaBERT model generates contextual embeddings for each token, including those new tokens. These embeddings capture the semantic meaning and context of each token in the input sequence.

3.2 Model Classification Heads for Enhanced Performance

For an input sentence S, we will get $S = \{t_1, t_2, \ldots, t_n\}$ after passing the sentence into the BanglaBERT tokenizer, where t_i represents the i-th token. In this case, we also incorporate the new tokens added to the BanglaBERT tokenizer as we discuss in Section 3.1.

After passing the sentence S through a BanglaBERT model, we obtain contextual representations for each token t_i, denoted as $H = \{h_1, h_2, \ldots, h_n\}$, where h_i represents the contextual representation of token t_i. In this case, we consider the last layer hidden representations of the BanglaBERT encoder.

3.2.1 MLP Head on CLS Token

To obtain a fixed-size representation for the entire sentence for classification, we typically use the special [CLS] token representation h_{CLS}. This representation can be extracted as: $h_{\text{CLS}} = h_1$. Then we pass this h_{CLS} representation through a two-layer Feed Forward Neural Network (FFN) for classification the get class logits.

3.2.2 Dropout-Enhanced CLS Token Head

We introduce an extended classification head, an expansion of the CLS_MLP head detailed in Section 3.2.1. In this variant, we apply dropout to the FFN layer. We explore a set of distinct dropout rates denoted as $D = \{d_1, d_2, \ldots, d_m\}$, where d_i represents the i-th dropout rate. For a given dropout rate d_i, we compute class representations z_i from the MLP. Once we obtain m distinct class representations (logits), we derive the final representation z by averaging these representations, as defined by the equation:

$$z = \frac{1}{m} \sum_{i=1}^{m} z_i$$

3.2.3 Attention-Based Head

For this classification head, once contextual representations H are obtained for a sentence S, an additional attention layer is added to compute learnable attention scores α_i for each token t_i in H, and its calculation is as follows:

$$\alpha_i = \textit{softmax}(W \cdot h_i + b), \quad i = 1, 2, \ldots, n$$

This results in a set of $\textit{attention\_scores} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ corresponding to the tokens in sentence S. These attention scores collectively represent the overall attention distribution across the sentence, indicating the relative importance or relevance of each token to the context of the entire sentence. After finding attention scores for each token, we find the context vector for the sentence S by multiplying the contextual representations of token t_i with its attention score α_i.

$$c = \sum_{i=1}^{n} \alpha_i \cdot h_i$$

After obtaining the context vector, it is further processed through a linear layer to perform the classification task.

4 Result and Analysis

During the development phase, we conducted various experiments, all of which are detailed in Appendix B. The experiment setup and hyperparameter specifics can be found in Appendix A. Our experiments for model selection encompassed a wide range, including machine learning models (SVM, RandomForest, XGBoost) with TF-IDF feature extraction, deep learning models (LSTM, LSTM+Attention), and multilingual Transformer models (mBERT, mDeBerta, XLM-Roberta base). Notably, mDeBerta exhibited superior performance. Additionally, we evaluated two Bangla language models, with the *csebuetnlp-BanglaBERT* model emerging as the top performer. For a concise summary of the experimental outcomes related to model selection, please refer to Table 4.

It's important to note that the LSTM and LSTM+Attention models were trained for 5 epochs, while all transformer-based models underwent finetuning for 3 epochs with the utilization of the [CLS]-based classification head, as detailed in Section 3.2.1.

Table 3 displays the main experimental findings. Each experiment employed a 5-fold cross-validation technique, with the Macro-F1 score as

Techniques	Classification Head	CV Score	Performance Metrics			
			Dev Set		Test Set	
		Macro F1	Accuracy	Macro F1	Accuracy	Macro F1
Without Adding New Tokens	CLS + MLP	79.13	82.63	80.82	80.01	75.96
	Dropouts Enhanced MLP	***79.26***	***81.80***	***80.07***	***80.10***	***76.04***
	Attention Pool	79.76	82.26	80.20	80.36	76.59
Including Dev Dataset	CLS + MLP	80.45	-	-	78.77	74.51
	Dropouts Enhanced MLP	80.49	-	-	78.52	74.14
	Attention Pool	80.29	-	-	79.86	75.80
With Addition of New Tokens	CLS + MLP	79.28	**83.38**	**81.55**	**80.86**	76.76
	Dropouts Enhanced MLP	79.20	82.78	80.94	80.31	76.79
	Attention Pool	79.39	82.86	80.60	80.65	**76.90**

Table 3: Performance of different classification heads at the top of the BanglaBERT with different techniques is shown here. ***Dropouts Enhanced MLP*** without new token addition indicates that the best performing model scores that were submitted to the competition. All the experiments with new token addition techniques and attention-based heads weren't submitted to the competition. But the experiments with new token addition + attention-based classification head give the beat the top leaderboard score, which is marked in underline.

Model Name	Acc ↑	F1 ↑
TF-IDF + SVM	62.26	53.76
TF-IDF +RandomForest	61.88	51.76
TF-IDF + XGBoost	62.83	52.49
LSTM	67.89	62.37
LSTM + Attention	70.76	66.31
mBERT-case	72.33	68.06
mDeBerta-v3 base	75.04	72.27
XLM-Roberta base	73.61	71.68
SagorSarker-BanglaBERT	71.35	67.63
csebuetnlp-BanglaBERT	**81.20**	**79.12**

Table 4: Different Types of Model Performance in Validation (Dev) Dataset. Epoch Size 3

the evaluation metric. Three distinct scenarios were examined with different classification heads, as outlined in Section 3.2. In the first scenario, the use of new token additions (described in Section 3.1) was omitted. In this context, we observed that the *CLS+MLP* configuration outperformed others in the development set. However, the *Dropouts Enhanced MLP* head demonstrated notable improvements, not only in cross-validation scores but also in the test set performance. The Attention based head had showed significant enhancements in both cross-validation scores and test set results, despite a slightly lower performance in the development set. Interestingly, incorporating the development dataset with the training dataset did not yield superior results in the test dataset, despite achieving a better cross-validation score.

Fascinating findings emerged when we incorporated new token additions as new special tokens (described in Section 3.1) into the pretrained BanglaBERT vocabulary. The words that are considered as new tokens are mentioned in Appendix C. In this experiment, we observed approximately a 1% improvement in both dev set and test set performance, measured by accuracy and macro-F1 metrics across all heads. Remarkably, the Attention pool combined with the addition of new tokens yielded the highest macro-F1 score. Notably, the *Dropouts Enhanced MLP* model without the new tokens addition , which secured the **2nd position on the leaderboard**, emerged as the top-performing model among the submissions.

All heads with new token addition and attention pool head without new tokens addition beat the top leaderboard score, which was 76.044% macro-F1 score. Unfortunately, those models weren't submitted during the competition. The models that beat the top leaderboard score are marked as underlined in Table 3.

5 Conclusion

In this study, an analysis is done when we add dataset-specific tokens (most frequent) to the pretrained vocabulary of BanglaBERT for which the BanglaBERT tokenizer gives the *[UNK]* token.

The addition tokens learn their embeddings during the finetuning. From the experiment, it has been seen that the addition of those type of tokens boosts the model's performance. To enhance the model prediction's further, different classification heads are applied.

Limitations and Future Plan

The aforementioned approach, adding dataset-specific most frquent tokens for which the pre-trained tokenizer gives *[UNK]* tokens, helps in this task. A proper investigation is needed to analyse if this approach performs better in some other tasks.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. 2009. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Md Saroar Jahan, Mainul Haque, Nabil Arhab, and Mourad Oussalah. 2022. BanglaHateBERT: BERT for abusive language detection in Bengali. In *Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis*, pages 8–15, Marseille, France. European Language Resources Association.

Muhammad F Mridha, Md Anwar Hussen Wadud, Md Abdul Hamid, Muhammad Mostafa Monowar, Mohammad Abdullah-Al-Wadud, and Atif Alamri. 2021. L-boost: Identifying offensive texts from social media post in bengali. *Ieee Access*, 9:164681–164699.

Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. 2020. Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5, Online. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Arnab Sen Sharma Api, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023a. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Nabeel Mohammed, Sudipta Kar, and Mohammad Ruhul Amin. 2023b. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understading. *textsIGitHub*.

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT capture semantics? a closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162, Online. Association for Computational Linguistics.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*.

A Experimental Setup and Hyperparameters

In this research endeavour, various text preparation procedures, such as eliminating punctuation, emojis, numeric characters, and potential web addresses, were implemented. LSTM and Bert were used as the text encoders. Their generated representation were used as the hidden layer representation for the text. Different models including TF-IDF combinations, LSTM variants and Bert variations were applied on Dev dataset. Then utilizing the representation, different models

including combinations involving TF-IDF, various LSTM variants, and variations of Bert were used in this study.

LSTM-based models including standalone LSTM and LSTM+Attention used embedding dimension of 128 for an embedding layer.The models include hidden dimension of 256, learning rate of 10^{-3}, batch size of 16 for this configuration. As it can be seen from table 7, this batch size bested all other variation for Dev dataset. Thus, batch size 16 was consider for the models to gain the optimal performance for all the models.

For the BERT model used in this study, we utilized the *Bangla BERT* variant that enables us to extract contextual representations and long-term dependency through fine-tuning and pretraining. In this case, the hidden dimension of the BERT model was set to 768. The learning rate for BERT was 2×10^{-5}, maximum token length was 64 and batch size was 16. From table 5, it can be seen that the Bert models outperformed all other variations for token length 64. Therefore, token length 64 was considered.

Both configurations included the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. To ensure robustness, we performed five-fold cross-validation and three different random seeds. Additionally, we set $\lambda = 10$ for all experiments. An ablation study investigating the effect of different λ values is presented in Table. All experiments were conducted using Python (version 3.10) and PyTorch, leveraging the free NVIDIA Tesla T4 GPU available in Google Colab, as well as a single NVIDIA Tesla P100 GPU provided by Kaggle.

B Ablation Study

This section represents the ablation studies performed in this study. Which includes token cutoff analysis, batch size effect analysis and Loss analysis. For each of the analysis we compare the variations in terms of the optimal value of accuracy and F1 score.

B.1 Token Length Effects

In this study, different token length were considered to gain the highest performance for all the performance matrices. Token length 512 showed poor performance for the implemented models. The value of token length 64 outperforms all other variations, while improving the lowest accuracy and F1 score by 1%-2%. Rest of the variations slightly lags behind.

Token Length	Dev Acc ↑	Dev F1 ↑
32	80.15	77.28
64	**80.75**	**78.35**
128	80.32	77.92
256	80.3	77.36
512	79.1	76.72

Table 5: Token Cutoff Experiment of *csebuetnlp/BanglaBert* in Validation (Dev) Dataset.

B.2 Batch Size Influence

Batch size effects were also considered to gain optimal results. Batch size 16 showed superior performance than all other variations. It bested the lowest performance of batch size 64 by 2% in accuracy, and 4% in F1 score. While batch size 8 and 16 closely tails behind in terms of performance matrices.

B.3 Loss Analysis

Loss has significant effects on overall outcome of the study. Thus, a detailed investigation was performed on different variant of loss. Cross Entropy loss bested all other loss variations. The Cross Entropy Loss and Focal loss variation (0.5*Focal + 0.5*CE) performed vary poorly among the losses. Weighted Cross Entropy Loss showed slight improvement. While The standalone Cross Entropy loss, and the combination of Cross Entropy Loss and Focal loss (0.3*Focal + 0.7*CE) showed improvement by 1% to 3% from the lowest values of accuracy and F1 score for all the performance matrices. Finally, the Cross Entropy Loss showed superior performance to all other variations.

C The words which are considered as new tokens

As per discussion in Section 3.1, it is very challenging to figure out the words for which the tokenizer is giving [UNK] tokens. The reason behind this, the tokenizer that BanglaBERT uses sub-

Preprocessing	Dev Acc ↑	Dev F1 ↑
No Preprocessing	80.30	77.92
Removing Punctuation & Emoji's	79.10	76.72
Removing Emoji's Only	78.50	76.23
Removing Punctuation Only	78.95	75.96
Adding Normalizer in the text	81.20	79.12
Adding BN-Unicode Noramlizer	80.15	79.00
Converting Emoji's into Text	80.90	78.44
Adding most frequent [UNK] tokens as new tokenizers	**83.01**	**81.40**

Table 6: Effect of different preprocessing in dev set. For experiment BanglaBERT is used with 4 epochs of training

Batch Size	Dev Acc ↑	Dev F1 ↑
8	79.77	77.59
16	**80.08**	**77.99**
32	80	76.95
64	78.12	73.22

Table 7: Batch Size Effect of *csebuetnlp/BanglaBert* in Validation (Dev) Dataset while Token Length = 64 were considered. Epoch Size 5

word tokenization for which the no. of tokens and white space based basic tokenizer word list of a sentence aren't equal. To takcle this issues we extracted those samples for which token legnth from tokenizer is equal to length white space based basic tokenizer word list. We find only 531 samples considering both train and validation dataset for which the condition is followed. For those 531 samples, the most frequent words for which the BanglaBERT tokenizer gives *[UNK]* tokens. The tokens that are considered as new tokens is shown in Table 8.

D Preprocessing Analysis

In Table 6, several experiment are done with different preprocessing techniques. The table shows that punctuation of emoji's carry some contextual information while classifying the texts. So, removing them didn't help the model. For bangla text, normalizer plays a vital role. Two differnet normalization techniques were experimented where *csebuetnlp/normalizer* proven effective rather than BN-unicode normalizer. Another experiment were done converting emoji's into but it didn't help. Finally, adding most frquent words for which pretrained BanglaBERT gives [UNK] tokens become more helpful for the model.

Word	Count
হয়	21
হয়ে	21
সময়	21
দেওয়া	17
মিডিয়া	16
নিয়ে	14
বয়কট	13
যায়	12
আওয়ামী	11
ভয়	10
বড়	10
হায়রে	9
দেয়া	8
দায়	8
আওয়ামিলীগ	8
হয়েছে	7
দিয়ে	7
এগিয়ে	7

Table 8: The list of words that are considered as new tokens to the model

SUST_Black Box at BLP-2023 Task 1: Detecting Communal Violence in Texts: An Exploration of MLM and Weighted Ensemble Techniques

Hrithik Majumdar Shibu[†], **Shrestha Datta**[†], **Zhalok Rahman**[†]
Shahrab Khan Sami, **Md. Sumon Miah**, **Raisa Fairooz**, **Md Adith Mollah**
Shahjalal University of Science and Technology, Sylhet, Bangladesh
{hrithik11804064, shresthadatta910, rahmanzhalok}@gmail.com
{shahrabkhan6620, iamsumon111, raisafairoozshafa, adibhasan35}@gmail.com

Abstract

In this study, we address the shared task of classifying violence-inciting texts from YouTube comments related to violent incidents in the Bengal region. We seamlessly integrated domain adaptation techniques by meticulously fine-tuning pre-existing Masked Language Models on a diverse array of informal texts. We employed a multifaceted approach, leveraging Transfer Learning, Stacking, and Ensemble techniques to enhance our model's performance. Our integrated system, amalgamating the refined BanglaBERT model through MLM and our Weighted Ensemble approach, showcased superior efficacy, achieving macro F1 scores of 71% and 72%, respectively, while the MLM approach secured the 18th position among participants. This underscores the robustness and precision of our proposed paradigm in the nuanced detection and categorization of violent narratives within digital realms.

1 Introduction

While fostering connections and facilitating information dissemination, social media has inadvertently become a platform for propagating hostility. Such hateful actions, encompassing communal violence, cyberbullying, and social platform attacks disrupt online communities and erode the foundational trust and safety intrinsic to such platforms Romim et al. (2021). By utilizing the latest advancements in artificial intelligence and natural language processing (NLP), we can effectively identify and prevent potential violent incidents, thus creating a safer environment. In this context, we will examine the BLP Shared Task 1: Violence Inciting Text Detection (VITD).

Recent advancements in the field have highlighted the potential of informal text embeddings in enhancing the accuracy of Hate Speech (HS) detection, evidenced by the work of (Romim et al., 2022). Furthermore, the advent of Masked Language Model (MLM) pre-training, exemplified by models such as BERT (Devlin et al., 2018), has revolutionized text classification tasks. The landscape of NLP has been significantly shaped by the adoption of transfer learning in recent years. Pioneering methodologies such as ULMFiT (Howard and Ruder, 2018; Khatun et al., 2020) have demonstrated the superiority of fine-tuning language models over traditional deep learning algorithms, especially when confronted with limited datasets and resources. This paradigm shift is further exemplified by models like BanglaBERT (Bhattacharjee et al., 2021), which builds upon the foundational BERT architecture, benefiting from extensive pre-training on diverse datasets. The burgeoning interest in Bangla text classification has catalyzed the development of several pivotal datasets and transformer-based approaches (Alam et al., 2020; Hasan et al., 2023; Islam et al., 2020), further enriching the ecosystem and setting the stage for our research.

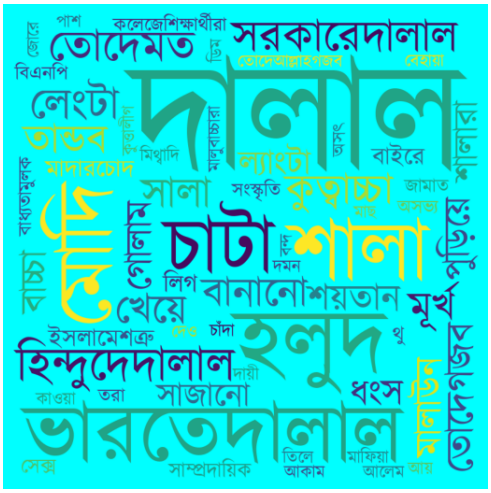


Figure 1: Words after exclusion of words in neutral class and discarding most of the positive and neutral words

Our approach to the VITD task (Saha et al.,

† These authors have equal contributions

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 208–213

2023a) is informed by these advancements, leveraging Transfer Learning (TL) and MLM training to incorporate informal texts into our models. During training, we have used a large volume of similar informal data collected from various domains (Islam et al., 2021; Kabir et al., 2023; Romim et al., 2022) using domain adaptation along with the VITD dataset. Utilizing our approaches, we have gotten better results than the benchmark models.

2 Dataset

The class distribution of training, validation, and test set of the VITD dataset by Saha et al. (2023b), facilitating the main classification task, is shown in Table 1. Each dataset contains three output classes namely Neutral(N), Passive Violence(PV), and Active Violence(AV).

Labels	Train	Validation	Test
Neutral	1,389	717	1,096
Passive Violence	922	417	719
Active Violence	389	196	201

Table 1: Class distribution of VITD datasets

In total, we have 2,700 instances in the final dataset for training and 1,330 instances for the development set. The mean text length of the instances is 17.51 ± 14.4 as shown in Figure 2 and detailed in Table 2.

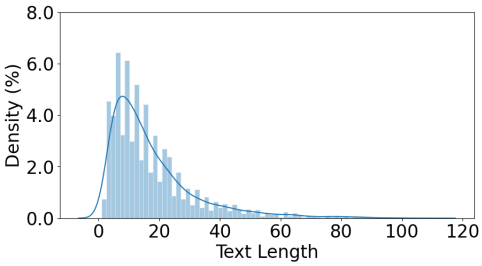


Figure 2: Histogram of text lengths of VITD dataset

Metrics	Values
Maximum Text Length	110
Minimum Text Length	1
Mean	17.51
Standard Deviation	14.4

Table 2: Some relevant metrics related to the length of the VITD dataset texts

In our endeavor to understand the linguistic nuances of the dataset, we constructed a word-cloud Filatova (2016) (as depicted in Figure 1). This was achieved by judiciously excluding words from the neutral class and systematically discarding a majority of the positive and neutral terms. This visualization offers insights into the specific linguistics that warrant detection. A salient observation from our analysis is the dataset's substantial inclusion of informal and colloquial expressions. Notably, such vernacular terms are often absent from the training corpora of widely recognized pre-trained models.

To get a deeper insight into the linguistic traits of the AV and PV classes, all words of the neutral class were excluded from the AV and PV classes. The resultant set of words of AV class and PV class is represented by the closed circular curve on the left and right respectively of the Venn diagram (Figure 3). This AV and PV set consists of 2702 and 8259 words respectively, while the intersection contains 245 words. From the word samples presented in the Venn diagram, the words unique to AV class(excluding the set of PV words from AV set) encompass most of the words that indicate violence of some form, and the words unique to PV class(excluding the set of AV words from PV set) hold most of the words related to dehumanization. While words common to both classes are predominantly linguistically dehumanizing, only a small portion of them consist of violence-inciting words. The ratio of dehumanizing-natured words within the intersection set significantly exceeds the ratio of such words in the exclusive PV class set. In these sets, neutral words also exist in a significant amount.

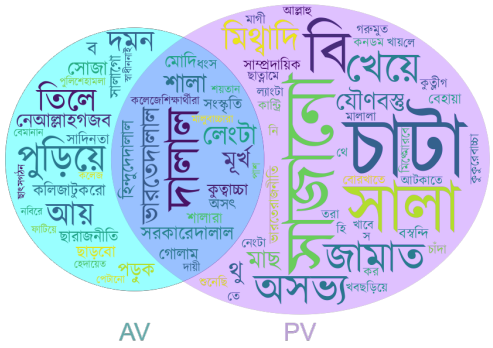


Figure 3: Venn diagram of AV and PV words set excluding neutral class words

3 System description

3.1 Dataset for MLM Training and TL

Task related datasets have been divided into three groups based on their usage, as shown in Table 3.

For MLM training, 9,674 text samples labeled as

'Negative' sentiment from the BanglaBook dataset by Kabir et al. (2023), 6,807 text samples labeled as 'Aggressive' from the BAD dataset by Sharif and Hoque (2022), 24,156 text samples labeled 'Hate Speech' from the BD-SHS dataset by Romim et al. (2022) and negative emotion-sentiment dataset from Alam et al. (2020). This group also includes the VITD (train and dev) dataset. Only the negative class samples have been taken for MLM training.

The BD-SHS containing 50,281 samples labeled 'Hate Speech' or 'Non Hate' has been used for TL.

Used for	Dataset(s)
MLM training	BanglaBook, BAD, BD-SHS, emotion-sentiment
Transfer learning	BD-SHS (with labels)
Fine-tuning	VITD (with labels)

Table 3: Used dataset groups

3.2 Masked Language Model Training

The Masked Language Model (MLM) is a pivotal neural network architecture in NLP that predicts omitted words within sentences. Leveraging hidden tokens minimizes the divergence between predicted and actual words while accounting for bidirectional context. To ensure that the linguistic representations align with specific domains, we have employed domain adaptation techniques to fine-tune the MLM. In our research, we have meticulously adjusted parameters such as learning rates, weight decay, and batch sizes and selectively frozen specific encoder layers for optimization. We have primarily used pre-trained BanglaBERT which is actually the **ELECTRA** model (Clark et al., 2020) for extensive contextual learning through **Masked Language Modeling** from our expansive dataset. In this model, tokens are replaced with feasible alternatives, enabling the model to distinguish between the original and substitute tokens. This discriminator model is quite effective and represents an intriguing development in NLP tasks.

3.3 Fine Tuning Pre-Trained MLM

Leveraging contextual linguistic knowledge, we have fine-tuned the pre-trained ELECTRA model from section 3.2 for improved text classification. Specifically, we froze the Encoder layers of the MLM-trained model to achieve desired classification results. Utilizing the best checkpoint from the pre-trained MLM and minimizing the difference between training and validation loss, we have obtained the highest macro F1 score.

3.4 Transfer Learning from BD-SHS dataset

We employed TL through downstream model training, leveraging the BD-SHS dataset, to train our model on the VITD dataset as our parallel approach for violence detection, which we refer to as TL approach. To address class imbalance, we upsampled the classes PV and AV by iteratively replicating samples until their sizes matched that of class N. To implement TL, initially we finetuned pre-trained BERT-based models with domain-related dataset as described in section 3.4.1. In the next step, as described in section 3.4.2, we further finetuned the model we had trained in the initial step (from section 3.4.1) keeping the embedding layer non-trainable. We utilized the models produced from section 3.4.2 for our validation and test on VITD dataset.

3.4.1 Training Transformers-based Models on BD-SHS Dataset

We have trained the BERT-based models, Monolingual BanglaBERT-base (sagorBERT) (Sarker, 2020), mBERT cased (Devlin et al., 2018), as well as XLM-RoBERTa (Conneau et al., 2019) on BD-SHS dataset keeping training epochs low. For each training epoch, we have randomized the order of our training data and implemented gradient clipping (Pascanu et al., 2013). We fine-tuned the pre-trained BERT variants using the Adam optimizer while limiting the input length to a maximum of 256 tokens. We took outputs experimenting with 1 and 2 layers of multi-head attention, followed by a linear layer as classification head.

3.4.2 Fine-Tuning Models Trained on BD-SHS dataset with VITD Dataset

We have discarded the classification head of the trained models on the BD-SHS dataset and added two tanh-activated nonlinear layers (for sagorBERT) and a linear layer as the new classification head for training on the VITD dataset. In the first training session, we had all the model layers frozen, including the embedding and encoder layers of the models except the classification head, and trained on the VITD dataset. In the second training session, we kept the classification head frozen and unfroze the encoder layers of the previously trained models. We have trained with gradient clipping on the upsampled dataset for both sessions by shuffling the data samples at each epoch.

3.5 Ensemble Approaches

In our study, we have primarily explored two ensemble techniques: Stacking (Wolpert, 1992) and Weighted Ensemble (WE). For stacking, we have incorporated four models: TL-based sagorBERT, mBERT, XLM-RoBERTa, and the MLM-trained BanglaBERT. Utilizing 60% of the VITD dataset's development set, we have trained a deep neural network comprising three non-linear ReLU layers, culminating in a softmax layer. This model was validated against the remaining 40% of the development set and subsequently evaluated on the test set. In our WE approach (Huber and Kim, 1996), we have selected seven models, all evaluated on the VITD dataset:

1. Four models have been trained solely on the training data including TL-based sagorBERT, mBERT, XLM-RoBERTa, and MLM-trained BanglaBERT.

2. Three models have been trained on both the training and validation data encompassing TL-based sagorBERT, mBERT, and XLM-RoBERTa. The optimal hyperparameters for these models were determined through rigorous validation.

To mitigate potential validation data leakage, models trained on both training and validation data were assigned minimal weights. Conversely, the model exhibiting the highest validation macro F1 score was accorded the maximum weight. After experimenting with diverse weight configurations on the validation set, we finalized the weights, opting for the label with the majority consensus.

4 Experimental Setup

We have presented our approach to strengthen a VITD model using a pre-trained MLM which is an ELECTRA model based on Transformers Network (Vaswani et al., 2017) which we have referred to as MLM approach. To facilitate the VITD model, first we have used pre-trained MLM on texts to comprehend contextual representations. The final classification has been done by freezing the 6 encoder layers of the ELECTRA model and fine-tuning the hyperparameters of the model. During both processes, we utilized a learning rate of 2e-5 and ran the model for 10 and 50 epochs respectively in which the epoch with the highest Macro F1-score is stored as the final result. For stacking, we have trained a deep neural network for 21 epochs with a learning rate of 0.03. For WE, we have assigned the fine-tuned BanglaBERT a weight of 3 and other TL-based models a weight of 1. We have used the mini-batch training paradigm for our experiments. Corresponding all the codes are publicly available at this repository.[1]

5 Results

We present our results using the macro F1 score for both the validation and test datasets, as detailed in Table 4. Notably, our SUST_Black Box's approach for BLP-2023 Task 1 achieved the highest macro F1 scores of 0.85 on the validation set and 0.72 on the test dataset. In Table 4, we delineate the methods, models, and their respective performances in terms of the macro F1 score. For the stacking approach, we incorporated TL models and MLM, as discussed in section 3.5. For the Weighted Ensemble (WE) method, TL models trained solely on the training dataset were termed TL models-1. Meanwhile, TL models trained on both the training and validation datasets were denoted as TL models-2. The MLM was assigned a weight of 3, as elaborated in 3.5.

Method	Model	Val	Test
Baseline	BanglaBERT	0.78	0.70
	sagorBERT	0.69	0.63
	mBERT (cased)	0.65	0.63
TL	sagorBERT	0.69	0.65
	mBERT	0.68	0.65
	XLM-RoBERTa	0.67	0.59
MLM	BanglaBERT	0.80	0.71
Stacking	TL models MLM	0.79	0.70
WE	TL models-1 TL models-2 MLM	0.85	0.72

Table 4: Validation and test macro-F1 score of each categorical models

In summary, as depicted in Table 4, our methods, particularly BanglaBERT with our MLM approach and WE, demonstrated superior performance on both validation and test sets. Notably, in both MLM and TL we have fine-tuned and used domain adaptation for linguistic representation. Afterward, we used these models in stacking and WE. In light of this, we discerned that MLM and WE methods spotted an impressive result. From Table 5 we see

[1] Github: https://github.com/Shibu4064/EMNLP

Method	Model	P	R	mF1
TL	sagorBERT	64	66	70
	mBERT	66	67	71
	XLM-RoBERTa	59	65	64
MLM	BanglaBERT	71	76	76
Stacking	TL models MLM	70	75	75
WE	TL models-1 TL models-2 MLM	72	74	76

Table 5: Macro Precision(**P**), Macro Recall(**R**) and Micro F1(**mF1**) score in percentage(%) for each categorical models on test data.

Model	Neutral	Passive	Direct
mBERT(t)	0.76	0.64	0.53
mBERT(t+v)	0.80	0.60	0.55
MLM(t)	0.84	0.67	0.61
RoBERTa(t)	0.73	0.58	0.48
RoBERTa(t+v)	0.77	0.40	0.52
sagorBERT(t)	0.77	0.64	0.53
sagorBERT(t+v)	0.79	0.64	0.53
WE(t+v)	0.84	0.67	0.64

Table 6: Individual class F1 score on test dataset

that MLM and WE methods achieved the highest micro F1 score of 76%, whereas, MLM achieved the best macro recall and WE best macro precision of 76% and 72% respectively. We have also presented individual class F1 scores from different TL and MLM approach models. N, PV and AV (2) these three class F1 scores are proffered in Table 6. Here (t) represents test sets and (t+v) represents both test and validation sets.

6 Discussion

In this section, we present the results of our experiments with MLM and TL methodologies, which have outperformed the base models. The primary reason for this improvement is the inclusion of informal words that were previously absent in the pretraining datasets of the pre-trained models. To further optimize our results, we used ensemble techniques. We prioritized MLM within the Weighted Ensemble (WE) framework by assigning it the highest weight, recognizing its superior accuracy. Interestingly, we found that integer weights of WE predominantly excelled in AV class detection, despite our initial expectations of learned weights from stacking yielding superior outcomes. This also highlights the importance of the inclusion of models in WE trained on both validation and training sets. To improve our outcomes further, we integrated upsampling, which, in certain instances, led to improved outcomes. During training with upsampled data, our approach of freezing the embedding layer throughout the training process and selectively freezing and unfreezing different layers at various stages of training lessens the chance of overfitting. Lastly, being dominated by the majority neutral class, the micro F1 score is considerably higher compared to the macro F1 score. This indicated that, as backed up by individual class F1, the finetuned models were able to classify between neutral and non-neutral classes more rigorously.

7 Conclusion and Future work

Our experiments aimed to explore various approaches to integrating informal words, and we found that the MLM and WE methods performed the best. Our MLM and TL approaches are still unexplored for all BERT baseline models, including exploring based on the same models. Discovering the effects of our approaches and their comparison will lead to promising future research directions and help improve our methods' robustness and scalability. The effect of freezing embedding layers and, selectively freezing and unfreezing other layers on overfitting due to upsampled data still needs in-depth study. As we worked to familiarize pretrained models with the nuances of informal words for the VITD task in Bangla, we hope to contribute to safer online spaces for everyone and unlock new frontiers in NLP.

8 Limitations

Several approaches were applied for the improvement of VITD. However, we encountered challenges such as a highly imbalanced dataset, limited computational resources, and a relatively small dataset size. During MLM training for the MLM approach and downstream model training on the BD-SHS dataset for the TL approach, although increasing the training time helps the models to adapt to the datasets, it also increases the knowledge decay of the models as we are not training with a huge dataset. The initial phases of our approaches also demand a huge amount of data from similar domains. Although freezing parameters reduce the chance of overfitting due to upsampling but the chance still remains.

References

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Bangla text classification using transformers. *arXiv preprint arXiv:2011.04446*.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2021. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *arXiv preprint arXiv:2101.00204*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Olga Filatova. 2016. More than a word cloud. *Tesol Journal*, 7(2):438–448.

Md Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023. Zero-and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv preprint arXiv:2308.10783*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Gary A Huber and Sangtae Kim. 1996. Weighted-ensemble brownian dynamics simulations for protein association reactions. *Biophysical journal*, 70(1):97–110.

Khondoker Ittehadul Islam, Md Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5. IEEE.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. Sentnob: A dataset for analysing sentiment on noisy bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271.

Mohsinul Kabir, Obayed Bin Mahfuz, Syed Rifat Raiyan, Hasan Mahmud, and Md Kamrul Hasan. 2023. Banglabook: A large-scale bangla dataset for sentiment analysis from book reviews. *arXiv preprint arXiv:2305.06595*.

Aisha Khatun, Anisur Rahman, Md Saiful Islam, Hemayet Ahmed Chowdhury, and Ayesha Tasnim. 2020. Authorship attribution in bangla literature (aabl) via transfer learning using ulmfit. *Transactions on Asian and Low-Resource Language Information Processing*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr.

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2022. Bd-shs: A benchmark dataset for learning to detect online bangla hate speech in different social contexts. *arXiv preprint arXiv:2206.00372*.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam. 2021. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*, pages 457–468. Springer.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Omar Sharif and Mohammed Moshiul Hoque. 2022. Tackling cyber-aggression: Identification and fine-grained categorization of aggressive texts on social media using weighted ensemble of transformers. *Neurocomputing*, 490:462–481.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

the_linguists at BLP-2023 Task 1: A Novel Informal Bangla FastText Embedding for Violence Inciting Text Detection

Md. Tariquzzaman, Md. Wasif Kader, Audwit Nafi Anam
Naimul Haque, Mohsinul Kabir, Hasan Mahmud, Md Kamrul Hasan
Systems and Software Lab (SSL)
Department of Computer Science and Engineering
Islamic University of Technology, Dhaka, Bangladesh
{tariquzzaman,wasifkader,audwitnafi,naimulhaque,mohsinulkabir,hasan,hasank}
@iut-dhaka.edu

Abstract

This paper introduces a novel informal Bangla word embedding for designing a cost-efficient solution for the task "Violence Inciting Text Detection" which focuses on developing classification systems to categorize violence that can potentially incite further violent actions. We propose a semi-supervised learning approach by training an informal Bangla FastText embedding, which is further fine-tuned on lightweight models on task specific dataset and yielded competitive results to our initial method using BanglaBERT, which secured the 7th position with an f1-score of 73.98%. We conduct extensive experiments to assess the efficiency of the proposed embedding and how well it generalizes in terms of violence classification, along with it's coverage on the task's dataset. Our proposed Bangla IFT embedding achieved a competitive macro average F1 score of 70.45%. Additionally, we provide a detailed analysis of our findings, delving into potential causes of misclassification in the detection of violence-inciting text.

1 Introduction

This study details our methods and results for the "Violence Inciting Text Detection (VITD)" task (Saha et al., 2023a), aiming to classify texts into three violence categories: Direct Violence, Passive Violence, and Non-Violence with a goal to identify texts that could lead to further violent actions. Unlike hate speech that targets groups based on attributes, violence-inciting texts advocate harm. The misuse of social media, especially in the Bengal Region, has escalated communal violence (Mathew et al., 2018), with hate speech being a primary cause. This task aims to understand and mitigate such violence.

Our study introduces a unique Bangla FastText(IFT) embedding trained on 3.8 million informal Bangla text samples collected from informal data sources such as Facebook and Youtube comments. We combine this with lightweight ML and DL models like Logistic Regression (LR), SVM, LSTM, BiLSTM, and GRU to detect violence-inciting texts and compare the performance with transformer models such as BanglaBERT, mBERT, XLM-RoBERTa. To the best of our knowledge, this is the first attempt to use FastText embeddings with lightweight models for detecting violence inciting texts in Bangla. Such methods have shown potential in various Bangla text classification methods in previous studies (Kowsher et al., 2022). Our contributions can be summarized as follows:

- An informal Bangla FastText(IFT) embedding trained on 3.8 million sample dataset with better vocabulary coverage on VITD dataset (Saha et al., 2023b) than the existing BanglaBERT's vocabulary.
- A cost-effective solution approach incorporating lightweight classification models and the proposed IFT embedding, that offers 17 times faster training and 1.54 times faster inference speed than BanglaBERT, while having only 4% lower macro-f1 score.
- Performance comparison of lightweight models like LR, SVM, LSTM, BiLSTM, GRU using the proposed IFT embedding with transformer models such as BanglaBERT, XLM-RoBERTa and mBERT.
- Analysis of the classification performance of all the models and how well IFT performs in detecting violence inciting text.

Our work is particularly noteworthy for its development of a versatile Bangla informal FastText embedding, which can have broader implications across various domains like Bangla text classification, token classification, sentiment analysis, etc. Both our informal FastText embedding and the

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 214–219

training corpus will be made publicly available to advance Bangla research [1].

2 Related Work

We found several studies that addressed hate speech detection and analysis in under-resourced Bangla language. The concept of utilizing informal word embeddings is derived from the work of Romim et al. (2022) where they discovered that word embeddings generated from informal Bangla texts are quite effective in identifying hate speech in online comments, a finding further reinforced by the work of Karim et al. (2020) using an LSTM model. The potential of developing a Bangla word embedding model from a vast corpus of Bangla news articles and then using these embeddings to classify Bangla document was also discussed in the work of Ahmad and Amin (2016). Romim et al. (2020) presented a hate speech dataset comprising 30,000 user comments, underscoring the efficacy of SVM while observing issues of overfitting in deep learning models when utilizing BengFastText embeddings due to class imbalance. Romim et al. (2022) also introduced a dataset with 50,200 offensive comments, emphasizing linguistic diversity and the challenges of identifying hate speech targets. The study by Islam et al. (2021) focused on sentiment analysis of informally written Bangla texts, emphasizing the challenges posed by this "noisy" text that includes various dialects, spelling errors, and grammatical inaccuracies. Additionally, it offered insights into the classification performance on informal texts using FastText embeddings.Karim et al. (2021) introduced DeepHateExplainer, where they utilized an ensemble transformer model for explainable hate speech detection, achieving an F1-score of 88%, while acknowledging potential overfitting due to limited dataset. Hate speech in romanized Bangla language on social media platforms was studied by Das et al. (2022). While there has been a considerable number of studies conducted for hate speech detection, notably less research has been dedicated to identify text that incites violence in the Bangla language.

3 Task Description

The primary objective of this task is to detect and categorize threats associated with violence, which have the potential to incite further acts of violence. The task features three distinct categories:

- Direct Violence: Explicit threats targeting individuals or communities, including murder, sexual assault, property damage, forced deportation, desocialization, and resocialization.
- Passive Violence: Violence expressed through derogatory language, abusive remarks, slang, or justifications for violence.
- Non-Violence: Content unrelated to violence, including discussions on social rights or general topics.

3.1 Dataset Description

The dataset (Saha et al., 2023b) employed for this task encompasses YouTube comments about the 9 most significant violent incidents occuring in the Bengal region, which includes both Bangladesh and West Bengal, in the last decade. The dataset contains text written in the Bangla language, with comment lengths of up to 600 words, and it is categorized as either Direct violence, Passive violence, or Non-violence. The dataset consists of the columns "text" and "label", where the "text" column contains textual data extracted from social media, while the "label" column assigns each sample a numerical value of 0, 1, or 2, representing non-violence, passive violence, and direct violence accordingly. Table 1 demonstrates a short instance of the dataset.

Label	Category	Example
DV	2	রক্ত যখন দিয়েছি রক্ত আরও দিবো তবুও নিউমার্কেটের আশেপাশে কোনো সাংবাদিকের মাথা না ফাটিয়ে ছাড়বো না ইনশা আল্লাহ!
PV	1	সরকারের সব লোক ভারতের দালাল মনে রাখিছ আল্লাহ ছাড় দেয় কিন্তু ছেড়ে দেয়না
NV	0	একজন বাবা কতোটা অসহায় হলে এই কথা বলতে পারে আল্লাহ তুমি বিচার করো

Table 1: Label Instances of Direct Violence (DV), Passive Violence (PV), and Non-Violence (NV)

[1] https://github.com/Tariquzzaman-faisal/VITD

4 System Description

The System proposed for the VITD shared task is based on IFT embedding that incorporates sub-word information, enabling effective handling of Out-Of-Vocabulary(OOV) words and capturing morphological patterns. We follow a semi-supervised methodology for training where the IFT embedding is created by our collected unlabelled data from social media comments. This embedding is then finetuned on the task specific VITD dataset (Saha et al., 2023b) and incorporated with lightweight models like Logistic Regression (LR), SVM, LSTM, BiLSTM, and GRU models. We carried out extensive experiments to validate the effectiveness of our method and utility of our proposed embedding. Our proposed system is illustrated in Figure 1. The configuration used for LSTM, BiLSTM, and GRU models are included in the Table 3.

4.1 Embedding Dataset Construction

We gather a large informal text dataset of 6.8 million samples from Facebook and YouTube, known sources of Bangla abusive content (Romim et al., 2020). To collect data efficiently, Facepager[2] was employed, using the Facebook Graph API. The preprocessing involves removal of redundant words, symbols, and non-Bangla content, which left us with a streamlined 3.8 million sample dataset. It's coverage on the VITD task's datasets is depicted in Table 2.

Dataset	IFT	BanglaBERT
Train	**58.32%**	35.00%
Dev	**62.45%**	40.49%
Test	**58.35%**	35.82%

Table 2: Vocabulary Coverage on task dataset

$$\text{Coverage} = \frac{|T \cap E|}{|T|} \quad (1)$$

In expression 1, $|T|$ denotes the total count of unique tokens in the task dataset, while $|E|$ represents the dataset of IFT embeddings and BanglaBERT's vocabulary in their respective columns as shown in Table 2. The term $|T \cap E|$ denotes the count of unique tokens common to both datasets. The term "Coverage" represents the proportion of unique tokens in the training dataset covered by the embedding dataset. It is evident that IFT provides better coverage compared to the existing vocabulary of BanglaBERT on this task.

[2] https://github.com/strohne/Facepager

4.2 Experimental Setup

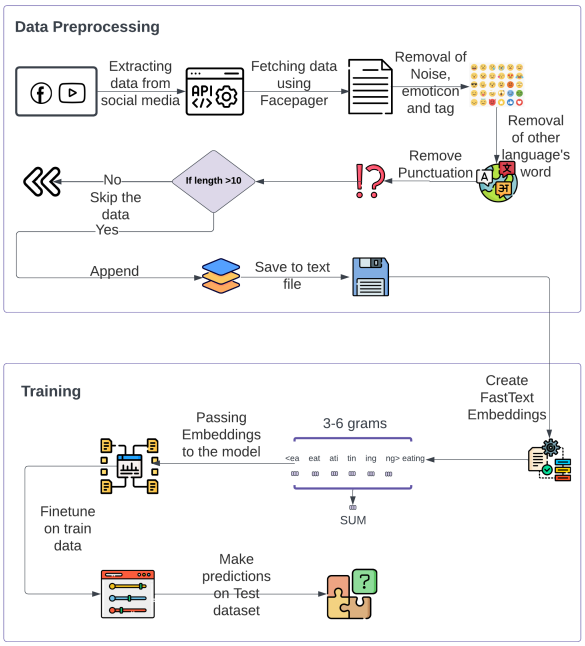


Figure 1: Methodology of the Proposed System

The collected data is used to train a FastText model using a 300-vector length and character n-grams ranging from 3 to 6. The model employs the Continuous Bag of Words (CBoW) algorithm and specifies a minimum word count of 2 for the training procedure. CBoW is chosen over Skip-Gram because it efficiently learns word embeddings from the context in a more computationally efficient manner, making it faster for training on large datasets (İrsoy et al., 2021). Additionally, CBoW tends to perform better on downstream tasks like text classification when contextual information is not as critical. Its simplicity and ability to handle frequent words effectively make it a practical choice for our use case.

During the training process, the FastText model picked up the ability to represent words as continuous vector representations by taking into account the character n-grams that make up individual words as well as information about their context. The model was able to effectively capture the semantic and syntactic subtleties of the language after it leveraged the subword information and contextual signals that were included within the dataset. Significant consideration is given to the settings of the hyperparameters, which helped

to ensure that an optimal configuration is used, which in turn maximized the embedding's quality and performance. The process is shown in Figure 1. After creating the IFT embedding, it is integrated with LR, SVM, LSTM, BiLSTM, and GRU models. Then the models are trained on the labeled data containing non-violence, passive violence, and direct violence. To check the effectiveness of our proposed IFT embedding, we also train a separate version of each of the models with CBoW embedding. Apart from this, all the configurations are kept similar across these models.

Hyperparameter	Value
Max sequence length	256
Batch size	32
Units	150
Dropout	0.3
Learning rate	0.001
Optimizer	Adam
Loss	SCC
Embedding dim	300

Table 3: Hyperparameters of LSTM, BiLSTM and GRU

5 Results and Findings

Table 4 demonstrates the positive impact of the proposed IFT embedding on model accuracy. For comparison, the accuracy of the transformer models is also presented in the same table. To provide a comprehensive validation of this improvement, we assessed the precision, recall, and F1 scores, as detailed in Table 6. The macro F1 score, which gives equal consideration to each class, provides a holistic view of model performance, guaranteeing a fair assessment that accounts for potential dataset variations. Intriguingly, the BiLSTM model's performance not only aligns with the transformer models but even surpasses mBERT and XLM-RoBERTa in macro-f1 score. Among the transformer models, BanglaBERT emerges as a standout performer, showcasing superior accuracy and F1 scores compared to mBERT and XLM-RoBERTa. This underscores the potential of specialized models tailored for specific languages or regions. Our macro F1 score of BanglaBERT improved to 74.6% as shown in Table 6 due to better tuning of the parameters and the highest accuracy score of 78.67% on test dataset.

If we focus on computational efficiency, table 5 shows the capabilities of our BiLSTM+IFT having an impressive 17 times faster training time than BanglaBERT and faster inference by a factor of 1.54. This remarkable speed, combined with competitive accuracy, positions BiLSTM+IFT as a cost-effective alternative for detecting texts that may incite violence. For clarity, our training spanned 6 epochs with 2,700 samples, while inference was executed on 2,016 samples. All tests were uniformly conducted on Google Colab using a T4 GPU.

Model	Without IFT	With IFT
LR	52.48%	70.29%
SVM	55.06%	72.02%
LSTM	**69.47**%	74.50%
BiLSTM	64.38%	**74.55**%
GRU	69.25%	74.45%
mBERT(base)	71.11%	-
XLM-RoBERTa(base)	72.22%	-
BanglaBERT(base)	78.67%	-

Table 4: Accuracy Comparison with and without the Proposed InformalFastText(IFT) Embedding

Model	Training	Inference
BanglaBERT	532.80	18.46
BiLSTM+IFT	**31.23**	**11.98**

Table 5: Speed comparison between BiLSTM+IFT and BanglaBERT in seconds

Key Observations:

- Incorporating IFT embeddings generally improves the performance across models. This is evident from the higher values in the rows with IFT as compared to their counterparts without IFT in table 4.
- BiLSTM with IFT has a macro F1 score of 70.5%, which is comparable to transformer models. Notably, it outperforms mBERT and XLM-RoBERTa, which have macro F1 scores of 65.8% and 67.4% respectively but falls short of BanglaBERT's 74.6%.
- BanglaBERT has the highest macro F1 score of 74.6% among all models, reinforcing its superior performance as observed in the accuracy Table.

Model	Non-violence			Passive Violence			Direct Violence			Macro Average		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LR(CBoW)	55.4	85.5	61.2	43.7	15.9	23.4	10.0	2.9	4.6	36.4	34.8	31.7
LR(IFT)	69.8	89.9	78.6	77.0	46.2	57.7	57.6	49.3	53.1	68.1	61.8	63.1
SVM(CBoW)	54.5	97.0	69.8	49.3	3.2	5.9	22.2	1.0	1.9	39.0	33.7	25.9
SVM(IFT)	68.9	94.1	79.5	81.9	44.8	57.9	78.6	49.6	60.6	**76.5**	62.7	66.0
LSTM(CBoW)	69.2	92.0	79.0	79.9	44.8	57.4	62.4	48.8	54.7	70.5	61.8	63.7
LSTM(IFT)	73.5	89.9	80.9	79.8	55.9	65.7	66.7	56.7	61.3	73.3	67.5	69.3
BiLSTM(CBoW)	54.5	99.4	70.4	47.1	0.0	0.0	0.0	0.0	0.0	33.8	33.5	24.2
BiLSTM(IFT)	76.9	84.1	80.4	74.0	63.0	68.1	62.1	63.7	62.9	71.0	**70.3**	**70.5**
GRU(CBoW)	54.4	99.5	70.3	46.2	0.0	0.0	0.0	0.0	0.0	33.5	33.4	24.0
GRU(IFT)	73.9	90.3	81.3	82.1	52.9	64.4	60.3	64.2	61.2	72.1	69.2	69.3
mBERT	77.4	79.7	78.5	74.6	58.4	65.5	43.3	69.7	53.4	65.1	69.3	65.8
XLM-RoBERTa	80.2	80.8	80.5	74.2	57.2	64.6	44.7	79.6	57.3	66.4	72.5	67.4
BanglaBERT	88.0	82.5	85.1	63.3	82.7	71.7	83.1	56.2	67.1	**78.1**	**73.8**	**74.6**

Table 6: Model Performances with and without the Proposed InformalFastText(IFT) Embedding

- BanglaBERT offers the best accuracy and overall performance, while BiLSTM+IFT presents a compelling case as a cost-effective and efficient alternative, especially for applications where speed is crucial as it is 17 times faster in training and 1.54 times in inference for this particular task.

Our empirical findings indicate that the BiLSTM+IFT model exhibits a significant enhancement in performance upon the incorporation of IFT embeddings. Furthermore, this model not only demonstrates a marked cost-effectiveness compared to transformer architectures like BanglaBERT, mBERT, and XLM-RoBERTa, but it also achieves accuracy metrics that are competitive. This underscores the dual advantage of BiLSTM+IFT: its efficiency in computational resources and its competitive accuracy in the realm of NLP tasks.

Observation: The challenge of distinguishing between passive and direct forms of violence is common across models as depicted in Table 6, likely due to the inherent textual similarity in violent content. Models struggle in these areas both with and without IFT embeddings. Yet, the incorporation of IFT embeddings shows a clear enhancement in classifying more challenging categories, supporting our claims of model performance improvement. The confusion matrix in Figure 2 highlights the predictive capabilities of our best model, BiLSTM, in per class classification and aiding a comprehensive analysis of its strengths and weaknesses in differentiating between violence forms.

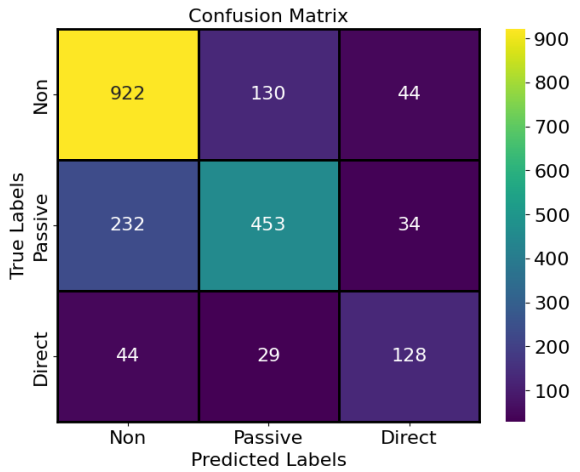


Figure 2: Confusion matrix for BiLSTM+FastText

6 Conclusion

This paper presents a cost-sensitive approach to the detection of violence-inciting text in Bangla using a semi-supervised method. Our results show that applying the proposed IFT embedding to lightweight models produces competitive performance compared to larger transformer models, all while maintaining cost-effectiveness. We believe that enhancing the dataset's size and coverage will lead to improved performance across various aspects when using IFT, thereby broadening the potential applications of our approach to other Bangla text classification tasks.

Limitations

Finding high-quality sources of diverse Bangla hate speech and violence inciting texts was a challenge for us. As a generalized informal embedding dataset, it shows the potential of enhancing the performance of detecting violence inciting texts. However, a larger dataset geared more towards violence inciting texts would yield better results. Furthermore, better bangla text preprocessing tools can also improve the overall scores of all the models. Also, the training data exhibited class imbalance where the neutral label had significantly more samples than the direct label. A more balanced dataset could potentially yield better results.

References

Adnan Ahmad and Mohammad Ruhul Amin. 2016. Bengali word embeddings and it's application in solving document classification problem. In *2016 19th International Conference on Computer and Information Technology (ICCIT)*, pages 425–430.

Mithun Das, Somnath Banerjee, Punyajoy Saha, and Animesh Mukherjee. 2022. Hate speech and offensive language detection in Bengali. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 286–296, Online only. Association for Computational Linguistics.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Md. Rezaul Karim, Bharathi Raja Chakravarthi, John P. McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network.

Md. Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Bharathi Raja Chakravarthi, Md. Azam Hossain, and Stefan Decker. 2021. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language.

Md Kowsher, Md. Shohanur Sobuj, Md Shahriar, Nusrat Prottasha, Mohammad Arefin, Pranab Dhar, and Takeshi Koshiba. 2022. An enhanced neural word embedding model for transfer learning. *Applied Sciences*, 12:2848.

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2018. Spread of hate speech in online social media.

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2022. BD-SHS: A benchmark dataset for learning to detect online Bangla hate speech in different social contexts. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5153–5162, Marseille, France. European Language Resources Association.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam. 2020. Hate speech detection in the bengali language: A dataset and its baseline evaluation.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Ozan İrsoy, Adrian Benton, and Karl Stratos. 2021. Corrected cbow performs as well as skip-gram.

UFAL-ULD at BLP-2023 Task 1: Violence Detection in Bangla Text

Sourabrata Mukherjee[1], Atul Kr. Ojha[2], Ondřej Dušek[1]
[1]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
[2]Insight SFI Centre for Data Analytics, DSI, University of Galway, Ireland
{mukherjee,odusek}@ufal.mff.cuni.cz
atulkumar.ojha@insight-centre.org

Abstract

In this paper, we present UFAL-ULD team's system, desinged as a part of the BLP Shared Task 1: Violence Inciting Text Detection (VITD). This task aims to classify text, with a particular challenge of identifying incitement to violence into Direct, Indirect or Non-violence levels. We experimented with several pre-trained sequence classification models, including XLM-RoBERTa, BanglaBERT, Bangla BERT Base, and Multilingual BERT. Our best-performing model was based on the XLM-RoBERTa-base architecture, which outperformed the baseline models. Our system was ranked 20th among the 27 teams that participated in the task.

1 Introduction

The rapid proliferation of social media platforms has revolutionized the way we communicate, share information, and engage with diverse communities online. However, with this newfound connectivity and freedom of expression, we have also witnessed a troubling trend – the weaponization of social media for the incitement of violence. The Bengal region, comprising Bangladesh and West Bengal, India, has not remained untouched by this unsettling phenomenon. Online platforms, once hailed as vehicles for progress and connection, are now grappling with the disturbing spread of violence-inciting language, leading to communal discord, destruction, and loss of life.

In this digital age, where the boundaries between the virtual and the real world blur, it becomes imperative to address the multifaceted manifestations of communal violence, particularly in regions like Bengal. The Violence Inciting Text Detection (VITD) shared task emerges as a beacon of hope and a clarion call for the natural language processing (NLP) community to confront this pressing issue head-on.

The VITD shared task centres on the precise categorization and discernment of violence-inciting text within social media comments, echoing the broader challenge of understanding the dark underbelly of online discourse. The violence we seek to detect and categorize transcends mere words on a screen; it has the potential to manifest as explicit threats, divisive propaganda, and derogatory language that can irreparably harm individuals and communities.

This paper discusses our team's system, built as a part of the BLP Shared Task 1: Violence Inciting Text Detection (VITD) (Saha et al., 2023b,a). In this work, we experimented with several pre-trained sequence classification models with the provided data only where we contributed to data augmentation, sampling stratgies, fine-tuning and hyper-parameter tuning to optimize the performance of these models.[1] Our system was ranked 20th among the 27 teams that participated in the task.

2 Related work

The proliferation of hate-speech, verbal threats, aggression, cyberbullying, trolling, abuse, offensive and misogny content are experiencing rapid growth on social media. A considerable number of researchers have been actively involved in investigating the automated detection of offensive and hate speech content as well as many shared tasks were organising (Waseem and Hovy, 2016; Kumar et al., 2018; Mandl et al., 2019; Zampieri et al., 2020; Davani et al., 2023). However, there is considerably less research on violence detection specifically. A few works are as follows: Cano Basave et al. (2013) present the Violence Detection Model (VDM), a probabilistic framework for identifying violent content and extracting violence-related topics from social media without requiring labeled data. VDM uses word prior knowledge derived from relative entropy to cap-

[1]Our code is available at https://github.com/souro/classification_tasks_bangla

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 220–224

ture word violence indicators, outperforming information gain methods in topic identification and violence classification. Chang et al. (2018) address the detection of aggression and loss in social media, particularly among gang-involved youth. Their system incorporates contextual representations and domain-specific resources, improving the Convolutional Neural Network's performance for detecting aggression and loss. Jahan et al. (2022) introduce BanglaHateBERT, a retrained BERT model for abusive language detection in Bangla. It outperforms generic pre-trained models on various datasets and includes a 15K Bangla hate speech dataset for research use. Zandam et al. (2023) explore the expression of threatening themes in the Hausa language on social media, developing a classification system using machine learning algorithms. XGBoost achieves the highest accuracy of 72% in classifying threatening content. Abercrombie et al. (2023) conduct a systematic review of resources for automated identification of online Gender-Based Violence (GBV), highlighting limitations in existing datasets, such as a lack of theoretical grounding and stakeholder input. The study recommends future resources grounded in sociological expertise and involving GBV experts and those with lived GBV experience.

3 Dataset

The VITD Shared Task 1 dataset (Saha et al., 2023b) was provided by the task organisers. Individual samples in the dataset are labeled as Direct Violence, Indirect Violence, and Non-Violence, which are represented numerically by 2, 1 and 0 respectively (see Saha et al., 2023b for further details).

The dataset is divided into training, development and test sets, consisting of 2,700, 1,330 and 2,016 samples respectively.[2]

4 Experiments

This section discusses an extensive account of the system we designed for the VITD and Sentiment Analysis of Bangla Social Media Posts tasks. Our strategy encompasses several stages, such as data preprocessing, model choice, hyperparameter adjustment, and advanced methods, all aimed at attaining commendable outcomes.

[2] https://github.com/blp-workshop/blp_task1/tree/main/dataset

4.1 Data Preprocessing

At the outset, a thorough data preprocessing and cleaning phase was performed for our system, which established a robust basis for subsequent operations. We harnessed the tools offered by the Bangla Natural Language Processing (BNLP) toolkit (Sarker, 2021). In addition to basic text processing, we implemented crucial transformations like setting `fix_unicode=True`, `unicode_norm=True`, and `unicode_norm_form="NFKC"`. These steps ensured consistent and standardized text representations, enhancing the quality of our dataset.

4.2 Model Selection

Our system employed a range of pre-trained sequence classification models to tackle the classification tasks effectively. Notable models we experimented with include `XLMRobertaForSequenceClassification`, `BertForSequenceClassification`, and their variants. Specifically, we explored the following models: XLM-RoBERTa (base and large versions) (Conneau et al., 2019), BanglaBERT '' (Bhattacharjee et al., 2022), Bangla BERT Base (Sarker, 2020) and BERT-base-multilingual-cased (Devlin et al., 2018). [3] After thorough evaluation, we found the XLM-RoBERTa-base model to perform best on this task.

4.3 Hyperparameter Tuning

Based on hyperparameter search on the development data, we chose the following hyperparameter settings: batch size of 5,learning rate (lr) 1e-5, using the AdamW optimizer (Loshchilov and Hutter, 2019), training for 15 epochs, setting gradient clipping to 1.0, a weight decay of 0.01, and a dropout rate of 0.1.

4.4 Sampling Strategies

Class imbalance arises when certain classes have notably fewer samples than others, potentially leading to bias in favour of the majority class within the model. This is the case in tasks such as violence detection, where violent texts are in the minority. To

[3] We use the models from HuggingFace: https://huggingface.co/xlm-roberta-base, https://huggingface.co/xlm-roberta-large, https://huggingface.co/csebuetnlp/banglabert, https://huggingface.co/sagorsarker/bangla-bert-base, https://huggingface.co/bert-base-multilingual-cased.

address class imbalance issues, we experimented with both oversampling and undersampling techniques. Although the outcomes were promising, our best-performing model ultimately adopted an alternative approach – focal loss.

Focal Loss (Lin et al., 2017) was incorporated as a specialized loss function to combat the class imbalance issues present in our classification tasks. Focal Loss (Lin et al., 2017) works by significantly reducing the loss for correctly classified examples with high confidence, effectively handling easy instances. Simultaneously, it provides a smaller reduction in loss for difficult-to-classify or misclassified examples, ensuring that the model concentrates on learning from problematic cases. The key idea behind Focal Loss is to give more attention to hard-to-classify examples while reducing the impact of well-classified examples. This is achieved through two essential parameters: `alpha` and `gamma`.

Alpha Parameter (`alpha`): In our system, we set `alpha` to 1. This value signifies that we assigned equal weight to all classes. By doing so, we aimed to ensure that our model did not exhibit bias towards any specific class. However, adjusting `alpha` allows for a flexible weighting scheme, where higher values give more importance to minority classes.

Gamma Parameter (`gamma`): We chose a `gamma` value of 2. This parameter regulates the rate at which the loss decreases as the predicted probability for the correct class increases. A higher `gamma` value, as in our case, slows down the loss reduction for well-classified examples. This design decision helped our model focus on challenging or misclassified instances, potentially leading to improved overall performance.

In summary, Focal Loss played a crucial role in enhancing the performance of our system, especially in scenarios with imbalanced class distributions. Our choice of `alpha` and `gamma` parameters aligns with standard practices for effectively leveraging Focal Loss to tackle classification challenges.

4.5 Data Augmentation

The diversity and robustness of our model was enhanced through data augmentation. A data augmentation strategy with a probability of 0.5 was introduced on the original data (Saha et al., 2023b).

Model	macro-F1
BanglaBERT Baseline	0.7879
XLM-RoBERTa base Baseline	0.7292
BERT multilingual base (cased) Baseline	0.6819
BLP Shared Task 1 winning system	76.044
Our system	69.009

Table 1: UFAL-ULD team and baseline systems results

The techniques employed included synonym replacement, insertion, deletion, swap, and shuffling (cf. Mukherjee and Dusek, 2023). Through a collective application of these techniques, a diverse set of augmented data was generated that proved vital to the performance of our best-reported model.

In summary, a systematic approach for data preprocessing, model selection, hyperparameter tuning, class imbalance handling, the integration of advanced loss functions, and data augmentation was employed to achieve competitive results for the VITD task.

5 Results

The macro-F1 metric has been used for evaluation measure in the BLP Shared Task 1 (Saha et al., 2023a), with comparisons made against the ground truth labels. This metric signifies the comprehensive effectiveness of our system in accurately categorizing text that incites violence into the specified classifications: Direct Violence, Passive Violence, and Non-Violence. The macro F1 score is a resilient measurement that considers precision and recall across all categories, making it particularly suitable for tasks with imbalanced class distributions. Our system achieved a macro F1 score of 69.01 on the test set (see Table 1), outperforming baselines. Our system was ranked 20th among the 27 teams that participated in the task.[4]

6 Conclusion

In this shared task on Violence Inciting Text Detection (VITD), we have presented our system's approach and results, emphasizing the significance of addressing the challenging problem of identifying and categorizing violence-inciting text in the Bangla language. Our system, equipped with a comprehensive set of natural language processing techniques, achieved a competitive macro F1 score of 69.009 on the test set. Our system was ranked 20th among the 27 teams that participated in the task.

[4] https://github.com/blp-workshop/blp_task1

We remain committed to further refining our system and exploring innovative approaches to contribute to the ongoing efforts in violence detection and prevention.

Limitations

While our system performed well in the VITD shared task, it is essential to acknowledge certain limitations:

Data Availability: Our system's performance heavily relies on the quality and quantity of training data. The availability of more extensive and diverse annotated datasets in Bangla could further enhance our system's capabilities.

Ethical Considerations: As with any content analysis task, there is the potential for bias and sensitivity in handling violent or offensive text. Ensuring ethical considerations and responsible AI practices are crucial in the development and deployment of such systems.

Ethics Statement

In developing our system for the Violence Inciting Text Detection task, we adhered to ethical principles and guidelines for responsible AI. We are committed to the following ethical considerations:

Data Privacy: We respect data privacy and ensure that any data used in our experiments are anonymized and do not contain personally identifiable information.

Bias Mitigation: We took measures to mitigate bias in our system, both in terms of model performance and the potential impact of our work on society. We recognize the importance of fairness and impartiality in automated content analysis.

Transparency: We are committed to transparency in our research and have provided a detailed system description, including preprocessing steps, model selection, and evaluation metrics.

Accountability: We are open to feedback and accountability for our work. We encourage responsible use and scrutiny of AI technologies, and we remain responsive to concerns or issues related to our system's functionality.

By adhering to these principles, we aim to contribute to the responsible development and deployment of AI systems for content analysis, with a focus on promoting online safety and mitigating harm.

Acknowledgements

This research was supported by the European Research Council (Grant agreement No. 101039303 NG-NLG) and by Charles University projects GAUK 392221 and SVV 260575. We acknowledge of the use of resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).
Atul Kr. Ojha would like to acknowledge the support of the Science Foundation Ireland (SFI) as part of Grant Number SFI/12/RC/2289_P2 Insight_2, Insight SFI Centre for Data Analytics.

References

Gavin Abercrombie, Aiqi Jiang, Poppy Gerrard-abbott, Ioannis Konstas, and Verena Rieser. 2023. Resources for automated identification of online gender-based violence: A systematic review. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 170–186, Toronto, Canada. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Amparo Elizabeth Cano Basave, Yulan He, Kang Liu, and Jun Zhao. 2013. A weakly supervised Bayesian model for violence detection in social media. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 109–117, Nagoya, Japan. Asian Federation of Natural Language Processing.

Serina Chang, Ruiqi Zhong, Ethan Adams, Fei-Tzin Lee, Siddharth Varia, Desmond Patton, William Frey, Chris Kedzie, and Kathy McKeown. 2018. Detecting gang-involved escalation on social media using context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 46–56, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Aida Mostafazadeh Davani, Mohammad Atari, Brendan Kennedy, and Morteza Dehghani. 2023. Hate speech classifiers learn normative social stereotypes. *Transactions of the Association for Computational Linguistics*, 11:300–319.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Md Saroar Jahan, Mainul Haque, Nabil Arhab, and Mourad Oussalah. 2022. BanglaHateBERT: BERT for abusive language detection in Bengali. In *Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis*, pages 8–15, Marseille, France. European Language Resources Association.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. *COLING 2018*, page 1.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandalia, and Aditya Patel. 2019. Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in indo-european languages. In *FIRE '19: Forum for Information Retrieval Evaluation, Kolkata, India, December, 2019*, pages 14–17. ACM.

Sourabrata Mukherjee and Ondrej Dusek. 2023. Leveraging Low-resource Parallel Data for Text Style Transfer. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 388–395, Prague, Czechia. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Nabeel Mohammed, Sudipta Kar, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Sagor Sarker. 2021. BNLP: natural language processing toolkit for bengali language. *CoRR*, abs/2102.00405.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 88–93. The Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

Abubakar Yakubu Zandam, Fatima Adam Muhammad, and Isa Inuwa-Dutse. 2023. Online threats detection in hausa language. In *Proceedings of the 4th Workshop on African Natural Language Processing, AfricaNLP@ICLR 2023, Kigali, Rwanda, May 1, 2023*.

Semantics Squad at BLP-2023 Task 1: Violence Inciting Bangla Text Detection with Fine-Tuned Transformer-Based Models

Krishno Dey[1], **Prerona Tarannum**[2], **Md. Arid Hasan**[1], **Francis Palma**[1]
[1]SE+AI Research Lab, University of New Brunswick, Fredericton, Canada
[2]Daffodil International University, Dhaka, Bangladesh
krishno.dey@unb.ca, prerona15-14134@diu.edu.bd,
arid.hasan@unb.ca, francis.palma@unb.ca

Abstract

This study investigates the application of Transformer-based models for violence threat identification. We participated in the BLP-2023 Shared Task 1 and in our initial submission, BanglaBERT large achieved 5th position on the leader-board with a macro F1 score of 0.7441, approaching the highest baseline of 0.7879 established for this task. In contrast, the top-performing system on the leaderboard achieved an F1 score of 0.7604. Subsequent experiments involving m-BERT, XLM-RoBERTa base, XLM-RoBERTa large, BanglishBERT, BanglaBERT, and BanglaBERT large models revealed that BanglaBERT achieved an F1 score of 0.7441, which closely approximated the baseline. Remarkably, m-BERT and XLM-RoBERTa base also approximated the baseline with macro F1 scores of 0.6584 and 0.6968, respectively. A notable finding from our study is the under-performance by larger models for the shared task dataset, which requires further investigation. Our findings underscore the potential of transformer-based models in identifying violence threats, offering valuable insights to enhance safety measures on online platforms.

1 Introduction

The global use of social media platforms has significantly increased due to the massive use of the internet and over the past few decades, internet use has rapidly expanded. People are now more able to share information and their opinions online because of easier access to the internet. Social media use has gained a curve that has been steadily increasing and shows no signs of stopping[1]. The study of (Tarannum et al., 2023) includes how some actors use social media platforms to spread false information that covers offensive language, cyber-bullying, cyber-aggression, rumors, and hate speech. As a result, spreading hate speech is now easy and common. This change also gives social media significant influence over our society. On these social media platforms, people are free to express different points of view on religion, politics, education, and other issues. The use of Bangla on social media platforms is growing because it is the seventh most spoken language[2], and internet usage is rising. While there has been much research on detecting hate speech in English, there is a knowledge gap in identifying hostile content in low-resource languages like Bangla. This type of information generated interest in identifying and flagging it due to its potentially misleading or dangerous nature, which might help stop its future spread. According to research by (Yin et al., 2009), machine learning algorithms are more accurate than keyword searches and textual content analysis for social media data. However, many of the proposed machine learning techniques are, in fact, topic-specific.

Even though there are some works on sentiment analysis for the Bangla language, there has not been much research done recently to identify abusive Bangla text on social networking sites. Due to the constantly changing nature of social media and the wide range of language used, identifying abusive text is difficult. Researchers tried to develop various methods to identify abusive or objectionable text (Nobata et al., 2016) to prevent abuse on online platforms.

In this study, we participated in BLP Shared Task 1: Violence Inciting Text Detection (VITD) (Saha et al., 2023b) and the dataset offered in the shared task has two columns (label, text), with the label in three categories (*Direct Violence*, *Passive Violence*, and *Non-Violence*) (Saha et al., 2023a). We conducted a series of Transformer-based experiments on the dataset provided by the organizers. In our blind submission with BanglaBERT large, we achieved a macro F1 score of 0.7441, securing the 5th position on the official leaderboard.

[1]https://www.pewresearch.org/internet/fact-sheet/social-media/

[2]https://www.ethnologue.com/insights/ethnologue200/

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 225–229

The highest baseline for the task was a macro F1 score of 0.7879, while the best-performing system, developed by DeepBlueAI, achieved a macro F1 score of 0.7604. Subsequently, we re-ran the experiments employing m-BERT (Devlin et al., 2018), XLM-RoBERTa base (Conneau et al., 2019), XLM-RoBERTa large, BanglishBERT (Bhattacharjee et al., 2022), BanglaBERT (Bhattacharjee et al., 2022), and BanglaBERT large (Bhattacharjee et al., 2022) models. BanglaBERT, m-BERT, and XLM-RoBERTa base came quite close to the respective official baseline performance in these evaluations. Other models, including XLM-RoBERTa large, BanglishBERT, and BanglaBERT large, demonstrated noteworthy and commendable performance.

The structure of this paper is as follows: Section 2 summarizes the relevant works for this study. Section 3 reports the methodology. A detailed discussion of the results of our study is provided in Section 4. Finally, we state limitations and future work in Section 5.

2 Related Works

Social media has integrated itself into everyone's daily lives. It makes it possible to communicate quickly, share information easily, and receive opinions across geographical boundaries. However, this manifestation of freedom has also contributed to the development of harsh language and hate speech on social media platforms. Unfortunately, the detection of hate speech in Bangla social media has received very little attention. The main issue is the unavailability of sufficient data. In addition, the terminology used in hate speech is extremely diverse. Social media language frequently deviates significantly from that of traditional print media. Numerous linguistic characteristics are present in it. Therefore, it is difficult to recognize hate speech automatically.

In heterogeneous language-speaking countries like Ethiopia, data in Amharic was gathered and annotated to detect hate speech by (Mossie and Wang, 2020) and proposed a method for automatic detection of hate speech directed towards vulnerable minority groups on social media. The authors reported that RNN-GRU exhibits the best performance with an accuracy of 92.56% and an AUC of 97.85%. The accuracy of all algorithms improved using word embeddings like Word2Vec. Early research by (Kiilu et al., 2018) created a method for identifying and categorizing hate speech using content from self-identified hate communities on X (formally known as Twitter) and suggested that the Naive Bayes classifier greatly outperformed the existing approaches with 67.47% accuracy. Another study with GPT-3 to identify sexist and racist text passages, (Chiu et al., 2021) discovered that the model accuracy could reach as high as 85% with few-shot learning. (Romim et al., 2021a) created the HS-BAN dataset on hate speech with their benchmark system, and the best outcome was obtained by combining Bi-LSTM with FT(SG) or Bi-LSTM+FT(SG), which achieved an F1 score of 86.85%. (Ishmam and Sharmin, 2019) built a dataset with 5,126 Bangla comments from social media and got an accuracy of 70.1% using GRU-based models, which gave 18% higher accuracy than ML algorithms.

A recent study by (Alam et al., 2020) using several publicly accessible datasets for the experiments by fine-tuning multilingual transformer models for Bangla text classification tasks in several areas to improve accuracy upon the prior results between 5%-29% across different tasks. The dataset for this study was obtained from X (previously known as Twitter). According to (Das et al., 2022), the XLM-Roberta model has the highest accuracy on their developed annotated dataset which consists of 10K Bangla posts where 5K is actual and 5K is Romanized Bangla tweets. By preparing only a multi-modal hate speech dataset, after experiments (Karim et al., 2022) reported F1 scores of 78% and 82%, respectively, using Conv-LSTM and XLM-RoBERTa models, which scored best for texts. ResNet-152 and DenseNet-161 models produced F1 scores of 78% and 79% for memes, respectively. Concerning multi-modal fusion, XLM-RoBERTa + DenseNet-161 demonstrated the best performance, producing an F1 score of 83%. (Islam et al., 2021) took data from some controversial pages of social media and after evaluation, the maximum accuracy of 88% was achieved by SVM using the entire dataset. On the dataset of (Romim et al., 2021b), the authors ran baseline experiments, applied several deep learning models, and extensively trained Word2Vec, FastTest, and BengFastText models on Bangla words to facilitate future research opportunities, and the experiment showed that SVM had the best outcome with 87.5% accuracy.

3 Experimental Methodology

This section describes our experimental methodology. We start with a brief overview of the dataset, then talk about our pre-processing steps, and present in-depth explanations of the models used in this study.

3.1 Dataset

We utilized the dataset offered by the BLP-2023 Shared Task 1. The dataset consists of YouTube comments about the top nine violent incidents in Bangladesh and West Bengal over the last decade between 2013-2023. The dataset includes Bangla-language content with comments that can be up to 600 words long. The dataset contains three data classes: *Direct Violence*, *Passive Violence*, and *Non-Violence*.

Split	Samples	DV	PV	NV
Train	2700	15%	34%	51%
Dev	1330	15%	31%	54%
Test	2016	10%	36%	54%

Table 1: Overview of the Data and Splitting Procedure. NV: Non-Violence, PV: Passive Violence, DV: Direct Violence.

- **Direct Violence:** Comments directly promoting or inciting violence.
- **Passive Violence:** Comments indirectly endorsing or facilitating violent actions.
- **Non-Violence:** Comments that do not relate to violence.

The offered dataset in the shared task can help identify and classify threats associated with violence, potentially leading to further incitement of violent acts. Table 1 shows the dataset distribution.

3.2 Pre-Processing

Several pre-processing steps were carried out in preparing the BLP-2023 shared task 1 dataset for analysis and classification. The text data underwent an extensive cleaning phase, during which special characters, URLs, and punctuation were eliminated. Tokenization was then used to separate the text into individual words or tokens. Then we eliminated all of the stop words, which are generally low-content words that are used frequently in a language. When classifying documents, the elimination of stop words enables the classification algorithm to concentrate on the keywords. These pre-processing steps further enhanced the quality of the dataset for subsequent analysis and classification tasks.

L	Acc	P	R	F1	F1-m
Multilingual BERT(m-BERT)					
NV		0.73	0.85	0.79	
PV	0.7138	0.77	0.52	0.62	0.6584
DV		0.51	0.63	0.57	
XLM-RoBERTa base					
NV		0.77	0.84	0.80	
PV	0.7376	0.76	0.59	0.66	0.6968
DV		0.55	0.72	0.63	
XLM-RoBERTa large					
NV		0.80	0.87	0.84	
PV	0.7679	0.80	0.61	0.69	0.7246
DV		0.55	0.78	0.65	
BanglishBERT					
NV		0.76	0.88	0.81	
PV	0.7321	0.81	0.50	0.62	0.7232
DV		0.52	0.78	0.62	
BanglaBERT					
NV		0.82	0.89	0.85	
PV	0.7867	0.82	0.64	0.71	**0.7441**
DV		0.58	0.79	0.67	
BanglaBERT large					
NV		0.81	0.88	0.84	
PV	0.7773	0.82	0.61	0.70	0.7344
DV		0.56	0.80	0.66	

Table 2: Comprehensive Breakdown of the Classification Results. Bold numbers indicate the best F1 score. NV: Non-Violence, PV: Passive Violence, DV: Direct Violence, L: Label, Acc: Accuracy, P: Precision, R: Recall, F1: F1 Score, F1-m: F1-macro.

.

3.3 Models

We employed several transformer-based models, including m-BERT, XLM-RoBERTa base, XLM-RoBERTa large, BanglishBERT, BanglaBERT, and BanglaBERT large. Each model was trained for five epochs, a duration sufficient for convergence on the test data. In order to enhance the model's

performance, a batch size of 32 was utilized to accelerate the training procedure except for XLM-RoBERTa large, for which we employed a batch size of 16 due to resource limitations, wherein gradient accumulation was calculated following every 32 data samples. The selection of a learning rate of $2e^{-5}$ was based on the principle that this rate facilitates more efficient learning of parameter estimates by the algorithm.

4 Result Analysis & Discussion

In our study, we evaluated a wide range of models, such as m-BERT, XLM-RoBERTa base, XLM-RoBERTa large, BanglishBERT, BanglaBERT, and BanglaBERT large, to determine how well they perform identifying and categorizing threats related to violence in the BLP-2023 dataset. One of the most critical findings from our analysis was the unexpected performance of the smaller model, BanglaBERT, which outperformed larger models like m-BERT, XLM-RoBERTa large, BanglishBERT, and BanglaBERT large. This unexpected outcome highlights the importance of model architecture and the flexibility with which it can be adapted to the specifics of the dataset. Despite its smaller size, BanglaBERT outperformed other models, suggesting its ability to capture the subtleties of language and context related to violence within the Bangla dataset. This superior performance can be attributed to its training on a Bangla dataset, enabling it to excel in this specific linguistic and contextual domain. This finding emphasizes the significance of pre-training data and architecture, in addition to size, when choosing models for particular NLP tasks.

Table 2 illustrates that BanglaBERT achieved the highest accuracy, reaching 0.7876, surpassing all other models in our evaluation. BanglaBERT outperformed other models with precision and recall of 0.7996 and 0.7808, respectively. The official scoring metric for the BLP2023 shared task 1 was the macro F1 score. The baseline macro F1 scores for the shared task set by the organizers were 0.7879 for BanglaBERT, 0.7292 for XLM-RoBERTa base, and 0.6819 for BERT multilingual. BanglaBERT achieved a macro F1 score of 0.7441 in our study, quite close to the baseline. BERT multilingual base (cased) achieved a macro F1 score of 0.7068, surpassing the baseline, while the XLM-RoBERTa base achieved an F1 score of 0.7347, also surpassing the baseline. Additionally, other models, namely XLM-RoBERTa large, achieved an F1 score of 0.7246, BanglishBERT achieved an F1 score of 0.7232, and BanglaBERT large achieved an F1 score of 0.7344. The macro F1 score of BanglaBERT in our study closely matches the baseline, with a little difference, and it outperforms other models in our study for this specific dataset of shared task 1. Table 3 shows the performance on the official leaderboard of our works compared to the baselines and other works.

System	F1 Score	Rank
Our Work	**0.7441**	5^{th}
DeepBlueAI	0.7604	1^{st}
Baseline(BanglaBERT)	0.7879	–
Baseline(XLM-RoBERTa)	0.7292	–
Baseline(mBERT)	0.6819	–

Table 3: Official results on the test set and overall ranking of Task 1: Violence Inciting Text Detection (VITD). **Bold** indicates our systems.

Overall, BanglaBERT stands out as a dependable and competitive solution to the challenging problem of identifying violence threats in Bangla. Its capacity to closely match the baseline macro F1 score and its strong precision and recall metrics highlight its potential to strengthen safety measures in the online environment, where the detection of violent threats is of utmost importance. The effectiveness of transformer-based models, particularly BanglaBERT, in identifying violent threats is reaffirmed by this comprehensive viewpoint, which also provides invaluable insights for enhancing online security measures.

5 Conclusion and Future Work

In this study, we performed a comparative study on several transformer-based models to detect violent text. We used the dataset offered by shared task 1 of the BLP workshop for this study. We used such as m-BERT, XLM-RoBERTa base, XLM-RoBERTa large, BanglishBERT, BanglaBERT, and BanglaBERT large to compare their result. The result shows that BanglaBERT outperformed other models in terms of performance measures. Despite being larger models m-BERT, XLM-RoBERTa large, BanglaBERT large could not outperform BanglaBERT. One of the limitations of our work is that we were not able to reveal the specific reason why our large models are not performing as they

are supposed to for this task.

To extend this study, we plan to employ transfer learning, efficient model designs, or model compression to improve the performance of large models such as m-BERT, XLM-RoBERTa, and BanglaBERT large. In the context of hate speech detection, we will investigate the ideal hyperparameters for transformer-based models, including learning rate schedules, model size, and optimization strategies. The development of more accurate, fair, and reliable hate speech detection algorithms may emerge from future research in these areas, thus, resolving the limitations of this study and enhancing the field of NLP.

References

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Bangla text classification using transformers. *arXiv preprint arXiv:2011.04446*.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Mubasshir, Md. Saiful Islam, Wasi Ahmad Uddin, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Lagnuage model pretraining and benchmarks for low-resource language understanding evaluation in bangla. In *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL 2022*.

Ke-Li Chiu, Annie Collins, and Rohan Alexander. 2021. Detecting hate speech with gpt-3. *arXiv preprint arXiv:2103.12407*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Mithun Das, Somnath Banerjee, Punyajoy Saha, and Animesh Mukherjee. 2022. Hate speech and offensive language detection in bengali. *arXiv preprint arXiv:2210.03479*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Alvi Md Ishmam and Sadia Sharmin. 2019. Hateful speech detection in public facebook pages for the bengali language. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pages 555–560. IEEE.

Tanvirul Islam, Nadim Ahmed, and Subhenur Latif. 2021. An evolutionary approach to comparative analysis of detecting bangla abusive text. *Bulletin of Electrical Engineering and Informatics*, 10(4):2163–2169.

Md Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Md Shajalal, and Bharathi Raja Chakravarthi. 2022. Multimodal hate speech detection from bengali memes and texts. In *International Conference on Speech and Language Technologies for Low-resource Languages*, pages 293–308. Springer.

Kelvin Kiema Kiilu, George Okeyo, Richard Rimiru, and Kennedy Ogada. 2018. Using naïve bayes algorithm in detection of hate tweets. *International Journal of Scientific and Research Publications*, 8(3):99–107.

Zewdie Mossie and Jenq-Haur Wang. 2020. Vulnerable community identification using hate speech detection on social media. *Information Processing & Management*, 57(3):102087.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2021a. Hs-ban: A benchmark dataset of social media comments for hate speech detection in bangla. *arXiv preprint arXiv:2112.01902*.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam. 2021b. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*, pages 457–468. Springer.

Sourav Saha, Jahedul Alam Junaed, Arnab Sen Sharma Api, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023a. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Nabeel Mohammed, Sudipta Kar, and Mohammad Ruhul Amin. 2023b. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Prerona Tarannum, Md Arid Hasan, Firoj Alam, and Sheak Rashed Haider Noori. 2023. Z-index at checkthat! 2023: Unimodal and multimodal checkworthiness classification.

Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian D Davison, April Kontostathis, Lynne Edwards, et al. 2009. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, 2(0):1–7.

LowResourceNLU at BLP-2023 Task 1 & 2: Enhancing Sentiment Classification and Violence Incitement Detection in Bangla Through Aggregated Language Models

Hariram Veeramani
Department of Electrical and Computer Engineering, UCLA, USA
hariram@ucla.edu

Surendrabikram Thapa
Department of Computer Science, Virginia Tech, Blacksburg, USA
sbt@vt.edu

Usman Naseem
College of Science and Engineering, James Cook University, Australia
usman.naseem@jcu.edu.au

Abstract

Violence incitement detection and sentiment analysis hold significant importance in the field of natural language processing. However, in the case of the Bangla language, there are unique challenges due to its low-resource nature. In this paper, we address these challenges by presenting an innovative approach that leverages aggregated BERT models for two tasks at the BLP workshop in EMNLP 2023, specifically tailored for Bangla. Task 1 focuses on violence-inciting text detection, while task 2 centers on sentiment analysis. Our approach combines fine-tuning with textual entailment (utilizing BanglaBERT), Masked Language Model (MLM) training (making use of BanglaBERT), and the use of standalone Multilingual BERT. This comprehensive framework significantly enhances the accuracy of sentiment classification and violence incitement detection in Bangla text. Our method achieved the 11th rank in task 1 with an F1-score of 73.47 and the 4th rank in task 2 with an F1-score of 71.73. This paper provides a detailed system description along with an analysis of the impact of each component of our framework.

1 Introduction

Natural Language Processing (NLP) has witnessed remarkable advancements in recent years, transforming the way we interact with and understand textual data (Khurana et al., 2023). From chatbots and machine translation to information retrieval and sentiment analysis, NLP has become an indispensable tool for extracting meaning from the vast sea of human-generated text (Sun et al., 2022). Among the diverse array of NLP tasks, sentiment analysis, and violence incitement detection stand out as pivotal areas with far-reaching implications for societal well-being and communication (Khalafat et al., 2021; Castorena et al., 2021).

THIS PAPER CONTAINS EXAMPLES OF VIOLENT TEXT.

Sentiment analysis, also known as opinion mining, is a fundamental NLP task focused on identifying emotional tones and polarities within the text (Cui et al., 2023). It plays a crucial role in various applications, including gauging public opinion, analyzing consumer feedback, monitoring social media, and managing brand perception. By providing insights into sentiment, it empowers informed decision-making, personalized communication, and more effective response strategies (Wankhade et al., 2022). Similarly, in an increasingly digital world, the spread of harmful content, including violence-inciting text, poses significant challenges (Parihar et al., 2021). Violence incitement detection is a critical aspect of content moderation, ensuring online platforms remain safe and free from content that promotes harm, hatred, or illegal activities. Early identification of such content is vital in mitigating potential harm, preserving online discourse, and upholding ethical standards in digital communication.

While the significance of sentiment analysis and violence incitement detection is widely recognized, applying these techniques to low-resource languages presents unique hurdles (Sen et al., 2022). The Bangla language, with its rich linguistic diversity, is a prime example. Despite its extensive speaker base, Bangla remains underrepresented in NLP research, often lacking the comprehensive language resources available for widely spoken languages (Kowsher et al., 2022). This scarcity of resources hinders the development of effective sentiment analysis and violence incitement detection tools for Bangla.

We address the aforementioned problems by presenting a novel approach based on the aggregation of BERT-based models. In this paper, we provide detailed descriptions of our systems for two tasks at the BLP workshop. Our contributions include:

- Our method encompasses three unique

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 230–235

Text	Translation	Label
ঢাকা কলেজে আগুন লাগিয়ে এই কুলাঙ্গার ছাত্রদের পুরিয়ে মারা উচিৎ,, এরাই এখন গলার কাটা	These Kulanga students should be killed by setting fire to Dhaka College, they are now cut throat	Direct Violence
শয়তান মেরে হাসবে না তো কাঁদবে!!	The devil will not laugh but cry!!	Passive Violence
যে মারা গেল তার ক্ষতিপূরনের ব্যবস্থা করে দেওয়া হোক।	Compensation should be paid to the person who died.	Non-Violence

Table 1: Examples of text used in task 1 (Violence Inciting Text Detection)

approaches: simultaneous fine-tuning of BanglaBERT for MLM and classification tasks, straightforward utilization of Multilingual BERT (mBERT), and a multi-head training strategy addressing two distinct topics (entailment and classification), collectively enhancing performance in natural language processing tasks.

- We conduct ablation studies to analyze the individual effects of each component in our proposed methodology, shedding light on their respective contributions.

2 Task Descriptions

Task 1: This task focuses on violence incitement text classification (Saha et al., 2023b). The primary objective is to identify and classify Bangla text comments that contain threats associated with violence, which have the potential to incite further acts of violence. Participants were required to categorize the comments into three distinct categories: "Direct Violence", "Passive Violence", and "Non-Violence".

Task 2: It addresses sentiment analysis, aiming to detect the sentiment expressed within a given Bangla text (Hasan et al., 2023a). It constitutes a multi-class classification challenge where participants are tasked with determining whether the sentiment in the Bangla text is "Positive", "Negative", or "Neutral".

3 Dataset

For task 1, participants are presented with a Bangla dataset comprising YouTube comments related to the top 9 violent incidents that have occurred in the Bengal region (comprising Bangladesh and West Bengal) over the past decade, with comments up to 600 words long (Saha et al., 2023a). The training set (2700 samples) comprises approximately 15% direct violence, 34% passive violence, and 51% non-violent instances. In the development set (1330 samples), a similar distribution is observed: 15% direct violence, 31% passive violence, and 54% non-violence. Table 1 shows examples of texts used in task 1.

For task 2, the given dataset combines two primary sources: the MUltiplatform BAngla SEntiment (MUBASE) (Hasan et al., 2023b) and SentNob (Islam et al., 2021) datasets. Thus, this dataset includes public comments on news and videos across 13 domains, and multiplatform content such as Tweets and Facebook posts, all manually annotated for sentiment polarity as shown in Table 2.

Text	Translation	Sentiment
বিবিসি মানের বাবাহীন সন্তান	BBC Standard Fatherless Child.	Negative
আমি আপনার সাথে সম্পূর্ণ একমত । ।	I totally agree with you. .	Neutral
শেখ রেহানা : এক সংগ্রামী জীবনের প্রতিচ্ছবি	Sheikh Rehana: A reflection of a struggling life.	Positive

Table 2: Examples of text used in task 2 (sentiment analysis)

4 System Description

In our methodology, we aggregate three BERT-based language models in order to tackle both classification tasks. The proposed methodology of our system is as shown in Figure 1.

Model A: In this model configuration, we incorporate two heads within the BanglaBERT-large (Bhattacharjee et al., 2022) framework. Our choice of incorporating two heads in the BanglaBERT-large architecture, one for Masked Language Modeling (MLM) and the other for classification, is driven by a thoughtful rationale and strong motivation. Firstly, this dual-headed approach enables us to retain the invaluable language understanding capabilities embedded in pre-

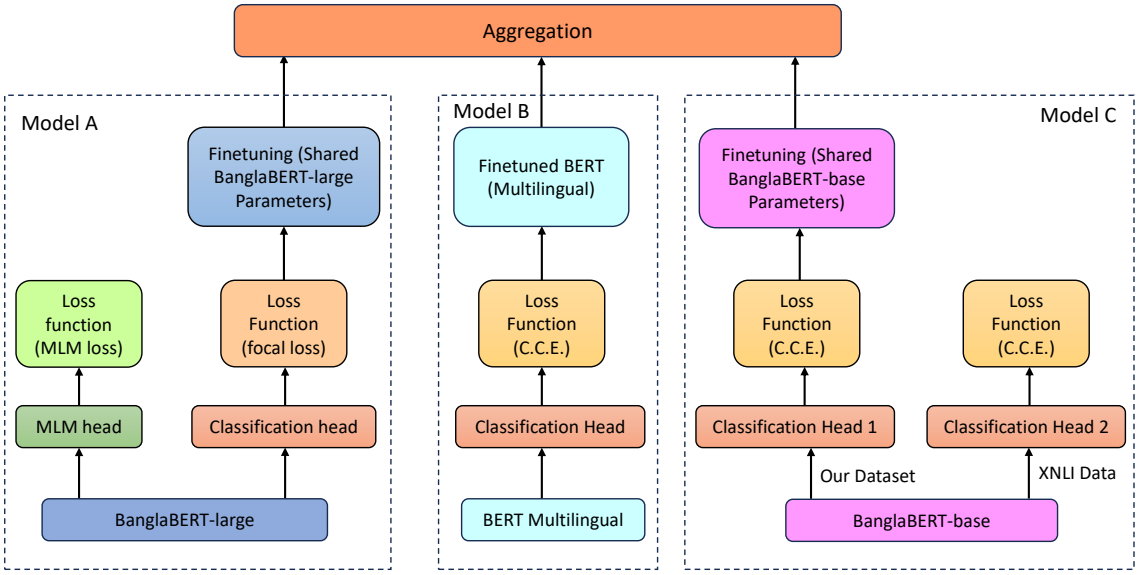


Figure 1: Proposed methodology for our system. Our methodology combines three BERT-based models capitalizing on the strength of each of them.

trained BanglaBERT. The MLM head, through cross-entropy loss (label vs. predicted), maintains and refines BanglaBERT's grasp of linguistic nuances, ensuring that it remains adept at capturing contextual word relationships. To enhance the MLM's effectiveness, we employ a balanced masking strategy. Specifically, within the MLM head, we utilize a 50% deterministic and 50% random masking approach. In random masking, we mask random words in the input text, while in deterministic masking, we append a mask token to our text. This dual approach enhances MLM's robustness in capturing contextual language information.

Secondly, the classification head, leveraging a specialized loss function like focal loss, empowers BanglaBERT to adapt swiftly and effectively to specific downstream tasks i.e. classification. This dynamic adaptability is crucial, as it enables BanglaBERT to excel in diverse applications, such as sentiment analysis, textual entailment, or any classification task at hand. Simultaneous training with shared parameters efficiently fuses the strength of both heads, resulting in a compact and versatile model that excels in various natural language processing tasks (Veeramani et al., 2023b,f,a), particularly classification.

Model B: It is Multilingual BERT (mBERT) (Devlin et al., 2019), a versatile architecture designed to handle multiple languages (Kassner et al., 2021; Xu et al., 2021; Veeramani et al., 2023c,e,d), including Bangla. Leveraging mBERT's rich multilingual knowledge, our methodology gains valuable linguistic insights, enhancing our understanding of Bangla text.

Model C: This model introduces a multi-head training strategy, simultaneously addressing two distinct yet interrelated tasks. The first head within BanglaBERT-base focuses on the XNLI dataset (Conneau et al., 2018), specifically targeting the task of textual entailment. This choice is motivated by the rationale of knowledge fusion, aiming to merge insights and linguistic patterns from both textual entailment and classification domains. XNLI has languages like Hindi, Urdu, and Swahili whose dialects and cultural nuances are similar to Bangla. We hypothesize that this helps the model to acquire better parameters. By sharing parameters across heads, the model seeks to develop a deeper and more comprehensive understanding of Bangla language nuances. The second head is dedicated to our data, which is centered around a classification problem. This dual-task approach not only boosts efficiency but also contributes to achieving our primary objective: solving the classification problem. The inclusion of the textual entailment task acts as an auxiliary training signal, facilitating the acquisition of versatile and adaptable language representations. This, in turn, aids in achieving superior performance in our core clas-

sification task, making Model C a powerful and efficient component of our methodology.

For all models A, B, and C, we made trials with focal loss and cross-entropy loss and used the loss function which gave the optimal performance. We also made trials with BanglaBERT-large and BanglaBERT-base and selected the most optimal framework as shown in Figure 1. All models have objective function as classification.

Aggregation: Our aggregation technique employs a multi-step process to effectively combine predictions from multiple models. Initially, we extract individual predictions from each model using the argmax function (Davani et al., 2022; Kanagasabai et al., 2023), selecting the class with the highest confidence score for each model. Subsequently, to consolidate these individual predictions, we apply another argmax operation, this time on the maximum logit values obtained from each model. This step ensures that we capture the most confident prediction across all models. If two labels have equal highest probabilities, we select the majority sample class.

5 Results

Performance on the task 1 and task 2 were evaluated on the basis of macro and micro F1-score respectively. Our team ranks 11th in task 1 with F1-score of 73.47. Similarly, our team ranks 4th in task 2 with macro F1-score of 71.72. Table 3 provides a comprehensive analysis of the impact of various models within our architecture, presenting macro-averaged F1-scores, precision, and recall for both tasks. In our analysis, we meticulously evaluate the impact of all models, focusing on a detailed assessment of Model A and Model C. We specifically delve into the effects of two crucial aspects: the integration of MLM (Masked Language Model) in Model A and the influence of joint pretraining with the XNLI dataset in Model C. Our Task 1 results demonstrate that Model A enhances the F1-score by a substantial margin, surpassing a 3.3-point improvement through the incorporation of MLM. Similarly, the joint pretraining with XNLI significantly enhances the performance of Model C by approximately 2.1 points. Model B alone gives an F1-score of 69.45. The combination of all components (Model A + B + C) exhibit superior performance as compared to use of single model alone.

In Task 2, which focuses on sentiment analysis,

Models	F1-score	Precision	Recall
Model A only	73.41	73.65	77.64
Model B only	69.45	70.28	70.87
Model C only	73.42	73.91	77.73
Model A w/o MLM	70.10	72.06	73.51
Model C w/o XNLI	71.34	73.17	76.00
Proposed (Model A + B + C)	**73.47**	**74.1**	**77.92**

Table 3: Results for Task 1 (Violence Incitement Text Detection). The F1-score, precision and recall are macro-averaged.

Table 4 provides a detailed performance analysis of various models. Model A without the inclusion of the Masked Language Model (MLM) component achieves an F1-micro score of 71.03, while Model C, operating without joint pretraining using the XNLI dataset, achieves an F1-micro score of 71.06. When evaluated independently, Model A attains an F1-micro score of 71.71, and Model C achieves a slightly higher F1-micro score of 71.72. Model B, on the other hand, was able to score an micro F1-score of 69.47. However, our proposed framework, which combines all three models (Model A, Model B, and Model C), outperforms these individual models. It achieves the highest F1-micro score of 71.73, highlighting the substantial improvement gained through the synergy of all models. Additionally, the framework excels in macro-averaged precision, recall, and F1-score, with values of 71.08, 71.73, and 71.36, respectively. These results underscore the effectiveness of our integrated approach in sentiment analysis, showcasing the value of combining multiple models for superior accuracy and performance.

Models	$F1_{mic}$	Pre_{mac}	Rec_{mac}	$F1_{mac}$
Model A only	71.71	70.43	71.72	70.67
Model B only	69.47	68.32	70.85	68.50
Model C only	71.72	71.06	71.70	71.34
Model A w/o MLM	71.03	68.95	71.00	69.00
Model C w/o XNLI	71.06	69.39	71.03	69.20
Proposed (Model A + B + C)	**71.73**	**71.08**	**71.73**	**71.36**

Table 4: Results for task 2 (sentiment analysis). The $F1_{mic}$ stands for micro-averaged F1-score. Similarly, Pre_{mac}, Rec_{mac}, and $F1_{mac}$ represents macro-averaged precision, recall and F1-score.

6 Conclusion

In conclusion, our methodology presents a detailed and novel approach to addressing the challenges of sentiment analysis and violence detection in Bangla text. By aggregating insights from three different language models, we achieve a high performance in both tasks. Through a detailed ablation analysis, we have analyzed the impact of each component, demonstrating the efficiency of our proposed approach. While our primary focus lies in sentiment analysis and violence detection, the consistently high performance across both tasks underscores the potential versatility of our method in various other text analysis applications in Bangla. In the future, more research can be done on bias mitigation, ensuring responsible and equitable deployment of our framework in a real-world context.

Limitations

We proposed a methodology primarily focused on sentiment analysis and violence incitement detection. In this process, we might be potentially overlooking other aspects of text analysis. The adaptability to different domains may require further fine-tuning, and the scalability of our approach could be challenged with very large datasets.

Ethics Statement

The framework may potentially generate biased interpretations, a critical aspect that requires thorough investigation before considering the deployment of our model in real-world applications. It is essential to note that we did not undertake a comprehensive bias analysis within the scope of this work, highlighting the need for future research to meticulously examine and mitigate any biases that might arise in practical implementations of our methodology.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Carlos M Castorena, Itzel M Abundez, Roberto Alejo, Everardo E Granda-Gutiérrez, Eréndira Rendón, and Octavio Villegas. 2021. Deep neural network for gender-based violence detection on twitter messages. *Mathematics*, 9(8):807.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jingfeng Cui, Zhaoxia Wang, Seng-Beng Ho, and Erik Cambria. 2023. Survey on sentiment analysis: evolution of research methods and topics. *Artificial Intelligence Review*, pages 1–42.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero-and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv preprint arXiv:2308.10783*.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rajaraman Kanagasabai, Saravanan Rajamanickam, Hariram Veeramani, Adam Westerski, and Kim Jung Jae. 2023. Semantists at ImageArg-2023: Exploring Cross-modal Contrastive and Ensemble Models for Multimodal Stance and Persuasiveness Classification. In *Proceedings of the 10th Workshop on Argument Mining*, Online and in Singapore. Association for Computational Linguistics.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual lama: Investigating knowledge in multilingual pretrained language models. *arXiv preprint arXiv:2102.00894*.

Monther Khalafat, S Alqatawna Jafar, Rizik Al-Sayyed, Mohammad Eshtay, and Thaeer Kobbaey. 2021. Violence detection over online social networks: An arabic sentiment analysis approach. *iJIM*, 15(14):91.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744.

Md Kowsher, Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, 10:91855–91870.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Sourav Saha, Jahedul Alam Junaed, Arnab Sen Sharma Api, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023a. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Nabeel Mohammed, Sudipta Kar, and Mohammad Ruhul Amin. 2023b. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Ovishake Sen, Mohtasim Fuad, Md Nazrul Islam, Jakaria Rabbi, Mehedi Masud, Md Kamrul Hasan, Md Abdul Awal, Awal Ahmed Fime, Md Tahmid Hasan Fuad, Delowar Sikder, et al. 2022. Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning-based methods. *IEEE Access*, 10:38999–39044.

Tian-Xiang Sun, Xiang-Yang Liu, Xi-Peng Qiu, and Xuan-Jing Huang. 2022. Paradigm shift in natural language processing. *Machine Intelligence Research*, 19(3):169–183.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023a. Automated Citation Function Classification and Context Extraction in Astrophysics: Leveraging Paraphrasing and Question Answering. In *Proceedings of the Second Workshop on Information Extraction from Scientific Publications*, Online. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023b. DialectNLU at NADI 2023 Shared Task: Transformer Based Multitask Approach Jointly Integrating Dialect and Machine Translation Tasks in Arabic. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023c. Enhancing ESG Impact Type Identification through Early Fusion and Multilingual Models. In *Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing (FinNLP)*. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023d. KnowTellConvince at ArAIEval Shared Task: Disinformation and Persuasion Detection in Arabic using Similar and Contrastive Representation Alignment. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023e. LowResContextQA at Qur'an QA 2023 Shared Task: Temporal and Sequential Representation Augmented Question Answering Span Detection in Arabic. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023f. Temporally Dynamic Session-Keyword Aware Sequential Recommendation system. In *2023 International Conference on Data Mining Workshops (ICDMW)*.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation. *arXiv preprint arXiv:2109.04588*.

Team Error Point at BLP-2023 Task 1: A Comprehensive Approach for Violence Inciting Text Detection using Deep Learning and Traditional Machine Learning Algorithm

Rajesh Kumar Das, Jannatul Maowa, Moshfiqur Rahman Ajmain, Kabid Yeiad, Mirajul Islam, Sharun Akter Khushbu
Department of Computer Science and Engineering
Daffodil International University, Dhaka, Bangladesh
{rajesh15-13032, jannatul15-14095, moshfiqur15-14090, yeiad15-14440, merajul15-9627, sharun.cse}@diu.edu.bd

Abstract

In the modern digital landscape, social media platforms have the dual role of fostering unprecedented connectivity and harboring a dark underbelly in the form of widespread violence-inciting content. Pioneering research in Bangla social media aims to provide a groundbreaking solution to this issue. This study thoroughly investigates violence-inciting text classification using a diverse range of machine learning and deep learning models, offering insights into content moderation and strategies for enhancing online safety. Situated at the intersection of technology and social responsibility, the aim is to empower platforms and communities to combat online violence. By providing insights into model selection and methodology, this work makes a significant contribution to the ongoing dialogue about the challenges posed by the darker aspects of the digital era. Our system scored 31.913 and ranked 26 among the participants.

1 Introduction

There is a great need for robust detection and classification algorithms in today's digital environment since violent incitement material is spreading so rapidly. This is especially essential for languages like Bangla, where regional context and little changes in language play a large role in determining how violent content operates. The EMNLP BLP shared task on "Violence Inciting Text Detection" serves as a strong appeal to address this topic directly. One of our goals is to make a system that can handle the complicated language of Bangla. This will make it easier and more accurate to find material that encourages violence.The idea for our study came from the important work of (Saha et al., 2023b) and the creation of the Vio-Lens dataset (Saha et al., 2023a). The fundamental purpose of VITD is to detect and classify texts that contain components of incitement to violence. Vio-Lens, a unique annotated collection of over 10,000 Bangla social media posts, marks a significant advancement in detecting and addressing violence-inciting language. With this resource, we aim to push the boundaries of threat assessment in Bangla narratives, including those up to 600 words, seeking to not only identify evident risks but also redefine detection parameters. This research makes a valuable contribution to the wider effort to promote secure digital environments.Several study subjects that have been discussed in the literature are location-independent machine learning approaches for early fake news detection (Liu, 2019), combining audio and text elements to find violent incidents (Anwar, 2022), and the creation of new methods like feature-based Twitter sentiment analysis with enhanced denial handling (Gupta and Joshi, 2021). There is also an investigation into the possible use of a memristive LSTM network for sentiment analysis (Wen et al., 2021). The method used in this study is based on the political security threat prediction framework, which is a mix of a lexicon-based approach and machine learning methods (Razali et al., 2023). Additionally, the system has a racism detection model that leverages a stacked ensemble GCR-NN architecture (Lee et al., 2022). These initiatives demonstrate the applicability of mood analysis in several domains pertaining to security and social justice. To get further details on our research, refer to the publication titled "Sentiment Analysis of Tweets using Heterogeneous Multi-layer Network Representation and Embedding" (Gyanendro Singh et al., 2020). Moreover, a significant advancement is shown in the MC-BERT4HATE model's ability to detect hate speech across many languages and translations (Sohn and Lee, 2019). Even though a lot of work has been made, these improvements also show how hard it is to understand Bangla language. Sometimes, traditional models have trouble understanding all the details in this language.Our proposed methodology employs a diverse range of machine learning models to address the issues

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 236–240

mentioned above. The algorithms included in this set are Logistic Regression, Decision Tree, Random Forest, Multi-Naive Bayes, K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD). This is in addition to using deep learning architectures like Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and LSTM-CNN hybrids to adapt to the unique features of the Bangla spoken form. The inclusion of all individuals under this methodology facilitates the identification of and categorization of potential hazards, hence streamlining the process. [1] final implementation with an anonymous GitHub link[2]..

2 Literature Review

Due to violence-inciting content, social media is both connecting and alarming. We found a new answer to this essential issue, giving hope. Traditional machine learning and deep learning models classify violence-inciting literature in this study. This study built on natural language processing and hate speech identification research. The NLP survey on hate speech identification is useful for its problem formulations and methods (Schmidt and Wiegand, 2017). The transformative deep bidirectional transformer model BERT by (Devlin et al., 2019) has changed natural language comprehension research. (Van Hee et al., 2015) Cyberbullying detection and classification work shows that online safety awareness has enhanced cyberbullying detection beyond hate speech. (Zhou et al., 2019) and (Zampieri et al., 2019) participated in SemEval-2019 Task 6, which identified and categorized social media offensive language. (Wu et al., 2019) from BNU-HKBU UIC NLP Team 2 employed a BERT model to detect foul language, enriching this field. These studies show the importance of identifying and regulating offensive digital content. Study social media bullying traces and their prognostic potential for online safety (Xu et al., 2012). The necessity of studying protected traits has helped Burnap and Williams improve Twitter cyber hate detection (Burnap and Williams, 2016). Comment embeddings for hate speech identification advance the field and demonstrate their efficacy (Djuric et al., 2015). Mehdad and Tetreault illuminated character-level abusive encounters, improving our comprehension of abusive language (Mehdad and Tetreault, 2016). Due to variances in methods and datasets, these research' results vary in accuracy despite their importance. This comprehensive review uses multiple methodologies and data augmentation to fill this critical gap in our knowledge. We want to improve Bangla sentiment analysis and offensive language identification datasets and models. Our research will illuminate content filtering and internet safety in underrepresented languages.

3 Data and Methodology

In this section, we present the data sources and preprocessing steps, along with the methodology encompassing machine learning and deep learning models.

3.1 Dataset Description

The dataset utilized in our research was sourced from BLP Shared Task 1: Violence Inciting Text Detection (VITD), a valuable resource consisting of two key columns: "text" and "label." The "text" column encompasses textual content harvested from diverse social media platforms. For clarity and reference, we introduce "Label Definition" in Table 1, elucidating the categories assigned to each label within our dataset. Furthermore, Figure 1 illustrates a compelling word cloud visualization, spotlighting the most frequently occurring words in our datasets.

Table 1: Label Definition for BLP Shared Task 1

Label	Category	Total
Direct Violence	2	389
Passive Violence	1	922
Non-Violence	0	1389

Figure 1: Word Cloud Visualization for Three Label (Non-Violence, Passive Violence, Direct Violence)

[1] https://github.com/blp-workshop/blp_task1#leaderboard

[2] https://anonymous.4open.science/r/EMNLP_2023_BLP_Workshop_Task1-0FB2

3.2 Preprocessing

The dataset was collected from BLP Shared Task 1: Violence Inciting Text Detection (VITD), which is a shared task in the context of violence inciting text detection. The dataset encompasses a multitude of elements including symbols, URLs, and concealed characters. It also incorporates non-standard characters, Unicode control characters, emoticons, emojis, variations in whitespace, special formatting elements, non-alphanumeric characters, instances of duplicated or reiterated characters, and escape sequences, among others. Hence, we have executed multiple preprocessing procedures to eliminate the noise from the data. We also executed the following actions: elimination of short conversations, exclusion of lengthy conversations, removal of non-Bangla characters, filtering out Stopwords and non-Bangla characters, and Finally we apply stemming. To address the initial label imbalance in our dataset, we employed Up-sampling specifically for the "Direct Violence" category. Table 2 illustrates a comparison between the values before and after the pre-processing phase.

Table 2: Comparison of Data Before and After Preprocessing

Label	Before Preprocessing	After Preprocessing
Non Violence	1389	1336
Passive Violence	922	881
Direct Violence	389	750
Total	2700	2967

3.3 Models

In our study, we employed a diverse set of models, encompassing both deep learning and traditional machine learning approaches. The deep learning models included Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and a hybrid model, LSTM-CNN, each tailored for text classification. These models excel at capturing sequential information and local features within the text data. Additionally, we leveraged traditional machine learning models such as Logistic Regression, Decision Tree, Random Forest, Multinomial Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD). We purposefully chose the models for our study based on their distinct advantages and applicability to solving the challenging problem of identifying texts that incite violence. Here are the reasons we chose these models: LSTM chosen for its expertise in capturing sequential information, making it perfect for analyzing the complex language in texts that incite violence. CNN selected for its ability to identify structural patterns and components indicating violent content in text. Combines LSTM and CNN advantages, using local features and sequential information for comprehensive text classification. Traditional machine learning models chosen for their diverse techniques and effectiveness in text categorization.

3.4 Experimental Setup

To initiate the training of our traditional models, we first converted the preprocessed data into TF-IDF vectors. We went a step further by incorporating weighted n-grams, encompassing not only unigrams but also bigrams and trigrams. This strategy allowed us to harness contextual information more effectively, enhancing our model's understanding. We meticulously fine-tuned the model parameters to optimize performance and ensure the robustness of our deep learning-based classification approach, as detailed in Table 3. The dataset is divided into two subsets: "Training set" containing 2373 samples for model training, and "Test set" comprising 594 samples for evaluation.

4 Results and Discussion

In this section, we present the results of our experiments and engage in a comprehensive discussion of the findings. Our study aimed to address the challenge of violence inciting text detection using a combination of machine learning and deep learning models. We used various algorithms and techniques to analyze and classify text data into different categories of violence, namely Direct Violence, Passive Violence, and Non-Violence.

The machine learning models displayed varying degrees of performance in classifying violence inciting text in table 4. Notably, the Random Forest and Support Vector Machine (SVM) models outperformed the others in terms of accuracy and F1 score. These models achieved accuracy levels above 76.09%, demonstrating their effectiveness in distinguishing between different categories of violence.

Table 3: Experimental Setup for Deep Learning Models

Model	Embedding Dimension	Input Length	Vocabulary Size	Number of Classes	Batch Size	Number of Epochs
LSTM	128	300	5000	3	64	50
CNN	128	300	5000	3	64	50
LSTM-CNN Combine	128	300	5000	3	64	50

Table 4: Machine Learning Model Performance

Model Name	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Logistic Regression	73.91	75.26	73.91	72.11
Decision Tree	69.02	69.33	69.02	68.72
Random Forest	76.09	77.60	76.09	74.02
Multi. Naive Bayes	70.54	71.52	70.54	70.13
KNN	61.78	62.93	61.78	61.48
SVM	76.94	76.50	76.94	76.10
SGD	76.94	76.75	76.94	75.64

Table 5: Deep Learning Model Performance

Model	Class	Accuracy	Precision	Recall	F1-Score
LSTM	No-Violence	67.68	82.44	81.20	81.82
	Passive Violence		76.72	60.75	67.81
	Direct Violence		50.22	69.05	58.15
CNN	No-Violence	68.69	73.03	83.46	77.89
	Passive Violence		73.63	68.60	71.02
	Direct Violence		56.80	57.14	56.97
LSTM-CNN	No-Violence	66.50	64.85	80.45	71.81
	Passive Violence		74.60	64.16	68.99
	Direct Violence		56.50	59.52	57.97

Our ensemble of deep learning models, including LSTM, CNN, and LSTM-CNN, displayed strong performance in classifying violence-inciting text listed in table 5. It is evident that the CNN model has the highest accuracy at 68.69%, followed closely by the LSTM model with an accuracy of 67.68%. The LSTM-CNN hybrid model, while still respectable, trails slightly behind with an accuracy of 66.50%.

5 Conclusion

Our research underscores the critical importance of detecting and classifying violent incitement text within the realm of Natural Language Processing (NLP). Drawing inspiration from the EMNLP BLP shared assignment on Violence Inciting Text Detection and building upon the foundational work, we aimed to redefine the parameters of danger assessment in the context of the Bangla language. This study undertakes a comprehensive evaluation of machine learning and deep learning models to assess their effectiveness in categorizing literature that incites violence. Conventional machine learning algorithms, such as Logistic Regression, Decision Tree, Random Forest, Multi-Naive Bayes, KNN, SVM, and SGD, consistently demonstrate strong and reliable performance. Notably, Support Vector Machines (SVM) and Stochastic Gradient Descent (SGD) stand out for their efficacy in accurately classifying violent content. Deep learning models, including Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and the hybrid LSTM-CNN, also exhibit significant capabilities. LSTM, in particular, emerges as a standout performer among the deep learning models. This study's limitations include language and dataset specificity, data imbalance, model interpretability, and computational resource requirements. Future research may encompass multilingual expansion, contextual analysis, user-level profiling, ethical considerations, human-in-the-loop approaches, cross-domain application, and real-world deployment of violence-inciting text detection models.

References

A. Anwar. 2022. Deepsafety: Multi-level audio-text feature extraction and fusion approach for violence detection in conversations.

P. Burnap and M. L. Williams. 2016. Us and them: iden-

tifying cyber hate on twitter across multiple protected characteristics. *EPJ Data science*, 5:1–15.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pretraining of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30.

I. Gupta and N. Joshi. 2021. Feature-based twitter sentiment analysis with improved negation handling. *IEEE Transactions on Computational Social Systems*, 8(4):917–927.

L. Gyanendro Singh, A. Mitra, and S. Ranbir Singh. 2020. Sentiment analysis of tweets using heterogeneous multi-layer network representation and embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

E. Lee et al. 2022. Racism detection by analyzing differential opinions through sentiment analysis of tweets using stacked ensemble gcrnn model. *IEEE Access*, 10:9717–9728.

H. Liu. 2019. A location independent machine learning approach for early fake news detection. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4740–4746, Los Angeles, CA, USA.

Y. Mehdad and J. Tetreault. 2016. Do characters abuse more than words? In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*, pages 299–303.

N. A. M. Razali et al. 2023. Political security threat prediction framework using hybrid lexicon-based approach and machine learning technique. *IEEE Access*, 11:17151–17164.

S. Saha, J. A. Junaed, A. S. S. Api, N. Mohammad, and M. R. Amin. 2023a. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

S. Saha, J. A. Junaed, N. Mohammed, S. Kar, and M. R. Amin. 2023b. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

A. Schmidt and M. Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*.

H. Sohn and H. Lee. 2019. Mc-bert4hate: Hate speech detection using multi-channel bert for different languages and translations. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 551–559.

C. Van Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. De Pauw, et al. 2015. Detection and fine-grained classification of cyberbullying events. In *Proceedings of the international conference recent advances in natural language processing*, pages 672–680.

S. Wen et al. 2021. Memristive lstm network for sentiment analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(3):1794–1804.

Z. Wu, H. Zheng, J. Wang, W. Su, and J. Fong. 2019. Bnu-hkbu uic nlp team 2 at semeval-2019 task 6: Detecting offensive language using bert model. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

J. M. Xu, K. S. Jun, X. Zhu, and A. Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666.

M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval).

C. Zhou, J. Wang, and X. Zhang. 2019. Ynu-hpcc at semeval-2019 task 6: Identifying and categorising offensive language on twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

NLP_CUET at BLP-2023 Task 1: Fine-grained Categorization of Violence Inciting Text using Transformer-based Approach

Jawad Hossain, Hasan Mesbaul Ali Taher, Avishek Das and **Mohammed Moshiul Hoque**
Department of Computer Science and Engineering
@Chittagong University of Engineering & Technology, Chattogram-4349, Bangladesh
{u1704039, u1804038}@student.cuet.ac.bd, {avishek, moshiul_240}@cuet.ac.bd

Abstract

The amount of online textual content has increased significantly in recent years through social media posts, online chatting, web portals, and other digital platforms due to the significant increase in internet users and their unprompted access via digital devices. Unfortunately, the misappropriation of textual communication via the Internet has led to violence-inciting texts. Despite the availability of various forms of violence-inciting materials, text-based content is often used to carry out violent acts. Thus, developing a system to detect violence-inciting text has become vital. However, creating such a system in a low-resourced language like Bangla becomes challenging. Therefore, a shared task has been arranged to detect violence-inciting text in Bangla. This paper presents a hybrid approach (GAN+Bangla-ELECTRA) to classify violence-inciting text in Bangla into three classes: *direct*, *passive*, and *non-violence*. We investigated a variety of deep learning (CNN, BiLSTM, BiLSTM+Attention), machine learning (LR, DT, MNB, SVM, RF, SGD), transformers (BERT, ELECTRA), and GAN-based models to detect violence inciting text in Bangla. Evaluation results demonstrate that the GAN+Bangla-ELECTRA model gained the highest macro f_1-score (74.59), which obtained us a rank of 3rd position at the BLP-2023 Task 1.

1 Introduction

Violence-inciting text refers to textual content that promotes or glorifies acts of violence or harm towards individuals, groups, or entities, including hate speech and extremist ideologies. Detecting such text is crucial for preventing harmful activity and maintaining safety on social media. Social media's widespread use by diverse religious and cultural factions has led to weaponization, inciting hatred and causing communal violence, resulting in significant loss of life and destruction. This issue persists not only in a specific geographical region but also globally, escalating the longstanding issue. This paper aims to classify various forms of communal violence to illuminate this complex phenomenon and contribute to its mitigation.

Violence has evolved with society's advancements, with physical and psychological abuse now predominantly occurring online and on social networks, even though it was once face-to-face (Golem et al., 2018). Previous studies reveal that social media platforms incite political and religious violence, thereby threatening communal harmony and societal stability (Patton et al., 2014). Social networks have become a virtual civilization where people share views, feelings, photos, videos, and blogs. However, there is no defined mechanism for restricting violent content on these platforms (Yadav and Manwatkar, 2015). In recent years, tech giants like Facebook, YouTube, and Twitter have been striving to achieve this goal (Ghanghor et al., 2021). However, it is impossible to manually monitor these violent inciting contents that surf social media (Sharif and Hoque, 2022). Therefore, developing such a system for detecting violence-inciting text is crucial to reducing illegal behavior and maintaining a clean information ecosystem.

This work aims to build a system that can detect violence inciting text from Bangla text concerning three different categories. This work's key contributions are illustrated in the following:

- Developed a hybrid model using GAN and Bangla-ELECTRA to detect and classify violence-inciting Bangla texts into three groups: direct violence (DV), passive violence (PV), and non-violence (NV).
- Investigated the model's effectiveness in detecting and classifying violence-inciting texts by comparing several ML, DL, and transformer-based models and analyzed in-depth errors, offering valuable insights into violence-inciting text detection.

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 241–246

2 Related Work

While providing platforms for individual freedom of expression, social media and other blogging platforms can facilitate antisocial conduct, including hate speech, cyberbullying, and online harassment (Karim et al., 2021). Several works has been conducted to detect aggressive comments (Sharif and Hoque, 2022), abusive comment, hate speech (Das et al., 2021), trolling (Zampieri et al., 2019). However, few studies have been conducted to detect violence-inciting text. Though several works have been done in high-resource languages, leaving low-resource languages like Bangla out of the focus. To identify abusive language, Eshan and Hasan, 2017 utilized a dataset comprising 2.5k instances of abusive Bangla text and evaluated the performance of several ML models (RF, NB, and SVM) and achieved a maximum accuracy of 85% using SVM with linear kernel and tri-gram features. Kumar et al., 2018 categorized 15k English and Hindi comments on aggression into overtly aggressive, covertly aggressive, and non-aggressive categories, expanding the corpus to include Bangla aggressive comments (Kumar et al., 2020). Aroyehun and Gelbukh, 2018 studied the effectiveness of DNN models in detecting aggression using enhanced data and pseudo-labeled samples. Ishmam and Sharmin, 2019 classified 5k Bangla abusive Facebook comments into six categories using a GRU-based model, achieving 70.10% accuracy. The introduction of BERT-based models significantly enhanced performance, surpassing all previous models on these datasets (Risch and Krestel, 2020, Sharif et al., 2021). Sharif et al., 2021 presented a Bangla aggressive text dataset, and later, they extended the previous dataset to create a new novel dataset named *BAD*. They used a transformer-based ensemble technique to identify and categorize aggressive texts in Bangla, achieving the highest weighted scores of 93.43% (coarse-grained) and 93.11% (fine-grained). As per our exploration, none of the past studies addressed classifying the violence-inciting texts in Bangla. This work uses a hybrid approach incorporating GAN and Bangla-ELECTRA models to address the downstream task.

3 Task and Dataset Descriptions

Task organizers[1] created a gold standard corpus to detect violence-inciting language in social media. To address this phenomenon, Saha et al., 2023 developed a Violence Inciting Text Detection (VITD) corpus[2] in the Bangla language. The task aims to implement a system that can detect offensive texts. The corpus consists of the text of three different classes: *non-violence*, *passive violence*, and *direct violence*. According to Saha et al., 2023, the definition of each class is illustrated in the following:

- **Direct Violence (DV):** Texts expressing explicit threats fall under direct violence.
- **Passive Violence (PV)**: Texts containing abusive or derogatory use of language.
- **Non-Violence (NV)**: The non-violence category consists of any discussions conducted by texts that do not involve any form of violence.

The dataset (VITD) accumulated 6046 texts from YouTube comments in Bangla. VITD is related to nine violent incidents during the previous 10 years. The task aims to quickly distinguish between violent threats to stop further incitement to violent acts. Contribution to the identification and prevention of stimulation to violent acts online is the primary goal of this task.

Table 1 illustrates the detailed statistics of the dataset. The dataset consists of training, validation, and test sets containing 2700, 1330, and 2016 texts. The dataset is imbalanced as there are more non-violence samples than direct and passive violence combined. The non-violence class includes the highest data (1389 texts) with 7128 unique words.

Table 1: Summary of the dataset statistics.

Classes	Train	Valid	Test	Total words
DV	389	196	201	13202
PV	922	417	719	39423
NV	1389	717	1096	54333
Total	2700	1330	2016	106958

We further analyzed the dataset in terms of sentence length. Figure 1 shows the length-frequency distribution of the dataset. The analysis of the length-frequency distribution revealed that there were fewer than 50 text samples whose text length was more than 128 words. Thus, this work used a maximum input sentence size of 128 words. The minimum sentence length is one word, whereas the average length is 18 words.

[1] https://blp-workshop.github.io/sharedtasks

[2] https://github.com/blp-workshop/blp_task1

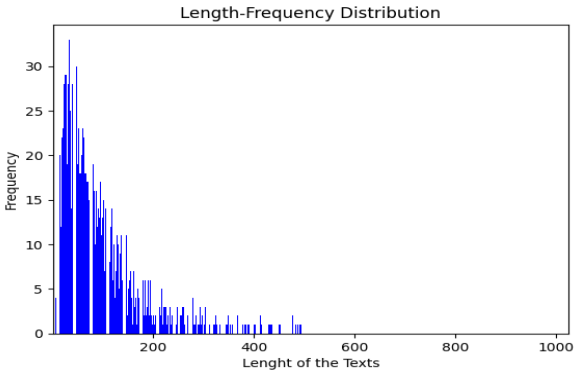


Figure 1: Length-frequency distribution of the dataset

4 Methodology

This work exploited several ML, DL, and transformer-based models to address the task. After investigating all models' performance, this work proposes a hybrid method using GAN and Bangla-ELECTRA to detect and classify violence-inciting Bangla texts. We used the 'scikit-learn' and TensorFlow library to build ML and DL models. Figure 2 shows an abstract view of the proposed system.

First, the unwanted characters (URLs, punctuation, and whitespace) are removed from the texts. We apply different feature extraction techniques (i.e., TF-IDF, Word2Vec) to extract the textual features. This work employed six traditional ML models, such as logistic regression (LR), decision tree (DT), support vector machine (SVM), multinomial naive Bayes (MNB), random forest (RF), and stochastic gradient descent (SGD). We also used three DL methods, such as CNN, BiLSTM, and BiLSTM, with attention.

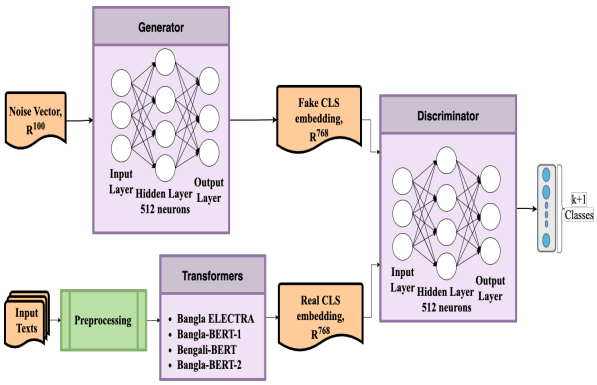


Figure 2: Proposed hybrid model using GAN and Bangla-ELECTRA to detect and classify violence inciting Bangla texts.

This work employed four transformers fetched from HuggingFace[3] library. We built the transformer models with PyTorch library to tackle the task, such as Bangla-ELECTRA (Bhattacharjee et al., 2021), Bangla-BERT-1 (Sarker, 2020), Bangla-BERT (Joshi, 2022), and Bangla-BERT-2 (Kowsher et al., 2022).

4.1 GAN+Bangla-ELECTRA

In the GAN, we used 2 sub-networks: generator and discriminator. The generator takes input noise and outputs fake data, which tries to resemble the original data distribution. The discriminator is trained over a $(k+1)$-class objective: the *true* examples are classified in one of the target $(1, ..., k)$ classes, while the generated samples are classified into $k+1$ class. The adversarial training procedure is applied (Goodfellow et al., 2020). The generator is penalized each time the discriminator discovers its output as fake. The discriminator is penalized each time the generator fools it; that is, it can identify the fake data created by the generator as real.

In the GAN+transformer-based approach (Croce et al., 2020), we consider labeled and unlabeled data where unlabeled data is accumulated by removing the label. The generator and discriminator are both multilayer perceptrons with a single hidden layer of 512 neurons. The input of the generator is a randomly generated vector of 100 dimensions, and it outputs a fake transformer embedding vector for a single token. The transformer-based model (BERT, ELECTRA) feeds the input text, generating a contextualized embedding vector of the CLS token. The embedding vectors generated by both the transformer and generator are used as input for the discriminator. The input of the discriminator can be expressed by Eq. 1.

$$H_* \epsilon R^D \tag{1}$$

Where, H_* can be either H_{FAKE} or H_{CLS}. H_{FAKE} denotes the outputs of the generator and H_{CLS} is the output of the transformer model. The output of the discriminator is extended to $k+1$ classes, where k is the number of classes, and the extra class is 'REAL'. The system identifies using $k+1^{th}$ class whether the embedding encoded by the transformer-based architecture is real or not. The goal is to acquire a good discriminator in k-class classification. The discriminator and final classification are defined by Eqs. 2-3.

$$D_{logits} = MLP(x) \tag{2}$$

[3]https://huggingface.co/models

$$P_{class_i} = \frac{e^{D_{logits_i}}}{\sum_{k=1}^{k+1} e^{D_{logits_k}}} \quad (3)$$

Here D_{logits} is the output of passing the input vector 'x' through the multi-layer neural network of the discriminator. P_{class_i} denotes the probability of a text sequence belonging to a given class.

A dropout rate of 0.1 is added to both the generator and discriminator architecture to prevent overfitting. The Adam optimizer with a batch size of 16 and a learning rate of 5e-5 is used to train the models for 10 epochs. For testing, we just discard the generator and use the BERT and discriminator model to classify the input data. We mask the prediction output for the 'REAL' class in testing.

5 Results

The efficacy of the models is determined based on the macro-F1 score (MF1). However, we also consider the precision (P) and recall (R) metrics to perform the analysis. Table 2 illustrates the performance of employed models for the task. Among the ML models, SGD achieved the highest MF1 score of 65.34, while LR performed poorly on the test set. On the other hand, DL-based methods did not surpass the performance of the best ML model (MF1 score of 65.34). Low amounts of data samples might cause this, as DL models are generally data-hungry. Adding attention (Vaswani et al., 2017) to BiLSTM improved its performance by almost 3.12%. All transformer-based models outperformed the ML and DL models, with Bangla-BERT scoring the highest (68.36). Although the GAN-based transformer models improved the scores of their respective transformers, the Bangla-BERT-based standalone and GAN-based models performed almost identically. GAN+Bangla-ELECTRA outperformed all the models, achieving the highest f1-score of 74.59. With the GAN+transformer approach, the inner representation of BERT is being fine-tuned by both labeled and unlabeled data. For this reason, the inner representation of BERT is more robust towards unseen data points.

Table 2: Performance of various models on the test set

Classifier	**P**	**R**	**MF1**
LR	63.08	57.34	29.28
DT	59.89	59.72	53.11
RF	71.88	68.01	59.92
MNB	69.07	68.80	63.91
SVM	73.01	65.62	55.50
SGD	71.34	70.68	65.3
CNN	66.67	65.58	57.26
BiLSTM	67.72	66.91	60.02
BiLSTM + Attention	67.83	67.81	61.89
Bangla-ELECTRA	72.34	72.77	67.18
Bangla-BERT-1	71.88	71.92	66.45
Bangla-BERT	76.13	73.12	68.36
Bangla-BERT-2	75.25	72.97	67.05
GAN+Bangla-BERT-1	71.31	71.23	66.33
GAN+Bangla-BERT-2	75.04	74.21	69.66
GAN+Bangla-BERT	76.32	76.49	72.35
GAN+Bangla-ELECTRA	**77.98**	**77.43**	**74.59**

Table 3 shows the class-wise performance (MF1) of hybrid models. Results demonstrated that the proposed approach (GAN+Bangla-ELECTRA) attained the highest scores in all classes than the other hybrid models.

Table 3: Class-wise violence inciting text detection performance on the test set

Class	NV	PV	DV
GAN+Bangla-BERT-1	0.79	0.62	0.58
GAN+Bangla-BERT-2	0.81	0.68	0.60
GAN+Bangla-BERT	0.82	0.70	0.65
GAN+Bangla-ELECTRA	**0.82**	**0.73**	**0.69**

5.1 Error Analysis

A detailed error analysis is performed quantitatively and qualitatively to provide in-depth insights into the performance of the proposed model.

Quantitative Analysis: A quantitative error analysis of the best-performed model is done using the confusion matrix (Fig. 3). The proposed GAN+Bangla-ELECTRA classified a total of 1561 samples correctly out of 2016 samples in the test dataset. The model did comparatively better results in the NV class. The model identified 910 instances of the NV class correctly. It incorrectly classified 171 samples as NV class of which 150 data samples were originally from PV and 21 data samples were originally from DV. The model becomes more confused between NV and PV as it misclassified a total of 311 instances between the two classes, whereas the instances that were misclassified as DV and the DV true instances that were misclassified as NV or PV total only 144. This may happen because a regular discussion with one person might be a derogatory or abusive use of language to another, as some words can be used for both peaceful and violent discussions.

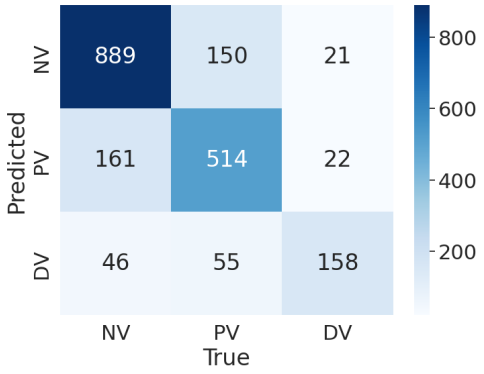


Figure 3: Confusion matrix of the proposed model ((GAN+Bangla-ELECTRA)

Qualitative Analysis: Figure 4 illustrates some predicted outcomes by the proposed model. The

Text sample	Actual	Predicted
Sample1. 'সাংবাদিকতা করতে এসে ঠান্ডা সন্ত্রাসী বন্ধ করো।' (Stop being a cold terrorist when you come to do journalism.)	PV	PV
Sample2. 'ভারতীয় দালাল নির্মূল এখন সময়ের দাবি, ভারত হলো বাংলাদেশের সকল সমস্যার মূল হোতা।' (Eradication of Indian brokers is the need of the hour, India is the root of all problems in Bangladesh.)	PV	DV
Sample3. 'এরশাদ ভালো না হইলেও ওর ভাই জিএম কাদের ভালো আছে।' (Even if Ershad is not good, his brother GM Kader is good.)	NV	NV
Sample4. 'সত্য কথা বলার জন্য ধন্যবাদ।' (Thanks for speaking the truth.)	NV	NV
Sample5. 'মোসল্লীরা ইটপাটকেল ছুড়তে দেখলাম না মিথ্যা বলছেন কেন?' (I didn't see Muslims throwing bricks, why are they lying?)	NV	PV

Figure 4: Few examples of predicted outputs by the proposed (GAN+Bangla-ELECTRA) model

proposed model correctly predicts text samples 1, 3, and 4, whereas text samples 2 and 5 are not predicted correctly. Text sample 2 is wrongly predicted as DV, whereas the actual class is PV. Similarly, text sample 5 is incorrectly predicted as PV instead of actual class (NV). The class imbalance issue might be the reason for wrong predictions, as a few instances of the DV class (201 samples) are available in the dataset. This scarcity of samples may be inadequate for the model to learn. Another reason might be that the words used in DV do not often overlap with the largest class (i.e., NV).

6 Conclusion

This work addresses the challenge of fine-grained classification of texts inciting violence in Bangla. We developed a solution by leveraging a benchmark dataset known as VITD. In this paper, we systematically investigated and compared 17 different baseline models, spanning various machine learning (ML), deep learning (DL), transformer, and generative adversarial network (GAN) architectures. The experimentation revealed that integrating GANs with transformers resulted in improved task performance. Specifically, the combination of GAN and Bangla-ELECTRA demonstrated the highest macro F1-score (74.59) among all the models we employed, surpassing their performance. We intend to enhance our solution by leveraging ensemble techniques in future research endeavors. Additionally, we will delve into the impact of resampling strategies on model performance, mainly as our dataset exhibits imbalance issues.

References

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Samin, M Sohel Rahman, Anindya Iqbal, and Rifat Shahriyar. 2021. Banglabert: Combating embedding barrier for low-resource language understanding. *arXiv preprint arXiv:2101.00204*.

Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2020. Gan-bert: Generative adversarial learning for robust text classification with a bunch of labeled examples.

Amit Kumar Das, Abdullah Al Asif, Anik Paul, and Md Nur Hossain. 2021. Bangla hate speech detection on social media using attention-based recurrent neural network. *Journal of Intelligent Systems*, 30(1):578–591.

Shahnoor C Eshan and Mohammad S Hasan. 2017. An application of machine learning to detect abusive bengali text. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Nikhil Ghanghor, Parameswari Krishnamurthy, Sajeetha Thavareesan, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2021. Iiitk@ dravidianlangtech-eacl2021: Offensive language identification and meme classification in tamil, malayalam and kannada. In *Proceedings of the first workshop on speech and language technologies for Dravidian languages*, pages 222–229.

Viktor Golem, Mladen Karan, and Jan Šnajder. 2018. Combining shallow and deep learning for aggressive text detection. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 188–198.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative

adversarial networks. *Communications of the ACM*, 63(11):139–144.

Alvi Md Ishmam and Sadia Sharmin. 2019. Hateful speech detection in public facebook pages for the bengali language. In *2019 18th IEEE International Conference on machine learning and Applications (ICMLA)*, pages 555–560. IEEE.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Md Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Md Azam Hossain, and Stefan Decker. 2021. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE.

Md Kowsher, Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, 10:91855–91870.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2020. Evaluating aggression identification in social media. In *Proceedings of the second workshop on trolling, aggression and cyberbullying*, pages 1–5.

Ritesh Kumar, Aishwarya N Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018. Aggression-annotated corpus of hindi-english code-mixed data. *arXiv preprint arXiv:1803.09402*.

Desmond Upton Patton, Jun Sung Hong, Megan Ranney, Sadiq Patel, Caitlin Kelley, Rob Eschmann, and Tyreasa Washington. 2014. Social media as a vector for youth violence: A review of the literature. *Computers in Human Behavior*, 35:548–553.

Julian Risch and Ralf Krestel. 2020. Bagging bert models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Omar Sharif and Mohammed Moshiul Hoque. 2022. Tackling cyber-aggression: Identification and fine-grained categorization of aggressive texts on social media using weighted ensemble of transformers. *Neurocomputing*, 490:462–481.

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2021. Nlp-cuet@ dravidianlangtech-eacl2021: Offensive language detection from multilingual code-mixed text using transformers. *arXiv preprint arXiv:2103.00455*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Shashank H Yadav and Pratik M Manwatkar. 2015. An approach for offensive text detection and prevention in social networks. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–4. IEEE.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Team_Syrax at BLP-2023 Task 1: Data Augmentation and Ensemble Based Approach for Violence Inciting Text Detection in Bangla

Omar Faruqe Riyad
Shahjalal University of Science & Technology
omar42@student.sust.edu

Trina Chakraborty
Shahjalal University of Science & Technology
trina41@student.sust.edu

Abhishek Dey
Shahjalal University of Science & Technology
abhishek21@student.sust.edu

Abstract

This paper describes our participation in Task 1 (VITD) of BLP Workshop [1] at EMNLP 2023, focused on the detection and categorization of threats linked to violence, which could potentially encourage more violent actions. Our approach involves fine-tuning of pre-trained transformer models and employing techniques like self-training with external data, data augmentation through back-translation, and ensemble learning (bagging and majority voting). Notably, self-training improves performance when applied to data from external source but not when applied to the test-set. Our analysis highlights the effectiveness of ensemble methods and data augmentation techniques in Bangla Text Classification. Our system initially scored 0.70450 and ranked 19th among the participants but post-competition experiments boosted our score to 0.72740.

1 Introduction

In today's social media-driven world, easy self-expression has brought a downside: a surge in harmful, violent content harming people physically and mentally (Mathew et al., 2019). This critical concern needs addressing.

The EMNLP-2023 organized the BLP Shared Task 1 (VITD) (Saha et al., 2023a), addressing a vital challenge: identifying violence-inciting text in Bangla. The aim was to build models that identify violent content, especially content that might provoke more violence. Yet, in Bangla, this is tough due to limited language resources. The text from YouTube comments often lacks clear context, making it even harder to understand. Additionally, the dataset (Saha et al., 2023b) used in this task is relatively small, limiting the variety of language patterns. To overcome these issues, we used pre-trained transformer models, fine-tuning them with VITD dataset. We also applied techniques like self-training on external data, back-translation for data augmentation, and ensemble learning (Bagging and Majority Voting). These techniques, particularly when combined with self-training and back-translation, as well as Ensemble approach across multiple models, moderately improved our model's performance.

Post-competition experiments, including self-training on external data and back-translation, raised our score to 0.72740. This paper details our approach, challenges, and methods for addressing violence-inciting text in Bangla.

2 Related Work

Hate speech, cyberbullying, harassment, and incitements to violence on social platforms can harm individuals and communities in online spaces. Increasing studies have been undertaken to detect violent content on social media (Dikwatta and Fernando, 2019; Jahan and Oussalah, 2023a; Zampieri et al., 2020). People usually confront violence on social media through text, images, and videos. Researchers use natural language processing (Jahan and Oussalah, 2023b) to analyze text, visual, and audio content on social media sites. These excellent initiatives are happening worldwide in many languages. Implementing the same method in languages with low resources, like Bangla, is problematic (Das et al., 2022a). Poorly annotated Bangla-language violence detection datasets are a widespread issue (Romim et al., 2022). Bangla has a large vocabulary and several sentence forms (Das et al., 2022b). Bangla dialects vary by region, which might alter text interpretation. Although Bangla is a low-resource language (Karim et al., 2021) with its own issues, numerous studies (Emon et al., 2022) are being undertaken to identify social media breaches in this language context. Modern models, such as BERT (Mridha et al., 2021), have been substantially altered and used in these studies. These evolving investigations are encouraging us to use these modern approaches for low-resource lan-

[1] https://blp-workshop.github.io/

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 247–254

guages like Bangla (Keya et al., 2023; Kumari et al., 2023). Changing studies have provided new perspectives on violence recognition in Bangla (Jahan et al., 2022; Caselli et al., 2020) and expanded our knowledge of it. BanglaBert (Sharif et al., 2022) was a key tool in our study for addressing BLP Task 1. We wanted to get the most out of ensemble methods by using pre-trained transformers in our experiments (Risch and Krestel, 2020). This is because the field of transformer model applications is still growing and changing (Das et al., 2023). We were able to use the combined knowledge of several cutting-edge transformer models with this new method, which made our experiments more in-depth and varied. Several well-thought-out tests with multiple models have yielded key results and refined our method to make it more accurate.

3 Task and Dataset Description

The BLP Shared Task 1, known as Violence Inciting Text Detection (VITD), offers an outstanding chance to address the significant problem of detecting violence-inciting text. The dataset being analyzed consists of YouTube comments containing the most significant violent incidents. Three distinct categories are established for the purpose of classification: Direct Violence, which includes explicit threats targeted towards individuals or communities; Passive Violence, which involves the utilization of derogatory language, abusive remarks, or justification of violence; and Non-Violence, which encompasses content that is unrelated to subjects involving violence. The task has a role in the identification and mitigation of potential threats that may lead to violent situations.

The VITD dataset is divided into three subsets: the training set, development set, and testing set, all of which are formatted in CSV structure. Each entry within these CSV files consists of two key columns: "text" and "label." The "text" column contains textual data collected from various social media sources, while the "label" column assigns a numerical value of 0, 1, or 2 to each entry, representing different categories of violence: Non-Violence, Passive Violence, and Direct Violence, respectively. In Appendix A.1, as shown in Figure 4, we tried to visualize the category distribution within each set and noticed that, the datasets are highly skewed towards Non-Violence. Occurrence of Direct Violence is very rare. The class distribution within the dataset is imbalanced, with Non-Violence being the dominant category. Detecting and classifying the less frequent instances of Passive Violence and Direct Violence poses a significant challenge. We also tried to visualize the texts associated with the labels through wordclouds in figure 5, 6, 7 in the A.1 appendix section. The distribution of words in the wordcloud provides some insights. We discovered some words that are uniquely associated with a given label. Along with that, we also noticed, the datasets contain instances of ambiguous labeling, in which the categorization of text into the correct category of violence is difficult due to the complexity and ambiguity of the language. Given the nature of text inciting violence, the dataset may contain instances of religious bias. During annotation, it is crucial to deal with this sensitivity and maintain an ethical perspective.

4 System Description

4.1 Data Pre-processing

In our data processing pipeline, cleaning and preprocessing the text data were involved as a necessary step. This was a meticulous and essential process that aimed to enhance the quality and reliability of the information we were working with. To begin with, we focused on the elimination of unwanted elements in the text. This included the removal of emojis and excess punctuation marks. Emojis, while adding expressive elements to text, are often regarded as noise in many natural language processing tasks. Removing emojis was essential to simplify the text and make it more amenable to analysis and modelling. Additionally, excess punctuation, such as multiple consecutive exclamation marks or question marks, can disrupt the flow of the text and create challenges for subsequent processing. By clearing the text of such redundancy, we aimed to make it cleaner and more straightforward. However, it's worth noting that we made a conscious decision not to remove Bangla stop words during this pre-processing stage. Stop words are commonly occurring words in a language are often excluded from text analysis because they don't carry substantial meaning on their own. However, when working with the Bangla language, we found that removing these stop words could sometimes alter the intended meaning of the text. To avoid such unintended alterations in meaning, we decided to retain Bangla stop words in our preprocessing steps.

4.2 Transformer based model

Transformer utilizes a mechanism called self-attention to process words in parallel, enabling it to capture intricate relationships and nuances within the text (Vaswani et al., 2017). By employing large-scale pre-training on vast text corpora, transformers gain a deep understanding of language. This general language knowledge, when fine-tuned for specific tasks, empowers them to excel in various applications including text classification. In our study of transformer based models for Bangla, we considered three main options: BanglaBert (Bhattacharjee et al., 2022), XLM-R (Conneau et al., 2019), and mBERT (Devlin et al., 2018). Both XLM-R and mBERT are pre-trained on a large amount of multilingual textual data but BanglaBert stands out due to its specific training on a large Bangla text dataset. This focused training equips it with a deep understanding of Bangla's unique language patterns, making it more effective than generic "BERT" models. It performs especially well in low-resource scenarios.

4.3 Semi-Supervised Learning: Self-Training

The VITD dataset is relatively small and has imbalanced class distribution (described in section 3). To address this, we adopted a semi-supervised learning method called self-training (Dong and de Melo, 2019). Initially, we trained our model on the train-set. Then, we used this model to label additional unlabeled data, expanding our training dataset. When we used test-set predictions as additional data, our model performed well in dev-set but not on the test-set. This happened because the test set contained some incorrect labelling from the model predictions. Additionally, we utilized self-training with external data. We selected 1500 data points from a Bangla Hate Speech dataset (Karim et al., 2020) and automatically annotated them. We filtered the newly annotated data, keeping all data points with labels 1 and 2 but only some with label 0 randomly, focusing on minority classes. Then, we combined this enriched dataset with our original training data. While this strategy resulted in a slight performance boost, it also diversified our dataset with a wider range of samples.

4.4 Data Augmentation: Back-Translation

We used back-translation technique (Sennrich et al., 2016) to increase diversity and size of data. We created a new dataset by translating Bangla sentences to English and back to Bangla using the Googletrans [2] API. We randomly combined the new dataset with the original data. This method enhances words and sentence variations by representing the words with semantic similarity in different form. Moreover, the VITD dataset, which includes YouTube comments, contains many grammatical errors and spelling mistakes. Back-translation using the Googletrans API corrects a significant portion of these errors. Combining both the back-translated data and the original data for training allows the model to recognize their semantic similarity and thus improving performance. It's essential to highlight that we conducted a manual quality check on the back-translated data to ensure its integrity and semantic similarity with the original dataset.

4.5 Ensembling

To enhance the robustness of our complex Transformer models, which tend to be sensitive to factors like initialization and data order, particularly when fine-tuned on small datasets (Dodge et al., 2020), we implement an ensemble method based on bootstrap aggregating (bagging) (Risch and Krestel, 2020) and hard majority voting. Bagging involves training multiple instances of the same model on various subsets of the training data through random re-sampling. This introduces randomness and reduces variance in the training process. In our study, we utilized seven different models for majority voting. The first model was trained on BanglaBert, while the second model was trained using a self-training approach on the first model. The remaining five models employed bagging, where we augmented the train-set with the dev-set. The final prediction was determined by taking the majority voting of individual model predictions. This ensemble strategy illustrated in Figure 2 was our best performing system during competition.

In the post-competition experiments, we implemented a majority voting system involving three top-performing models (Figure 1). The first model used a combination of the train-set and model-annotated external data. The second model combined the train-set, back-translated train-set, and back-translated dev-set. The third model was a result of a majority voting ensemble involving various experimented models. If there was a tie in the votes for two or more labels, we selected the label based on the model with the highest F1-score

[2] https://pypi.org/project/googletrans/

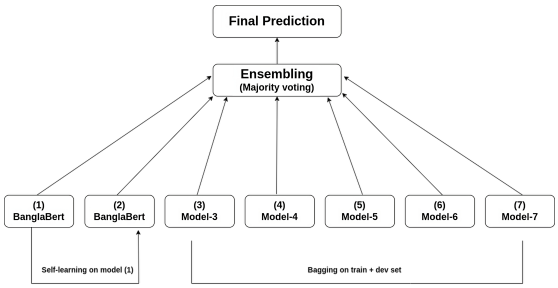


Figure 1: The overall best performing system

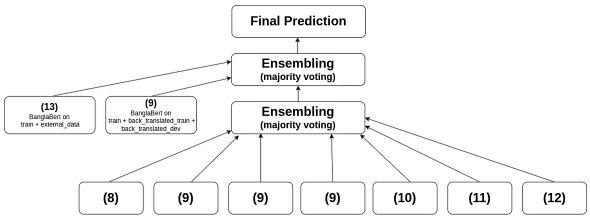


Figure 2: The best performing system during competition

(model 9). Our final macro-F1 score improved to 0.72740 compared to our competition performance, which achieved a macro-F1 score of 0.70450.

5 Experiment and Results

5.1 Experimental Setup

We utilized the Huggingface Transformers [3] library to construct our system. We employed pre-trained tokenizers and language models for fine-tuning. Training was conducted with a learning rate of 1e-5 and a batch size of 16. AdamW (Loshchilov and Hutter, 2019) optimizer is used to update the parameters. Model performance was assessed every 250 steps, with metrics including accuracy, precision, recall, and macro-F1 scores. Training lasted for 50 epochs, with early stopping implemented to select the best checkpoint based on the highest validation macro-F1 score. Our code is publicly available at Github [4].

Model	Macro-F1	
(1)	0.70296	
(2)	0.69288	
(3)	0.70079	Bagging
(4)	0.67752	
(5)	0.67632	
(6)	0.70280	
(7)	0.70919	
(8)	0.71326	Back Translation Included
(9)	**0.71977**	
(10)	0.70519	
(11)	0.70521	
(12)	0.71136	
(13)	0.71866	Extra-Data

Table 1: Individual Model Performance Metrics

5.2 Results

In this section, we present the performance results of our trained models [5], evaluated on the test-set released at the end of the competition.

Table 1 showcases the macro F1-scores of various models we experimented with, both during and after the competition. Notably, the dev-set scores were the main factor of our model selection during the evaluation phase. However, we observed that the best-performing model on the dev-set did not always translate to superior performance on the test-set. For instance, while model (2) outperformed model (1) on the dev-set, but this wasn't the case in the test-set. Our analysis revealed that the inclusion of back-translated data and model-annotated external data moderately improved model performance. For example, model (13), which is BanglaBert trained on the train-set combined with the model-annotated external dataset, achieved a macro F1-score of 0.71866. Model (9), which is BanglaBert trained on the train-set, back-translated train-set, and back-translated dev-set, achieved the highest macro F1-score of 0.719771 among models without utilizing majority voting.

To further improve our results, we employed ensemble methods. Table 2 presents the macro F1-scores of our ensemble approach. The first model (E1) with an F1-score of 0.70450 represented our final submission during the competition. One thing to note from second ensemble method (E2) that, we incorporated 3 votes from model (9), as it consistently demonstrated the highest accuracy throughout our experiments. Our post-competition experimentation unveiled that the third model (E3) exhibited a score of 0.72740 which is the highest overall F1-score. This was attained by employ-

[3] https://huggingface.co/docs/transformers/index
[4] https://tinyurl.com/bde9cf6w
[5] The models corresponding to the numbers in the figure can be found in the Table 3 of Appendix A.2

Model	Ensemble	Macro-F1
(E1)	(1)(2)(3)(4)(5)(6)(7)	**0.70450**
(E2)	(8)(10)(11)(12)(9)(9)(9)	0.71808
(E3)	(13)(9)(E2)	**0.72740**

Table 2: Ensemble Model Performance Metrics

ing a majority voting strategy among the three best-performing models. The experimental results emphasizes the significance of ensemble methods and data augmentation techniques in improving the detection of Violence Inciting Text in the Bangla language. The inclusion of back-translated data and model-annotated external data enriched our training dataset, leading to substantial performance gains.

6 Limitations and Error analysis

Error analysis is challenging in this task. A model may fail on certain datasets for many reasons. Our top performing model and the test dataset indicate the model's inaccurate classification of certain texts as direct violent or passive violent, and vice versa. Disparities in dataset labeling are a big issue. Why certain texts are labeled "2" for direct violence and others "1" for passive-violent texts is unclear. For instance, "2" is placed next to Figure 3-a and "1" is placed next to Figure 3-b. Religious biases of annotators should also be considered. This prejudice is evident when some texts are termed passive-violent and others comparable to them as non-violent or directly violent. Additionally, single-word messages like Figure 3-c are arbitrarily allocated the label "1" creating ambiguity. The inaccurate classification of shorter texts is due to lack of context. The model reveals classification accuracy of longer texts differ from shorter ones. The model's emphasis on the words of a sentence may explain this discrepancy. Longer sentences strengthen the model's contextual foundation, enabling more exact classification. After thoroughly studying the test set, we observed 472 label discrepancies between test set labels and best model predictions. Our model identified 207 of these texts as non-violent (label 0), while the test set classified them as passive-violence (label 1). The second greatest label differences was 91 instances between the test set's identification of texts as non-violent (label 0) and our model's labeling as Passive Violence (label 1). More than 50% of the mistakenly predicted classifications are Non-Violence and Passive Violence. This gap may be due to subtle distinctions between indirect Passive Violence and Non-Violence sentences. Besides, back-translation data augmentation improved model performance, but it might alter text meaning and structure, therefore NLP tasks should be used with caution. It is important to evaluate this potential impact on augmented data quality.

a. "তোদের উপর আল্লাহর গজব পড়বে"
b. "আল্লাহ তুমি এদের উপর গজব নাজিল করো"
c. "দালাল"

Figure 3: Examples of texts from train dataset about ambiguous labeling

7 Conclusion and Future Work

The objective of this research was to classify texts into three groups and determine whether or not they promote violence in any way. We have experimented with some prominent transformer based models for text classification before trying out other approaches to make those models perform better. After the test set was made public, we were able to strengthen the performance of our model by running further tests. In order to accurately identify violent texts in social media comments, there is still work to be done in the future. It is necessary to conduct more and more experiments with low resource languages like Bangla. We think that our efforts prepared the groundwork for this to happen.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Avishek Das, Mohammed Moshiul Hoque, Omar Sharif, M. Ali Akber Dewan, and Nazmul Siddique. 2023. Temox: Classification of textual emotion using ensemble of transformers. *IEEE Access*, 11:109803–109818.

Mithun Das, Somnath Banerjee, Punyajoy Saha, and Animesh Mukherjee. 2022a. Hate speech and offensive language detection in bengali. *arXiv preprint arXiv:2210.03479*.

Rajesh Kumar Das, Samrina Sarkar Sammi, Khadijatul Kobra, Moshfiqur Rahman Ajmain, Sharun Akter khushbu, and Sheak Rashed Haider Noori. 2022b. Analysis of bangla transformation of sentences using machine learning. In *International Conference on Deep Learning, Artificial Intelligence and Robotics*, pages 36–52. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

U Dikwatta and TGI Fernando. 2019. Violence detection in social media-review. *Vidyodaya Journal of Science*, 22(2).

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Xin Dong and Gerard de Melo. 2019. A robust self-learning framework for cross-lingual text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6306–6310, Hong Kong, China. Association for Computational Linguistics.

Md Imdadul Haque Emon, Khondoker Nazia Iqbal, Md Humaion Kabir Mehedi, Mohammed Julfikar Ali Mahbub, and Annajiat Alim Rasel. 2022. Detection of bangla hate comments and cyberbullying in social media using nlp and transformer models. In *International Conference on Advances in Computing and Data Sciences*, pages 86–96. Springer.

Md Saroar Jahan, Mainul Haque, Nabil Arhab, and Mourad Oussalah. 2022. Banglahatebert: Bert for abusive language detection in bengali. In *Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis*, pages 8–15.

Md Saroar Jahan and Mourad Oussalah. 2023a. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546:126232.

Md Saroar Jahan and Mourad Oussalah. 2023b. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, page 126232.

Md. Rezaul Karim, Bharathi Raja Chakravarti, John P. McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In *7th IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA,2020)*. IEEE.

Md Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Md Azam Hossain, and Stefan Decker. 2021. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE.

Ashfia Jannat Keya, Md. Mohsin Kabir, Nusrat Jahan Shammey, M. F. Mridha, Md. Rashedul Islam, and Yutaka Watanobe. 2023. G-bert: An efficient method for identifying hate speech in bengali texts on social media. *IEEE Access*, 11:79697–79709.

Versha Kumari, Khuhed Memon, Burhan Aslam, and Bhawani Shankar Chowdhry. 2023. An effective approach for violence detection using deep learning and natural language processing. In *2023 7th International Multi-Topic ICT Conference (IMTIC)*, pages 1–8.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '19, page 173–182, New York, NY, USA. Association for Computing Machinery.

M. F. Mridha, Md. Anwar Hussen Wadud, Md. Abdul Hamid, Muhammad Mostafa Monowar, M. Abdullah-Al-Wadud, and Atif Alamri. 2021. L-boost: Identifying offensive texts from social media post in bengali. *IEEE Access*, 9:164681–164699.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61, Marseille, France. European Language Resources Association (ELRA).

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2022. Bd-shs: A benchmark dataset for learning to detect online bangla hate speech in different social contexts. *arXiv preprint arXiv:2206.00372*.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language*

Processing (BLP-2023), Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2022. M-bad: A multilabel dataset for detecting aggressive texts and their targets. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 75–85.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

A Appendices

A.1 Dataset Description

Here, we illustrate frequency plots for the train-set, dev-set, and test-set's three different classes as well as wordclouds that indicate various texts that incite violence for the three classes.

The frequency distribution displayed in Figure 4 shows that non-violent classes are more frequently reported than passive and direct forms of violence. This illustration makes it clear that the non-violent text class dominates, skewing the dataset in that direction. The labels 0, 1, and 2 stand for the three types of violence: Direct, Passive, and Non-Violence, respectively.

Figures 5, 6, and 7 show wordcloud where we can see words that are primarily responsible for inciting violence or Non-Violence in the text.

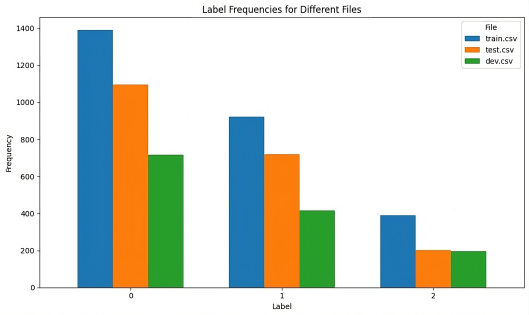


Figure 4: Label Frequency Distribution Across Different Dataset splits

Figure 5: Distinctive Language Patterns in Direct Violence Category

A.2 Experimental Results

Table 3 describes different model names denoted as numbers from 1 to 13 with their experimental approach.

Figure 6: Distinctive Language Patterns in Passive Violence Category

Figure 7: Distinctive Language Patterns in Non-Violence Category

Model Name	Approach
(1)	BanglaBert
(2)	BanglaBert using self learning on (1)
(3)	BanglaBert trained on subset of train + dev sets
(4)	BanglaBert trained on subset of train + dev sets
(5)	BanglaBert trained on subset of train + dev sets
(6)	BanglaBert trained on subset of train + dev sets
(7)	BanglaBert trained on subset of train + dev sets
(8)	BanglaBert trained on train + back_translated_train
(9)	BanglaBert trained on train + back_translated_train + back_translated_dev
(10)	BanglaBert trained on train + back_translated_train + predicted_test_on_best_model_during_competition
(11)	BanglaBert trained on train + back_translated_train + back_translated_dev + external_data + back_translated_external_data
(12)	BanglaBert trained on train + back_translated_train + back_translated_dev + external_data
(13)	BanglaBert trained on train + external_data

Table 3: Approaches of Different Models

BLP-2023 Task 1: Violence Inciting Text Detection (VITD)

Sourav Saha † ♣, **Jahedul Alam Junaed** † ♣, **Maryam Saleki** ♠, **Mohamed Rahouti** ♠, **Nabeel Mohammad** ♢, **Ruhul Amin** ♠
♣ Shahjalal University of Science and Technology, Bangladesh,
♢ North South University, Bangladesh, ♠Fordham University, USA
{sourav95,jahedul25}@student.sust.edu, *
{msaleki, mrahouti, mamin17}@fordham.edu,
nabeel.mohammed@northsouth.edu

Abstract

We present the comprehensive technical description of the outcome of the BLP shared task on Violence Inciting Text Detection (VITD). In recent years, social media has become a tool for groups of various religions and backgrounds to spread hatred, leading to physical violence with devastating consequences. To address this challenge, the VITD shared task was initiated, aiming to classify the level of violence incitement in various texts. The competition garnered significant interest with a total of 27 teams consisting of 88 participants successfully submitting their systems to the CodaLab leaderboard. During the post-workshop phase, we received 16 system papers on VITD from those participants. In this paper, we intend to discuss the VITD baseline performance, error analysis of the submitted models, and provide a comprehensive summary of the computational techniques applied by the participating teams.

Warning: The paper examples and the corresponding dataset contain violent inciting, derogatory, abusive, and racist comments. .

1 Introduction

Social media's growth over the past decade has reshaped the distribution of information to the broader public (Ferguson et al., 2014). However, it has also surfaced as a potential breeding ground for provoking violence among different groups, from religious to ethnic to gender-based distinctions. In fact, many of the violent incidents of the recent past era can directly or indirectly be attributed to incitement from social media (Mengü and Mengü, 2015). Such platforms can act as catalysts for the incitement of violence and the radicalization of individuals or groups (Recuero, 2015). Extremist ideologies and hate speech can spread rapidly, leading to real-world acts of violence. Acts of violence, triggered or fueled by content shared on social media, can inflict physical harm to individuals and communities with dire consequences that include physical injuries, destruction of properties, and even loss of human lives.

In the recent past, numerous studies were conducted into areas like hate speech detection (Warner and Hirschberg, 2012; Waseem and Hovy, 2016; Davidson et al., 2017; Karim et al., 2020; Romim et al., 2021), abusive content identification (Nobata et al., 2016), and misinformation detection (Shu et al., 2017; Hossain et al., 2020), aiming to understand and prevent harmful social media activities. There have been several workshops that contributed datasets and organized shared tasks on online harmful content detection in different languages (Bosco et al., 2018; Fersini et al., 2018; Zampieri et al., 2019; Basile et al., 2019). However, to the best of our knowledge, there exists no research work on the violence incitement in the Bengal Region (Bangladesh and West Bengal in India), the residence of more than 272 million[1,2] people of many diverse background. Therefore, this shared task seeks to bridge this gap by contributing a novel dataset on VITD for the development of new systems and methodologies with the objective to advance our collective understanding and capabilities in this crucial domain. In this paper, we discuss the following:

1. **Dataset Overview:** VITD task presents an intriguing challenge centered around the catego-

† Authors have equal contributions

[1] https://en.wikipedia.org/wiki/Bangladesh
[2] https://en.wikipedia.org/wiki/West_Bengal

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 255–265

Category	Definition	Example
Direct Violence	It refers to killing, rape, vandalism, deportation, desocialization, and resocialization.	দোকানে আগুন জ্বালিয়ে দেওয়া উচিৎ (The shop should be set on fire)
Passive Violence	It refers to use of derogatory language, abusive remarks, slang or any form of justification for violence.	সরকারের দোষ, সরকারের দালালি বন্ধ কর (Blame the government, stop the government brokering)
Non-Violence	It refers to discussions about social rights or general conversational topics that do not involve any form of violence.	সত্য প্রকাশে যমুনা টিভিকে ধন্যবাদ (Thanks to Jamuna TV for revealing the truth)

Table 1: The Table depicts examples of 3 different categories: Direct Violence (Red), Passive Violence (Yellow), & Non-Violence (Green). We also show the English translation using Google Translator service.

rization of textual content into three distinct and vital categories: Direct Violence, Passive Violence, and Non-Violence. We discuss how this dataset was prepared for the task.

2. **Baseline Performance:** We present the Macro-F1 score of VITD using both multilingual and Bangla BERT models.

3. **Team Statistics:** We discuss the participant's demographics in terms of gender and background.

4. **Error Analysis:** We present a detailed error analysis of each model submitted by the 27 teams.

5. **Comprehensive System Summary:** We also discuss the computational techniques used by different teams for the shared task.

2 Dataset Overview

The Vio-Lens dataset addresses the challenges of Violence Incitement Text Detection (VITD). It comprises data from YouTube comments related to violent content from Bangladesh and West Bengal. The dataset categorizes violence incitement into three classes: *Direct Violence, Passive Violence, and Non-Violence*. The description of each category along with relevant examples is provided in Table 1. The dataset features 6046 samples: 786 samples for direct violence, 2058 for passive violence, and the remaining 3202 for non-violence. This distribution illuminates a discernible class imbalance within the dataset, underscoring the need for careful consideration when designing and implementing classification algorithms or methodologies. For a detailed description of the Vio-Lens dataset, we refer the reader to the dataset paper Saha et al. (2023)[3].

[3]The dataset is publically available in https://github.com/blp-workshop/blp_task1/tree/main/dataset

3 Task Description and Evaluation

3.1 Task Definition

The shared task provides a classification task on three categories of violence, *Direct Violence, Passive Violence, and Non-Violence*, as discussed below:

- **Direct Violence:** This category encompasses explicit threats directed towards individuals or communities, including actions such as killing, rape, vandalism, deportation, desocialization (threats urging individuals or communities to abandon their religion, culture, or traditions), and resocialization (threats of forceful conversion). The detection of direct violence is crucial due to its potential to have severe consequences in the future.

- **Passive Violence:** This category includes instances characterized by the employment of derogatory language, derogative terms, or abusive remarks aimed at individuals or communities. Moreover, any attempt to rationalize or justify violence is classified within this category. Acknowledging these nuanced forms of hostility is key to understanding the breadth of online aggression.

- **Non-Violence:** Content within this category addresses non-violent matters, ranging from discussions about social rights to general conversations that are free from any violent implications. It's crucial to distinguish these benign exchanges from those that carry a more harmful intent.

3.2 Task Organization

We ran our competition on the CodaLab [4]. platform. There were two primary phases: (i) the Trail

[4]https://codalab.lisn.upsaclay.fr/competitions/14620

phase started on 16 July 2023 and ended on 15 August 2023, and (ii) the Test Phase, which began on 16 August 2023 and ended on 18 August 2023. We provided a training phase with the text and label, while the test phase contained only text data.

Models	F1 Score (Macro)
Majority Voting	23.350
MBERT	63.282
DistillBERT	59.863
XLM-RoBERTa (base)	66.062
BanglaBERT (base)	**71.073**

Table 2: The table shows the outcomes (macro-F1) classification using majority voting, MBERT, DistillBERT, XLM-RoBERTa, and BanglaBERT for the test set. All the experiments used the same dataset and parameters for a fair evaluation. We observe that BanglaBERT achieved the best macro F1 score.

3.3 Evaluation Metrics and Baselines

We evaluated all participating systems with Macro-F1 score. We are providing five baseline models (see Table 2) to benchmark a range of simple to complex systems for VITD. The simplest baseline model is the Majority Baseline, where all the categories are predicted as the majority Non-violence class. We provided four other fine-tuned Large Language models: XLM-RoBERTa (Liu et al., 2019), MBERT (Devlin et al., 2019), DistillBERT (Sanh et al., 2019), and BanglaBERT (Bhattacharjee et al., 2021). The first two are Multilingual models, while the third were monolingual ones. We ran all the models using the following parameters: learning rate 1e-5, train batch size 8, evaluation batch size 8, epochs 50, evaluation steps 250, and early stopping patience 5. Among the four baselines, the monolingual BanglaBERT provided the best Baseline with the highest macro F1 score of 78.791 on the dev set and 71.073 on the test phase.

3.4 Team Statistics

Our contest attracted 27 teams containing members from around the world. Among the contestants, 69 were male and 19 were female (Figure 1). The contest attracted participants including undergraduate students, graduate students, and professionals containing 13 undergraduates majority, 7 graduates majority, and 7 professionals majority teams.

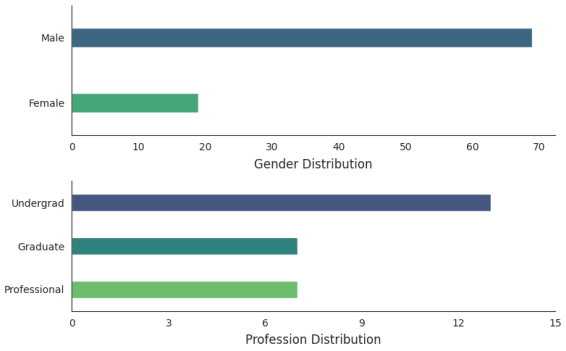


Figure 1: The figure shows gender distribution among the contestants and professions of each category of participants.

4 Participants Results

The baseline model with the best performance, BanglaBERT (Bhattacharjee et al., 2021), was outperformed by 16 teams. We display the ranking and best-performing models performance for each team in Table 3. We also report precision, recall, and F1 score for each category. Team DeepBlueAI achieved the highest overall performance, obtaining the Macro-F1 score of 76.044.

We observe that the highest precision, recall, and F1 score were reported for the **Non-Violence** category and worst on the *Direct Violence* category - indicating potential challenges in identifying explicit content. This may be due to the data imbalances in the dataset. Specifically, *Non-Violence* occupies 51.44%, 53.90%, and 54.37% of data on the train, validation, and test sets, respectively. On the other hand, *Direct Violence* is represented in only 14.41%, 14.74%, and 9.97% of the corresponding sets. In terms of team performance, a total of 20 teams surpassed the benchmark F1 score for the *Direct Violence*, and 17 teams achieved that for *Non-Violence*, while only 11 teams were found to cross the benchmark for *Passive Violence*. In particular, three teams: DeepBlueAI, Aambela, and NLP_CUET, exhibited high F1 scores across all three categories.

4.1 Error Analysis

A total of 27 teams participated in the VITD task. Among the 2,016 test samples, 506 unique samples were accurately predicted by all participating teams. There are a total of 72 samples that were incorrectly predicted by all the 27 teams. Additionally, there are a total of 214 unique samples that were incorrectly predicted by exactly one of the 27

Rank	Team	F1 score (macro)	Direct			Passive			Non-Violence		
			P	R	F1	P	R	F1	P	R	F1
1	DeepBlueAI	76.044	56.811	85.075	68.127	85.634	63.839	73.147	83.800	90.146	86.857
2	Aambela	76.041	59.286	82.587	69.023	84.404	63.978	72.785	82.872	90.055	86.314
3	NLP_CUET	74.587	61.004	78.607	68.696	73.745	71.488	72.599	83.868	81.113	82.468
4	Team Embeddings	74.418	52.761	85.572	65.275	81.122	66.342	72.992	84.755	85.219	84.986
5	Semantics Squad	74.413	57.664	78.607	66.526	81.607	63.561	71.462	82.149	88.595	85.250
6	NLP_BD_PATRIOTS	74.313	54.276	82.090	65.347	78.537	67.177	72.414	85.141	85.219	85.180
7	the_linguists	73.978	54.485	81.592	65.339	80.000	65.090	71.779	83.540	86.131	84.816
8	Panda	73.808	54.430	85.572	66.538	85.655	57.302	68.667	81.870	91.058	86.220
9	EmptyMind	73.797	52.266	86.070	65.038	82.130	63.282	71.485	83.554	86.223	84.868
10	Mavericks	73.699	55.932	82.090	66.532	82.863	61.196	70.400	80.840	87.774	84.164
11	LowResourceNLU	73.468	54.574	86.070	66.795	85.983	57.163	68.672	80.590	89.781	84.937
12	VacLM	72.656	50.286	87.562	63.884	80.536	62.726	70.524	83.183	83.942	83.560
13	LexicalMinds	72.551	51.562	82.090	63.340	83.080	60.779	70.201	81.453	86.953	84.113
14	Score_IsAll_You_Need	72.376	55.805	74.129	63.675	82.163	60.223	69.502	79.624	88.777	83.952
15	winging_it	71.207	45.316	89.055	60.067	83.622	60.362	70.113	83.212	83.668	83.439
16	Semantic_Savants	71.179	51.235	82.587	63.238	82.200	57.163	67.432	79.530	86.496	82.867
–	**Baseline**	**71.073**	**46.690**	**84.081**	**60.033**	**79.680**	**62.732**	**70.194**	**83.271**	**82.663**	**82.970**
17	BpHigh	70.978	53.741	78.607	63.838	80.639	56.189	66.230	78.624	87.591	82.866
18	SUST_Black Box	70.680	47.500	85.075	60.963	83.128	56.189	67.054	81.368	86.861	84.025
19	Team_Syrax	70.450	56.226	74.129	63.948	84.703	51.599	64.131	76.390	91.515	83.271
20	Blue	70.012	45.938	81.592	58.781	82.927	56.745	67.382	81.320	86.588	83.871
21	Team CentreBack	69.390	50.530	71.144	59.091	78.435	57.163	66.130	79.074	87.226	82.950
22	UFAL-ULD	69.009	47.447	78.607	59.176	75.215	60.779	67.231	80.399	80.839	80.619
23	BanglaNLP	68.110	53.650	73.134	61.895	78.602	51.599	62.301	74.646	86.496	80.135
24	KUET_NLP	60.332	36.557	77.114	49.600	75.204	38.387	50.829	76.327	85.310	80.569
25	Shibli_CL	38.427	37.727	41.294	39.430	68.421	01.808	03.523	58.469	94.799	72.329
26	Team Error Point	31.913	08.150	18.408	11.298	31.959	08.623	13.582	63.816	79.653	70.860
27	lixn	31.426	36.000	17.910	23.920	25.000	00.139	00.277	55.126	96.168	70.080

Table 3: The table shows the performance of each team along with the best-performing baseline model (BanglaBERT-base). It contains precision (P), recall (R), and F1 scores of individual categories, and finally a macro F1 score across all categories for final judgment.

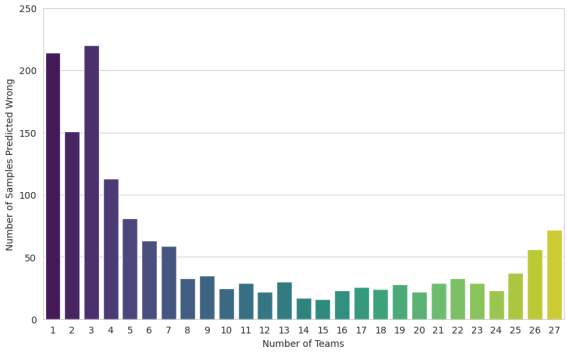


Figure 2: The bar plot shows the number of unique samples (Y-axis) that were predicted wrong by exactly N number of teams (X-axis) out of total 27 teams.

teams. A detailed visualization of these errors can be seen in Figure 2. In summary, a total of 1,510 samples were predicted incorrectly by one or more teams.

For the *Direct Violence* category, out of 201 test instances, 75.05% were predicted accurately by all the teams, while 6.80% were mistakenly identified as *Passive Violence*, and 18.15% were misclassified as *Non-Violence*. The *Passive Violence* test set comprises of 719 samples. Of those, 53.37% were correctly classified by all the teams, while 13.45% were categorized incorrectly as *Direct Violence*, and the rest samples were erroneously categorized as *Non-Violence*. For the *Non-Violence* category, which had 1,096 samples in the test set, an impressive 87.19% were correctly categorized by all the teams. Only 5.54% of those samples were incorrectly identified as *Direct Violence*, and the remaining 7.27% were misclassified as *Passive Violence* (see Figure 3).

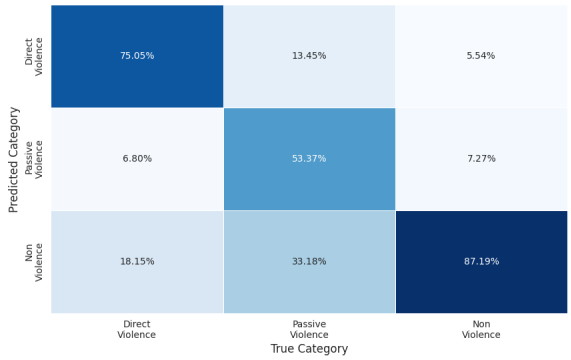


Figure 3: Confusion matrix illustrating category distribution among 27 teams.

We present a few examples from each of the categories, that were predicted wrong by all the teams (see Table 4). For the *Non-Violence* category, no

teams misclassified the same samples, indicating that the category may be easier to predict than the rest.

Example	Category
তোদের মত দাঙ্গাবাজ কুকুরদের বিচার আমি করব (I will judge riotous dogs like you)	Direct Violence
দেখা হবে ভাই মাঠে ময়দানে কোরআন নিয়ে উলটা পালটা কিছু বল্লে (See you in the field if you say something bad about Qur'an)	Direct Violence
ইসলামী আইন অনুযায়ী এই মেয়ের ঘরের মধ্যে আবদ্ধ থাকা উচিত,এ বাইরে কেনো। ইসলামে তো নারীদের যৌন দাসী হিসাবে ব্যবহার করে,এ ঘরের বাইরে গেলে তো ইসলামের অবমাননা করা হয়। (According to Islamic law, this girl should be confined inside the house, why she outside? In Islam, women are used as sex slaves, if she goes out of this house, Islam is insulted.)	Passive Violence
ধর্ম মানেই পাগলামি। সংঘাত, গালাগালি, মারামারি, খুন, ধর্ষন। (Religion means madness, conflict, abuse, fighting, murder, rape.)	Passive Violence

Table 4: This table presents some samples that all the teams predicted wrongfully. It is also to be noted that such wrong predictions were only observed either for *Direct or Passive Violence* categories.

5 Participants System Description

In this section, we present a comprehensive summary of each submitted system for the shared task.

AAmbela (Fahim, 2023) stood second in the competition with an overall Macro-F1 score of 76.040 for the test set. They propose an instruction-finetuned csebuetnlp-BanglaBERT (Bhattacharjee et al., 2022) with three classification heads. As BanglaBERT's vocabulary does not fully cover the tokens in the data, the team added them as special tokens that were learned during the training phase. They also observe the significance of emojis in the dataset, and removing them often leads to a minor result. On the other hand, converting emojis to text and normalizing the text leads to a better result. They experimented with various approaches such as traditional classifiers (SVM, Random Forest, XG-Boost) with Tf-IDF embeddings, Deep learning models (LSTM), and transformer-based architectures (mBERT-case, mDeBerta-v3 base (He et al., 2021a,b), XLM-Roberta base, SagorSarker-BanglaBERT (Sarker, 2020), BanglaBERT (Bhattacharjee et al., 2022). Finally, BanglaBERT trained on three epochs with a batch size of 16 came out on the top.

NLP_CUET (Hossain et al., 2023) achieved 3rd rank in this task with an overall Macro-F1 score of 74.587. They preprocessed data by removing unwanted characters and employed feature extraction methods like TF-IDF and Word2Vec. After investigating several machine learning, deep learning, and transformer-based models, they propose a hybrid method using GAN (Goodfellow et al., 2020) and Bangla-ELECTRA. Here, they considered both labeled data and unlabeled data for model training. The generator and discriminator are both multilayer perceptrons with a single hidden layer of 512 neurons. The generator input is a randomly generated vector of 100 dimensions, and it outputs a fake transformer embedding vector for a single token. The transformer-based model processed the input text, generating a contextualized embedding vector for the CLS token. These embedding vectors from the transformer and generator were then input into the discriminator. The output of the discriminator is extended to K+1 classes where k is the number of classes in this classification task, and the extra class is "REAL." In this approach, they focused on determining whether the embedding produced by the transformer-based architecture is real or fake. During the testing phase, they discarded the generator and used the BERT and discriminator model to classify the input data. They masked the prediction output for the 'REAL' class during testing.

Seamntic Squad (Dey et al., 2023) received the fifth rank with an overall Macro-F1 score of 74.413. They applied a preprocessing step of removing punctuation, lemmatization, and oversampling/undersampling. Afterward, they used different transformer-based models such as XLM-Roberta (base and large), BanglaBERT (Bhattacharjee et al., 2022) (base and large), and mBERT. Among the approaches, BanglaBERT-base achieved the highest result.

nlpBDpatriots (Raihan et al., 2023) received sixth in the competition with a macro f1 score of 74.313. They applied a rigorous data augmentation process, including translation and back-translation to make the dataset 7 times larger. They applied Statistical machine learning models (Linear Regression, Support Vector Machine), GPT-3.5, and various transformer-based approaches. Their two-step approach first classified violence and non-violence with MuRIL(Khanuja et al., 2021), and later XLM-RoBERTa to classify violence and non-violence on the larger dataset performed best.

the_linguists (Tariquzzaman et al., 2023) achieved 7th rank in this task with an overall

Macro-F1 score of 73.978. Firstly they collected 6.8 million data samples from Facebook and YouTube. Then they applied some preprocessing steps which resulted in a refined dataset containing 3.8 million samples. After that, they applied a semi-supervised methodology for training where the training of the informal FastText word embedding model was done by making use of the preprocessed unlabeled data. These embeddings were then integrated into the LR, SVM, LSTM, BiLSTM, and GRU models which were fine-tuned using the labeled data. And they got the best result from BiLSTM.

EmpytyMind (Das et al., 2023b) achieved 9th rank in this task with an overall Macro-F1 score of 73.797. They first preprocessed the dataset and then normalized the text. After that, they applied statistical machine learning-based approaches (Random Forest and Support Vector Machine, XG-Boost), deep learning-based approaches (one three bidirectional LSTM layers and the other four LSTM layers), and transformer-based approaches using a two-step hierarchical approach. In the hierarchical approach, they first classified the text into violence and non-violence categories, then further classified the violence category into direct violence and passive violence to combat the imbalance dataset, and it yielded the best performance.

Mavricks (Page et al., 2023) received 10th place in the competition with an overall Macro-F1 score of 73.699. They applied different transformer-based models (BanglaBERT, BanglishBERT, MuRIL, XLM-Roberta, and BengaliBERT) and ensembled them. They applied different ensembling methods among which hard voting came out on top.

LowResourceNLU (Veeramani et al., 2023) achieved 11th rank in this task with an overall Macro-F1 score of 73.468. Here, they aggregate three BERT-based language models. They configured the first model by incorporating two heads, one for Masked Language Modeling (MLM) and the other for classification, within the BanglaBERT-*large* framework. They used mBERT as their second model. As their third model, they used BanglaBERT-*base* by incorporating two classification heads. The first head focuses on the Bangla version of the XNLI dataset (Conneau et al., 2018). The second head is dedicated to the dataset. Initially, they extracted individual predictions from each model using the argmax function, selecting the class with the highest confidence score for each model. Then they applied another argmax operation, this time on the maximum logit values obtained from each model. Because of the incorporation of MLM in the first model, the F1 score is enhanced by a substantial margin. Similarly, the joint pretraining with XNLI significantly increased the performance of the third model. The combination of three models exhibits superior performance as compared to the use of a single model alone.

VacLM (Chatterjee et al., 2023) ranked 12th on the competition with an overall Macro-F1 score of 72.656. They introduced external information by incorporating data from Karim et al. (2020) and manually annotating them. They observed augmenting data from external sources in this way actually hampers the performance in the 3-way classification task but generally performs better for the violence and non-violence classification task.

Score_Is_All_You_Need (Ahmed et al., 2023) received 14th place in the competition with an overall Macro-F1 score of 72.376. They applied a two-step approach to first classify violence and Non-Violence. Afterward, from the violence category, they classify direct and passive violence using transformer-based approaches. They applied BanglaBERT, M-BERT, and XLM-RoBERTa using an exhaustive hyperparameter search to fit the model.

SUST_Black_Box (Shibu et al., 2023) ranked 18th in the competition with an overall Macro-F1 score of 70.680. They applied to incorporate data from similar sentiment and hate speech-related datasets for data augmentation. They used different transformer-based techniques such as SagorSarker-BanglaBERT(Sarker, 2020), M-BERT, and RoBERTa on the augmented dataset. Finally, they applied different ensembling methods to the augmented dataset.

Team_Syrax (Riyad et al., 2023) received 19th in the competition with an overall Macro-F1 score of 70.450. They applied traditional preprocessing steps such as emoji and punctuation removal. Then, they applied data augmentation from the Bengali hate speech detection dataset (BAD, BD-SHS). They applied different ensemble methods such as bagging and hard majority voting for the classification.

Team CentreBack (Alamgir and Haque, 2023)

ranked 21st in the competition with an overall F1 score of 69.390 in the test set. They applied several approaches using transformer-based architectures (BanglaBERT and XLM-Roberta) and a two-stage approach where they first classified violence and non-violence and then further classified the violence into direct and indirect violence. They also applied a few-shot approach with SBERT but it ultimately resulted in a poor performance. Among those approaches, BanglaBERT (20 epochs received the highest approach with the stage approach closely behind.

UFAL-ULD (Mukherjee et al., 2023) ranked 22nd in the competition with an overall Macro-F1 score of macro 69.009 for the test set. They applied different transformers-based models: XLM-Roberta-base, XLM-Roberta-large, BanglaBERT-Sagor, BanglaBERT-BUET and BanglaBERT-BUET-large. They used focal loss to handle the issue of class imbalance and applied simple data augmentation techniques like synonym replacement, insertion, deletion, swap, and shuffle.

BanglaNLP (Saha and Nanda, 2023) ranked 23rd in the competition with an overall Macro-F1 score of 68.110 for the test set. They used a general paraphrasing technique for data augmentation. In addition using general classification techniques such as logistic regression, SGD classifier, and multinomial naive bayes with ensembling techniques such as majority voting and stacking. They finally used BanglaBERT (Sarker) (Sarker, 2020) and Multilingual-E5-base as transformer-based model, with the later ultimately provided the best performance.

Team Error Point (Das et al., 2023a) ranked 26th with an overall Macro-F1 score of 31.913. They applied different traditional machine learning classifiers along with CNN and LSTM. Their combination of LSTM and CNN achieved the highest performance.

6 Discussion

6.1 Popular Architechture

The large majority of the participants (14 teams) employed transformer-based methods. They used mBERT, mDeBerta-v3 base, XLM-Roberta (*base* and *large*), SagorSarker-BanglaBERT, BanglaBERT (base and large), MuRIL, etc. Notably, variants of BanglaBERT consistently outperformed other models. Several submissions explored statistical machine learning methods leveraging FastText and Word2Vec for word-embeddings and subsequently used SVM, Logistic Regression, and XGBoost for classification. Another popular technique used by some teams is the two-steps approach to first classify the violence and non-violence and then subsequently classify them into *Direct and Passive Violence*. NLP_CUET used a GAN-based architecture. Please see Table 5 for details.

6.2 Popular Methods

Ensembling of different classifiers and transformers is the most prominent method used by the participants. Among the ensembling methods, hard voting gave the best results. Some teams used a two-step approach to classify the violence category and then the direct and passive violence from that category. Some teams tended to add more data to the dataset. They primarily adopted two approaches: One of the approaches included operations on the dataset such as insert, substitution, deletion, translation, and back-translation. The other approaches included datasets from similar datasets such as the Bangla Hate Dataset (Romim et al., 2021), and XNLI Dataset (Conneau et al., 2018), etc.

6.3 Insights

Generally, most of the successful process has been monolingual pre-trained language model modified with various task-specific process. Specially BanglaBERT (Bhattacharjee et al., 2022) has been the most impactful monolingual model. Emojis played a crucial role in the dataset build-up process and played a crucial role in the annotation. So, removing those has a negative impact on the prediction (Fahim, 2023). Also, statistical machine learning methods such as SVM, and XGBoost embedded after Fastext or Word2Vec don't capture the complex context of the dataset and fall short in the prediction. Deep Learning methods such as RNN, LSTM, and Bi-LSTM generally perform better than the statistical machine especially Das et al. (2023b) showed a significant score using a combination of lstm and bi-lstm with a two-step approach. Ultimately BanglaBERT (Bhattacharjee et al., 2022) was the most prominent for all the teams having a vast amount of pretrained knowledge of Bangla at its disposal.

Team Name	Embedding			Statistical Machine Learning							Deep Learning					Transformer Based Approach								GAN
	FastText	Word2Vec	Tf-Idf	LR	SVM	KNN	DT	RF	MNB	SGD	XGBoost	RNN	LSTM	Bi-LSTM	GRU	CNN	BanglaBERT (Sagor)	BanglaBERT (csebuetnlp)	MBERT	XLM-RoBERTa	MuRIL	Indic Bert	mDeBerta -v3 base	
AAmbela			✓		✓			✓			✓		✓				✓	✓	✓				✓	
NLP_CUET		✓	✓	✓	✓		✓	✓	✓	✓				✓		✓		✓						✓
Seamntic Squad																	✓	✓	✓	✓				
nlpBDpatriots				✓	✓														✓	✓	✓			
the_linguists	✓			✓	✓								✓	✓	✓									
EmpytyMind		✓			✓			✓			✓		✓	✓				✓						
Mavricks																		✓		✓	✓			
LowResourceNLU																		✓	✓					
VacLM																		✓			✓	✓		
Score_Is_All_You_Need																		✓	✓	✓				
SUST_Black_Box																	✓	✓	✓	✓				
Team_Syrax																		✓	✓	✓				
Team CentreBack																		✓		✓				
UFAL-ULD																	✓	✓	✓	✓				
BanglaNLP			✓	✓					✓	✓							✓		✓					
Team Error Point			✓	✓	✓		✓	✓	✓	✓			✓			✓								

Table 5: This table shows the most popular techniques used by different teams.

7 Limitations

Quantitative Limitations: The main limitation of the shared task arises from the dataset. First of all, the dataset is small in size, with only 4k data points for the training and validation sets, and around 2k data points for the test set. This often creates problems in terms of over-fitting on large models. Additionally, the dataset is highly imbalanced with only a minor fraction of the data for direct violence creating a challenge for class detection which is also reflective of the participant's results.

Qualititive Limitations: Emojis play a crucial role in sentence classification, so removing any emoji during preprocessing leads to a loss of context. The dataset consists of data from Bangladesh and West Bengal, comprising only in Bengali language. Therefore, the nature of violence-inciting text's nature may differ based on culture and language. Finally, the dataset requires a hectic process to annotate and validate thus expanding the correct data is much more difficult.

Procedural Limitations: The dataset is fully annotated by Bangladeshi residents, all undergraduate students, with an expert resolving the dispute. The annotation is done based on previous literature, personal observations, and a strict framework for annotators to rely on. Then relying on a majority vote and expert adjudication to produce is used to reach a gold standard label. Several previous studies reveal that annotator identity is a critical determinant of data annotation patterns (Sap et al., 2019; Larimore et al., 2021; Waseem, 2016) and so majority voting doesn't always capture the subjective nature of the annotation (Davani et al., 2022). Nonetheless, the definition of violence and its subcategories in taxonomy and how the authors' builders built their dataset and the annotators applied their best judgment are based on societal papers primarily from Galtung (1969, 1990), does not take FRS (Faith, Religion and Societal Impact) into account. Therefore, any dataset and corresponding systems will have the mentioned limitations. Thus, others with different cultural, societal, or religious backgrounds may disagree with some of the annotations.

8 Conclusion

In this paper, we have presented an overview of the shared task on the Violence Inciting Text Detection (VITD) dataset. The contest fostered submissions from 27 teams with 16 teams outperforming the highest baseline system BanglaBERT (Bhattacharjee et al., 2021), and 17 teams submitted the system paper. The system description and subsequent analysis and limitations discussion demonstrate the successful completion of the task.

The task has some vast scope for improvement. As mentioned in Saha et al. (2023), there is significant unlabelled data ready for further improvement of the systems to invoke larger systems without over-fitting the larger models. A potential scope for improvement is adding more data from huge unlabelled data. A future version of the task may be arranged with the challenge of more data from dif-

ferent sources, languages, and regions. Also, real-time violence detection models can be the next step of the task.

Ethical Considerations

We release the dataset and baseline classes and individual systems for specific classes containing violence-inciting texts. We also shared the participants' system descriptions. The malicious actors can use this information to train a generative model and use it for malicious purposes (Kirk et al., 2022). However, we believe that the risk is negligible to the huge potential of such systems in detecting violence-inciting text detection. The annotators were interviewed by the task organizers and they assured that they were given proper mental support and did not face any challenges at the time or after completing the annotation procedure.

References

Kawsar Ahmed, Md Osama, Md. Sirajul Islam, Md Taosiful Islam, Avishek Das, and Mohammed Moshiul Hoque. 2023. Score_isall_you_need at blp-2023 task 1: A hierarchical classification approach to detect violence inciting text using transformers. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Refaat Mohammad Alamgir and Amira Haque. 2023. Team centreback at blp-2023 task 1: Analyzing performance of different machine-learning based methods for detecting violence-inciting texts in bangla. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 54–63.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Samin, M Sohel Rahman, Anindya Iqbal, and Rifat Shahriyar. 2021. Banglabert: Combating embedding barrier for low-resource language understanding. *arXiv preprint arXiv:2101.00204*.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad Uddin, Kazi Mubasshir, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Lagnuage model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL*.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, Tesconi Maurizio, et al. 2018. Overview of the evalita 2018 hate speech detection task. In *Ceur workshop proceedings*, volume 2263, pages 1–9. CEUR.

Shilpa Chatterjee, P J Leo Evenss, and Pramit Bhattacharyya. 2023. Vaclm at blp-2023 task 1: Leveraging bert models for violence detection in bangla. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Rajesh Kumar Das, Jannatul Maowa, Moshfiqur Rahman Ajmain, Kabid Yeiad, Mirajul Islam, and Sharun Akter Khushbu. 2023a. Team error point at blp-2023 task 1: A comprehensive approach for violence inciting text detection using deep learning and traditional machine learning algorithm. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Udoy Das, Karnis Fatema, Md Ayon Mia, Mahshar Yahan, Md Sajidul Mowla, MD Fayez Ullah, Arpita Sarker, and Hasan Murad. 2023b. Emptymind at blp-2023 task 1: A transformer-based hierarchical-bert model for bangla violence-inciting text detection. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Krishno Dey, Prerona Tarannum, Md. Arid Hasan, and Francis Palma. 2023. Semantics squad at blp-2023 task 1: Violence inciting bangla text detection with fine-tuned transformer-based models. In *Proceedings of the 1st Workshop on Bangla Language*

Processing (BLP 2023), Singapore. Association for Computational Linguistics.

Md Fahim. 2023. Aambela at blp-2023 task 1: Focus on [unk] tokens: Analyzing violence inciting bangla text with adding dataset specific new word tokens. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Caleb Ferguson, Sally C Inglis, Phillip J Newton, Peter JS Cripps, Peter S Macdonald, and Patricia M Davidson. 2014. Social media: a tool to spread information: a case study analysis of twitter conversation at the cardiac society of australia & new zealand 61st annual scientific meeting 2013. *Collegian*, 21(2):89–93.

Elisabetta Fersini, Paolo Rosso, Maria Anzovino, et al. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. *Ibereval@ sepln*, 2150:214–228.

Johan Galtung. 1969. Violence, peace, and peace research. *Journal of peace research*, 6(3):167–191.

Johan Galtung. 1990. Cultural violence. *Journal of peace research*, 27(3):291–305.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Jawad Hossain, Hasan Mesbaul Ali Taher, Avishek Das, and Mohammed Moshiul Hoque. 2023. Nlp_cuet at blp-2023 task 1: Fine-grained categorization of violence inciting text using transformer-based approach. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. Banfakenews: A dataset for detecting fake news in bangla. *arXiv preprint arXiv:2004.08789*.

Md Rezaul Karim, Bharathi Raja Chakravarthi, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 390–399. IEEE.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Hannah Kirk, Abeba Birhane, Bertie Vidgen, and Leon Derczynski. 2022. Handling and presenting harmful text in nlp research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 497–510.

Savannah Larimore, Ian Kennedy, Breon Haskett, and Alina Arseniev-Koehler. 2021. Reconsidering annotator disagreement about racist language: Noise or signal? In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 81–90.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Murat Mengü and Seda Mengü. 2015. Violence and social media. *Athens Journal of Mass Media and Communications*, 1(3):211–227.

Sourabrata Mukherjee, Atul Kr Ojha, and Ondrej Dusek. 2023. Ufal-uld at blp-2023 task 1: Violence detection in bangla text. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Saurabh Page, Sudeep Mangalvedhekar, Kshitij Deshpande, Tanmay Chavan, and Sheetal S. Sonawane. 2023. Mavericks at blp-2023 task 1: Ensemble-based approach using language models for violence inciting text detection. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Md Nishat Raihan, Dhiman Goswami, Sadiya Sayara Chowdhury Puspo, and Marcos Zampieri. 2023. nlpbdpatriots at blp-2023 task 1: Two-step classification for violence inciting text detection in bangla - leveraging back-translation and multilinguality. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Raquel Recuero. 2015. Social media and symbolic violence. *Social media+ society*, 1(1):2056305115580332.

Omar Faruqe Riyad, Trina Chakraborty, and Abhishek Dey. 2023. Team_syrax at blp-2023 task 1: Data augmentation and ensemble based approach for violence inciting text detection in bangla. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam. 2021. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*, pages 457–468. Springer.

Saumajit Saha and Albert Aristotle Nanda. 2023. Banglanlp at blp-2023 task 1: Benchmarking different transformer models for violence inciting text detection in bangla. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2019. Social bias frames: Reasoning about social and power implications of language. *arXiv preprint arXiv:1911.03891*.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understading.

Hrithik Majumdar Shibu, Shrestha Datta, Zhalok Rahman, Shahrab Khan Sami, MD. SUMON MIAH, Raisa Fairooz, and Md Adith Mollah. 2023. Sust_black box at blp-2023 task 1: Detecting communal violence in texts: An exploration of mlm and weighted ensemble techniques. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Md. Tariquzzaman, Md Wasif Kader, Audwit Nafi Anam, Naimul Haque, Mohsinul Kabir, Hasan Mahmud, and Md Kamrul Hasan. 2023. the_linguists at blp-2023 task 1: A novel informal bangla fasttext embedding for violence inciting text detection. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023. Lowresourcenlu at blp-2023 task 1 2: Enhancing sentiment classification and violence incitement detection in bangla through aggregated language models. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

BanglaNLP at BLP-2023 Task 2: Benchmarking different Transformer Models for Sentiment Analysis of Bangla Social Media Posts

Saumajit Saha
saha.saumajit@gmail.com

Albert Nanda
albert.nanda@gmail.com

Abstract

Bangla is the 7th most widely spoken language globally, with a staggering 234 million native speakers primarily hailing from India and Bangladesh. This morphologically rich language boasts a rich literary tradition, encompassing diverse dialects and language-specific challenges. Despite its linguistic richness and history, Bangla remains categorized as a low-resource language within the natural language processing (NLP) and speech community. This paper presents our submission to Task 2 (Sentiment Analysis of Bangla Social Media Posts) of the BLP Workshop. We experimented with various Transformer-based architectures to solve this task. Our quantitative results show that transfer learning helps in better learning of the models in this low-resource language scenario. This becomes evident when we further finetuned a model that had already been finetuned on Twitter data for sentiment analysis task and that finetuned model performed the best among all other models. We also performed a detailed error analysis, finding some instances where ground truth labels need to be looked at. We obtained a micro-F1 of 67.02% on the test set and our performance in this shared task is ranked at 21 in the leaderboard.

1 Introduction

Sentiment analysis is the task of determining the attitude or opinion expressed in a piece of text. Typically sentiment categories are of three types: Positive, Negative, and Neutral. In today's increasingly interconnected world, where digital communication abounds, sentiment analysis has emerged as a vital component of natural language processing (NLP) and computational linguistics. It enables us to gauge public sentiment on diverse topics, monitor social media trends, and make data-driven decisions in various domains, including marketing, customer service, and politics. Social media provides an interesting platform to study sentiment analysis. People have diverse opinions regarding any topic and they express them accordingly. Mining sentiments from them often become very critical due to the trending social media lingo.

The use of slang, informal language, and emojis in social media posts can further complicate the task of sentiment analysis. The scarcity of resources and research initiatives dedicated to Bangla sentiment analysis can be attributed to several factors. Firstly, Bangla is considered a low-resource language within the NLP and speech community, primarily due to limited and scattered research efforts undertaken by individual researchers or small teams. Secondly, the development of robust deep learning models pre-trained on monolingual bengali data is not that widely available like we have numerous models pre-trained on English data.

2 Related Works

Early work on sentiment analysis in Bangla relied on lexicon-based and rule-based methods like in (Chowdhury and Chowdhury, 2014). Lexicon-based methods use a dictionary of sentiment words to identify the sentiment of a text. Rule-based methods use a set of rules to identify sentiment words and phrases. However, with the advancement of deep learning models, these approaches were outperformed by them because they are more capable of understanding the contextual meaning of the sentence and they do not require handcrafted rules or a set of lexicons to identify the sentiment present in text segment.

Bhowmick and Jana (2021) performed sentiment analysis using Bert and XLM-Roberta on three datasets - Prothom Alo (Islam et al., 2020), YouTube-B (Sazzed, 2020) and Book-B (Hossain et al., 2021). Kabir et al. (2023) introduced a large-scale Bangla dataset for sentiment analysis from book reviews. Islam et al. (2023) introduced a multi-domain Bangla sentiment analysis dataset

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 266–272

across 30 different domains.

3 System Description

This section describes our system which is developed to classify sentiment present in Bangla social media posts. This section starts with the shared task description, followed by the description of the dataset released by the shared task organizers, then our proposed architecture which produced our team's standing on the leaderboard, and finally the results achieved and observations made. All the codes and datasets used for performing the experiments are available in https://github.com/Saumajit/BanglaNLP/tree/main/Task_2.

3.1 Shared Task Description

The objective of this shared task[1] (Hasan et al., 2023a) is to identify the sentiment associated with a given text segment. Given a Bangla text segment, the output produced by the system should belong to one of the 3 classes - *positive*, *negative*, and *neutral*.

3.2 Dataset Description

Table 1 shows a sample sentence from the given dataset for each of the 3 sentiment categories. The dataset under consideration in this shared task combines data from two distinct sources: MUBASE (Hasan et al., 2023b) and SentNob (Islam et al., 2021). The SentNob dataset consists of public comments from various social media platforms related to news and video content. These comments are curated from 13 diverse domains such as politics, education, and agriculture. On the other hand, the MUBASE dataset is a large collection of multi-platform dataset that includes manually annotated Tweets and Facebook posts, each labeled with their respective sentiment polarity. Table 2 highlights the count of positive, negative, and neutral sentences across train and development splits of the dataset respectively.

We find that almost 80% of the sentences across train and development sets have less than 20 words for each of the three sentiment categories. We illustrate this analysis in the appendix.

3.3 Our Approaches

We have performed several experiments by using different transformer (Vaswani et al., 2017) models as well as several traditional machine learning algorithms. We report the promising approaches here and the rest of our approaches and their results are presented in the Appendix.

3.3.1 Proposed Approach : Finetuning twitter-xlm-roberta-base-sentiment[2]

Barbieri et al. (2021) pretrained xlm-roberta-base[3] model from scratch on the tweet data. The tweets were from diverse languages as they did not want to focus on any specific language. Then they finetuned their pre-trained language model on a multilingual Sentiment Analysis dataset using adapter technique (Pfeiffer et al., 2020).

We use their finetuned model checkpoint as released in Hugging Face and further finetune it on our dataset. Since this model is already well aware of multilingual linguistic features, it performs the best on this shared task compared to all the other models that we have experimented with. Pre-existing knowledge of multilingual sentiment analysis might have helped the model in better transfer learning on our data during finetuning.

We used a learning rate of $5e-5$, AdamW (Loshchilov and Hutter, 2017) as optimizer and a batch size of 32. We used V100 GPU for finetuning. With EarlyStopping, our best finetuned model was obtained after 2 epochs and the time taken for finetuning it on our dataset was approximately 1 hour. For finetuning the transformer-based models for SequenceClassification, we had used AutoModelForSequenceClassification class from Hugging Face throughout this paper, unless otherwise specified. During the development phase of this shared task, this finetuned model gave the best performance on the *dev_test* data split. We therefore used this model for inference on the test set released by the shared task organizers.

3.3.2 Other Approaches

Two other interesting models and approaches, which lie just behind our proposed approach in terms of performance, are discussed here.

1. **Finetuning BanglaBERT**[4] Sarker (2020) proposed BanglaBERT by pretraining base ELECTRA (Clark et al., 2020) model with the Replaced Token Detection objective. Their pretraining data consists of web-crawled data

[1] https://github.com/blp-workshop/blp_task2

[2] https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base-sentiment

[3] https://huggingface.co/xlm-roberta-base

[4] https://huggingface.co/sagorsarker/bangla-bert-base

Sentence	Sentiment
টানা দুই হারের পর জয়ের স্বাদ পেল ইউভেন্তুস।	Positive
করোনায় আক্রান্ত হয়ে আরো ১ জনের মৃত্যু	Negative
চিন্তা করেন যারা বক্তব্যে দিচ্ছে তাদের কণ্ঠ ও ছবি দেখাতে সাহস ও পায় না	Neutral

Table 1: Sample data for each of the three sentiment categories.

	Train	Dev
Positive	12364	1388
Negative	15767	1753
Neutral	7135	793

Table 2: Dataset statistics

and post-filtering to include only bengali data from crawled webpages. We finetuned BanglaBERT on this shared task's dataset using the learning rate of $5e-5$, AdamW as an optimizer, batch size of 32, and the number of epochs as 10.

2. **P-Tuning XLM-Roberta-Large**[5] Models having billions of parameters often suffer from poor transferability. Yue et al. (2020) discussed that these models are too large to memorize the finetuning samples. Liu et al. (2021) introduced P-tuning, a technique which does not change the pre-trained models' parameters but evoke the stored knowledge by finding a better continuous prompt. In finetuning, all the models' parameters get updated. However in P-tuning, the parameters corresponding to continuous prompt get updated but these parameters are of several magnitude orders smaller than the pre-trained models' parameters. The advantage of P-tuning over discrete prompts is that P-tuning helps us to find better continuous prompts beyond the original vocabulary of the pre-trained language model. We used P-tuning on XLM-Roberta-Large for the sentiment classification task. We used the learning rate of $1e-4$, the number of epochs set to 15, and the batch size set to 8. This approach trained only 42.86% of the model parameters thereby saving compute and time without impacting model performance to a great extent.

[5]https://huggingface.co/xlm-roberta-large

3.4 Results and Findings

This subsection highlights the results we had obtained during the development phase of this shared task, the metric we used for evaluating model performance, results, and error analysis on the test set.

Approach	Model	Micro-F1
FT	twitter-xlm-roberta-base-sentiment	**0.68**
FT	BanglaBERT	0.65
PT	xlm-roberta-large	0.63

Table 3: Performance of different models on the development set. FT : Finetuning, PT : P-Tuning.

3.4.1 Evaluation Metric

The evaluation metric for this shared task is micro F1. Micro F1 calculates metrics globally by counting the total number of true positives, false negatives, and false positives.

3.4.2 Performance on Development and Test Set

Table 3 highlights the performance of our approaches on the given dataset during the development phase. We see that the *twitter-xlm-roberta-base-sentiment* model performed the best in terms of evaluation metrics. This might have happened due to transfer learning (Farahani et al., 2021) which aims to benefit pre-trained models that need to be further trained on low-resource languages. We also finetuned *BanglaBERT*, a monolingual model, to evaluate how it performs in comparison to the other models. We see that there is a gap in its performance and that may be attributed to the monolingual nature of a model trained on a low-resource language. Finally our P-tuning approach on *xlm-roberta-large* gave a competitive performance with the above models with less number of trainable parameters. On the test set shared with us by the organizers, we obtained a micro F1

Sentence	Ground Truth	Prediction
সিরিয়ায় অবস্থান করা বিদেশি বাহিনীর সমালোচনা করেছেন পুতিন ।	Positive	Negative
আজ আইন এই রকম বলেই দিন তো দিন বেড়ে যাচ্ছে দর্শন । তাই এই বিষয় দল , বল খোঁজবেন না কঠিন শাস্তি দিবেন ।	Positive	Negative
ভারতীয় ব্যাটসম্যানদের দাঁড়াতেই দেয়নি ইংলিশ বোলাররা ।	Positive	Negative
মহাকাশে কি এলিয়েন আছে ?	Positive	Neutral
ভাগ্যরেখা অনুযায়ী আপনার আজকের দিনটি কেমন কাটতে পারে ?	Positive	Neutral
খুব বিরক্তিকর একটা জিনিস । খুলতে গিয়ে টাকা ছিড়ে যায় ।	Neutral	Negative
আশা করেছিল ড্রন থেকে ফুলের তুরা , বুকে , পরবে ভুলে বোমা পারে গেছে	Neutral	Negative
বিষয়টা বেশ হাস্যরস সৃষ্টি করে	Neutral	Positive
ছবিটা চমৎকার ভাবে এডিট করা হইছে	Neutral	Positive
এবার হয়তো আপনাদের তালিবদের দুঃসহ বেদনাটা একটু কমবে আশা করি	Negative	Positive
সহজে বলতে গেলে শাক দিয়ে মাছ ঢাকা হচ্ছে এই আরকি	Negative	Positive
হিসাব টা কিভাবে বের করলেন ব্রো	Negative	Neutral
শুধুই জাতীয় বিশ্ববিদ্যালয় ফোকাস করছেন কেনো পাবলিক বিশ্ববিদ্যালয়ের বেকারেরর সংখ্যা তুলে ধরুন	Negative	Neutral

Table 4: Samples where model predictions look good but ground truths look incorrect.

of 67.02% using our finetuned *twitter-xlm-roberta-sentiment* model. We therefore observe that the model performance slightly ($\sim$ 1%) drops on the evaluation phase test set compared to the development phase. This helps us to understand that our finetuned model also generalized well to unseen data and thus is fairly stable in nature.

3.4.3 Error Analysis on Test set

While visually analyzing the model predictions, we find that there are several instances where our model had predicted the correct sentiment class while the corresponding ground truth labels do not seem to be correct. Table 4 shows some of the samples where our model's predictions actually look correct but ground truth annotations look incorrect. Inspite of incorrect ground truths, the model through its prior knowledge (both from transfer learning as well as finetuning on our data) was able to correctly predict the output which looks far more realistic. This stable nature of the model will help to improve data quality and get tagged data by using it to create weak sentiment labels on unseen data and then get them verified by a human-in-the-loop (Wu et al., 2021) setting.

Figure 1 denotes the confusion matrix we got by our model's predictions on the test set in the evaluation phase. We found that 67.78% positive sentences, 78% negative sentences, and 37% neutral sentences have been predicted correctly. We also found that neutral sentences got misclassified the most into positive and negative classes. Intuitively, this could happen due to the availability of less number of neutral samples in the training data in comparison to positive and negative samples. From the dataset distribution in Table 2, we observe that the higher the number of samples seen during training, the less the number of samples getting incorrectly predicted by the model.

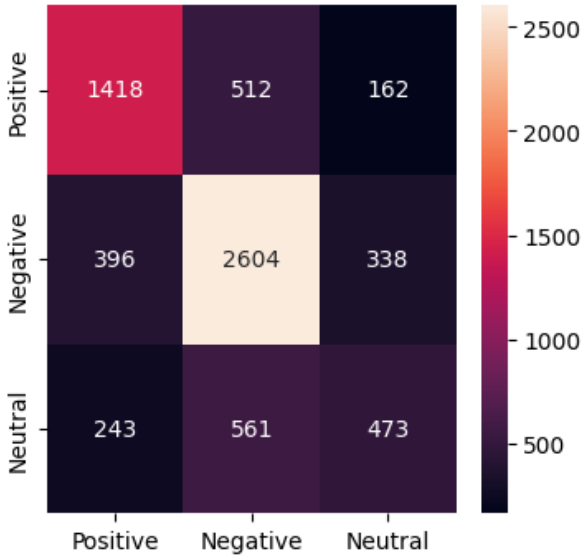


Figure 1: Confusion Matrix obtained for the test set.

4 Conclusion

We have provided an overview of how some of the promising approaches using transformer-based models perform with Bengali text data. We have

also pointed out a few flaws in the annotation quality of the data, which if corrected, may lead to better performance of the models. We find that a transfer learning-based approach with a multilingual model works best in such a low-resource scenario when there are not too many models available that are pre-trained on a huge corpus of monolingual data. An interesting future research direction seems to be the application of recently released Large Language Models (LLMs) in the NLP space and see how they perform with a low-resource language like Bengali.

5 Limitations

The experiments performed, models chosen, and results that have been discussed here are purely based on a low-resource language like Bangla and the particular dataset shared for use in the Shared Task. All experiments are mostly run in v100, T4 GPU, and rarely in A100 using Google Colab. Recently released Large Language Models and ChatGPT are not used here due to compute and pricing constraints.

References

Francesco Barbieri, Luis Espinosa Anke, and José Camacho-Collados. 2021. XLM-T: A multilingual language model toolkit for twitter. *CoRR*, abs/2104.12250.

Anirban Bhowmick and Abhik Jana. 2021. Sentiment analysis for Bengali using transformer based models. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 481–486, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Shaika Chowdhury and Wasifa Chowdhury. 2014. Performing sentiment analysis in bangla microblog posts. In *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, pages 1–6. IEEE.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Abolfazl Farahani, Behrouz Pourshojae, Khaled Rasheed, and Hamid R. Arabnia. 2021. A concise review of transfer learning. *CoRR*, abs/2104.02144.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2021. Sentiment polarity detection on bengali book reviews using multinomial naïve bayes. In *Progress in Advanced Computing and Intelligent Engineering*, pages 281–292, Singapore. Springer Singapore.

Khondoker Ittehadul Islam, Md. Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Md. Ekramul Islam, Labib Chowdhury, Faisal Ahamed Khan, Shazzad Hossain, Md Sourave Hossain, Mohammad Mamun Or Rashid, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023. Sentigold: A large bangla gold standard multi-domain sentiment analysis dataset and its evaluation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 4207–4218, New York, NY, USA. Association for Computing Machinery.

Mohsinul Kabir, Obayed Bin Mahfuz, Syed Rifat Raiyan, Hasan Mahmud, and Md Kamrul Hasan. 2023. Banglabook: A large-scale bangla dataset for sentiment analysis from book reviews.

Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2005. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence*, pages 488–499, Berlin, Heidelberg. Springer Berlin Heidelberg.

L. Lam and S.Y. Suen. 1997. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27(5):553–568.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*, abs/2103.10385.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Salim Sazzed. 2020. Cross-lingual sentiment classification in low-resource Bengali language. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 50–60, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2021. A survey of human-in-the-loop for machine learning. *CoRR*, abs/2108.00941.

Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. 2020. Interventional few-shot learning. *CoRR*, abs/2009.13000.

6 Appendices

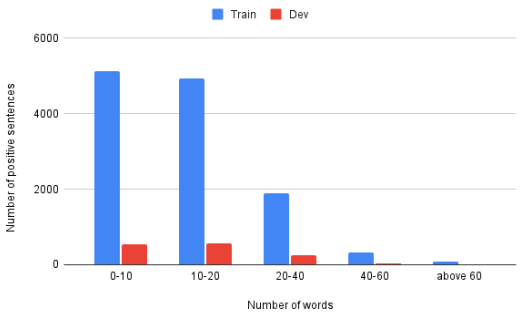
In this section, we report the word count analysis per sentence across the train and development dataset. We also report some of the additional experiments we had done, which did not give satisfactory outcomes.

6.1 Word count distribution

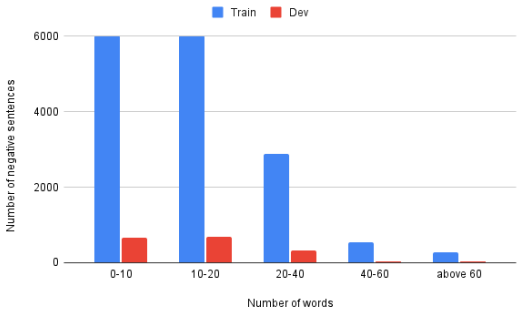
Figure 2 analyzes the number of sentences which lie in the different word count intervals. For all the categories of sentiment, we find that the majority of the data samples have less than 20 words across both the train and development splits of the dataset.

6.2 Other experiments

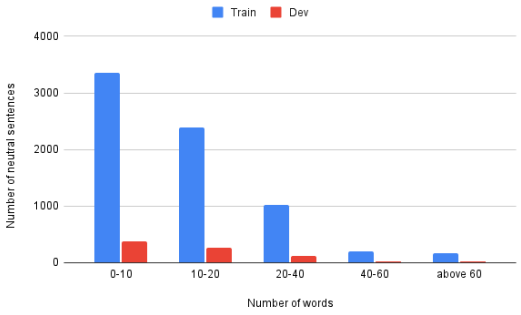
Before moving to using deep learning models, we had also initially tried out several traditional machine learning algorithms like *Logistic Regression*, *Multinomial Naive Bayes* (Kibriya et al., 2005), *SGD classifier*, *Majority Voting* (Lam and Suen, 1997) of previous three classifiers and *Stacking* with XGBoost (Chen and Guestrin, 2016) as the final classifier. We had used TF-IDF (Ramos, 2003) vectorization to convert words into a vectorized representation before passing them into these classification algorithms for sentiment classification.



(a) Positive



(b) Negative



(c) Neutral

Figure 2: Analysis of number of sentences to the number of words present in each sentence across train and development dataset.

Approach	Model	Micro-F1
Traditional ML	Logistic Regression	0.55
	Multinomial Naive Bayes	0.56
	SGD classifier	0.47
	Majority Voting of above 3	0.55
	Stacking	0.54
Finetuning	Bert-base-multilingual	0.64
Finetuning	Flan-t5-base	0.47

Table 5: Additional Experiments

Table 5 presents the findings achieved in this task with the mentioned algorithms during the development phase.

We also used *bert-base-multilingual*[6] model to check how it performs on our task. Since it was pretrained on top of 104 languages, our intuition behind trying out this model is that the linguistic features learned by the model across different languages may help in performing our task better. From Table 5, we observe that finetuning this model gives a competitive performance.

We also tried to instruction-finetune flan-t5-base[7] model on our task. The result in Table 5 does not look promising as we have just tried to experiment with it using only a fixed setting of hyperparameters. We do not do any hyperparameter optimization here due to compute constraints. We use a learning rate of $3e-4$, batch size of 32, and number of epochs set to 5. We prepend the prompt (পাঠ্য অংশের অনুভূতি শ্রেণীবদ্ধ করুন:) to the input text to finetune the flan-t5-base model. This particular approach generates the ground-truth class label instead of classifying it into one of the pre-defined class labels which happens in a multi-class classification setting.

[6] https://huggingface.co/bert-base-multilingual-cased

[7] https://huggingface.co/google/flan-t5-base

Knowdee at BLP-2023 Task 2: Improving Bangla Sentiment Analysis Using Ensembled Models with Pseudo-Labeling

Xiaoyi Liu, Teng Mao, Shuangtao Yang, Bo Fu
Lenovo Knowdee (Beijing) Intelligent Technology Co., Ltd., Beijing, China
{liuxy, maoteng, yangst, fubo}@knowdee.com

Abstract

This paper outlines our submission to the Sentiment Analysis Shared Task at the Bangla Language Processing (BLP) Workshop at EMNLP 2023 (Hasan et al., 2023a). The objective of this task is to detect sentiment in each text by classifying it as Positive, Negative, or Neutral. This shared task is based on the MUltiplatform BAngla SEntiment (MUBASE) (Hasan et al., 2023b) and SentNob (Islam et al., 2021) dataset, which consists of public comments from various social media platforms. Our proposed method for this task is based on the pretrained Bangla language model BanglaBERT (Bhattacharjee et al., 2022). We trained an ensemble of BanglaBERT on the original dataset and used it to generate pseudo-labels for data augmentation. This expanded dataset was then used to train our final models. During the evaluation phase, 30 teams submitted their systems, and our system achieved the second highest performance with F1 score of 0.7267. The source code of the proposed approach is available at https://github.com/KnowdeeAI/blp_task2_knowdee.git.

1 Introduction

While English dominates as the most resource-rich language in the Natural Language Processing (NLP) community, Bangla which ranked as the 6th most spoken language still faces resource scarcity. Despite three decades of BNLP research, progress has lagged mainly due to scarce resources and associated challenges (Alam et al., 2021).

The objective of the Sentiment Analysis Shared Task is to detect sentiment in each text by classifying it as Positive, Negative, or Neutral. This task utilizes a combined dataset of MUltiplatform BAngla SEntiment (MUBASE) (Hasan et al., 2023b) and SentNob (Islam et al., 2021). MUBASE contains manually annotated social media posts from Twitter and Facebook labeled with sentiment polarity. SentNob consists of social media comments from multiple platforms related to news and videos covering 13 different domains (Islam et al., 2021).

Bangla is a language with rich morphology, many dialects, and unique linguistic nuances. (Alam et al., 2021). Additionally, the dataset used consists of noisy social media comments with a mix of dialects and grammatical errors (Islam et al., 2021). The combination of Bangla's inherent linguistic challenges and the informal, non-standard nature of the dataset creates difficulties for sentiment analysis.

In this work, we present our solution and experimental attempts at the sentiment analysis shared task in Section 2, Our main approach involves an ensembling technique with pseudo-labeling to maximize performance given the limited training data. Results and analysis are followed in Section 3. Finally, Section 4 concludes with a summary of results and an outlook on future directions to advance low-resource natural language processing tasks for Bangla and other languages.

2 System Description

We discuss our proposed solution for the shared task from Section 2.1 to Section n three steps: 1) finetuning an ensemble of models on the provided supervised training data, 2) Using the ensemble models from step 1 to generate pseudo-labels for unlabeled data, 3) Training a new ensemble on the combination of the original training data and pseudo-labeled dataset, to make final predictions.

Additionally, we discuss other pre-trained models we experimented using the proposed solution and another attempted solution in Section 2.4. The experiments result is discussed in Section 3.2.

2.1 Supervised Finetuning

The first step of our solution was to finetune pretrained language models on the downstream sentiment classification task using the provided training data. We split the training data equally into 10 folds.

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 273–278

And we finetuned the same base language model 10 times, using a different fold for validation and the remaining 9 folds for training each time. This generated an ensemble of 10 finetuned classifiers, each trained on a unique subset of the data.

Additionally, we incorporated the Fast Gradient Method (FGM) as an adversarial training technique to improve model robustness and prevent overfitting during finetuning. FGM works by adding small perturbations to the input embeddings based on gradient of the loss. The adversarial noise injections force the model to learn more generalizable representations. The basis of our solution is BanglaBERT (Bhattacharjee et al., 2022), which is a BERT-based language model pre-trained in Bangla using Google Research's ELECTRA (Clark et al., 2020). ELECTRA is a method for efficient self-supervised language representation learning, which can be used to pre-train transformer networks. Specifically, ELECTRA models are trained with the Replaced Token Detection (RTD) objective – to identify which tokens in an input sequence have been replaced by plausible alternatives generated by a small neural network.

2.2 Data Augmentation

After finetuning the 10 models, we utilized them to generate pseudo-labels for unlabeled data as a mean of dataset expansion. The models made predictions on the provided test set, along with confidence scores for each of the 3 sentiment labels per sample.

For each test sample, we summed the confidence scores predicted across the 10 models separately for each sentiment label. If the highest accumulated confidence score exceeded our predefined threshold, we added that sample to the pseudo-labeled dataset with its highest scored label. The higher the threshold is set, the fewer samples are selected for the pseudo-labeled dataset, as only those with very high confidence in the majority of models will pass the cutoff. To obtain a pseudo-labeled dataset with more reliable labels, we set a stringent threshold of 9 out of ten. This ensured that only samples for which the majority of models were highly certain about the sentiment label (the average of the 10 models' confidence scores on the selected label was 0.9 or higher) would make it into the pseudo-labeled set. Samples where the maximum confidence score fell below the threshold were discarded and not added to the pseudo-labeled data.

2.3 Generating final predictions

After creating the pseudo-labeled dataset, we augmented each model's original training set with this pseudo-labeled dataset. Using this expanded dataset, we repeated the finetuning process described in Section 2.1 to train 10 new finetuned models. Each of these 10 models independently predicted sentiment labels for the test set.

To generate the final predictions, we summed the confidence scores per label across the ensemble for each test sample, similar to our pseudo-labeling approach. However, rather than applying a threshold, we directly assigned the label with the maximum summed confidence score as the final prediction.

The ensemble of 10 models helped mitigate noise and overfitting. Combining models exposed to slightly different data distributions reduced individual idiosyncrasies and enabled more robust predictions. The models were less likely to jointly make incorrect high-confidence predictions on ambiguous samples, improving generalization though the training sets were predominantly shared.

2.4 Attempted Models and solutions

Besides BanglaBERT mentioned in Section 2.1, we also experimented other language models with the same training methodology: 1) MuRIL (Khanuja et al., 2021), a BERT model pre-trained on a large corpus of 17 Indian languages; 2) XLM-RoBERTa (Conneau et al., 2019)), a multilingual version of RoBERTa and is pre-trained on data containing 100 languages; 3) mT5 (Xue et al., 2021), a multilingual T5 pre-trained on dataset covering 101 languages.

In addition to utilizing the original dataset, our study incorporated a reformatting approach to conduct in-context learning with the mT5 and BanglaBERT. This method involved a restructuring of the dataset, imbuing each sample with contextual information. For each case, we selected 3 similar samples and their labels from the training set, one for each sentiment label (positive, negative, neutral). The reconfigured dataset was used to finetune mT5 on a text generation task to predict the sentiment label. For BanglaBERT, we finetuned on sequence classification task. It is worth noting that, aside from the variance in the format of training and test data, all other procedural aspects pertaining to the generation of predictions remain consistent with descriptions in Sections 2.1 and 2.3.

3 Experiments and Results

This section presents the official results of our submitted solution for the sentiment analysis shared task. Additionally, we conducted post-evaluation experiments using the gold standard labels to compare the performance of our submitted system against alternative approaches on the test set.

3.1 Experimental Set-up

Our submitted solutions used banglabert_large[1], but we have experimented with various models of different sizes - banglabert[2], muril-large-cased[3], muril-base-cased[4], xlm-roberta-base[5], and mt5-large[6].

We used different hyperparameter configurations for each data format. For the original format, models were trained for 15 epochs with a batch size of 64, max sequence length of 128 tokens, and a learning rate of 2e-05. For the in-context learning format, models were trained also for 15 epochs, but we decreased the batch size to 16, and increased the max sequence length to 384 tokens and the learning rate to 5e-05 in order to accommodate longer contexts.

We also conducted post evaluation experiments on comparing one round, two rounds, and no rounds of pseudo-labeling on different models. All other hyperparameters were held constant across experiments. For both evaluations on dev and test set, we used the official scorer scripts to score the output.

3.2 Results and Analysis

The official results of the top five ranked solutions and baseline solutions for the sentiment analysis shared task are shown in Table 1. Our submitted system achieved an F1-micro score of 0.7267, which ranked 2nd out of 30 participating systems.

Table 2 shows all our experiment results on dev and test set. Our initial experiments (no pseudo-labeling) with various pre-trained language models showed noticeable differences in performance. Across models, we observed up to 3% variance in F1 scores on the dev set. Banglabert achieved the highest dev F1 at 0.7345 (Exp. 4), while multilingual model xlm-roberta-base performed worst at 0.7076 (Exp. 7). However, on the test set muril-large-cased obtained the best F1 of 0.7307. The poorer performance of xlm-roberta-base compared to BanglaBERT and MuRIL models indicates the importance of language-specific pretraining. While xlm-roberta-base was pretrained on multiple languages, BanglaBERT focused specifically on Bangla pretraining and MuRIL on Indian languages including Bangla. The results show that pretraining on closer languages leads to better transferability for Bangla sentiment analysis.

Ranking	Username	F1-Micro
1	MoFa_Aambela	0.7310
2	Our System	0.7267
3	amlan107	0.7179
4	Hari_vm	0.7172
5	PreronaTarannum	0.7164
-	n-gram Baseline	0.5514
25	Baseline (Majority)	0.4977
29	Baseline (Random)	0.3356

Table 1: Official result of the top five ranked solutions and official baseline solutions

To compare training methods, we finetuned mt5-large to generate labels (Exp 8), achieving F1 scores of 0.7095 (dev) and 0.7070 (test). For in-context learning, we constructed similar examples as context to provide more information. With mt5-large (Exp 9), in-context learning improved over direct generation (Exp 8), with F1 of 0.7189 (dev) and 0.7082 (test). However, with banglabert_large (Exp 11), in-context learning decreased performance versus direct classification (Exp 3), scoring 0.7256 (dev) and 0.6675 (test). In summary, providing relevant examples improved the generative task but not the classification task. However, classification still outperformed generation on this shared task.

Based on the above experimental results, we chose classification-based training using banglabert_large for further optimization. we experimented with pseudo-labeling methods. Experiment 1 added 1 round, results show improvement over no pseudo-labeling. Experiment 2 added 2 rounds but gained little versus 1 round, slight dev F1 increase, slight test decrease. Pseudo-labeling boosted performance over no augmen-

[1] https://huggingface.co/csebuetnlp/banglabert_large

[2] https://huggingface.co/csebuetnlp/banglabert

[3] https://huggingface.co/google/muril-large-cased

[4] https://huggingface.co/google/muril-base-cased

[5] https://huggingface.co/xlm-roberta-base

[6] https://huggingface.co/google/mt5-large

ID	Base Model	Training Objective	# of Pseudo-Labeling Rounds	F1-Micro on Dev Set	F1-Micro on Test Set
Original Data Format					
1	banglabert_large	Classification	1	0.7376	0.7267
2	banglabert_large	Classification	2	**0.7384**	0.7224
3	banglabert_large	Classification	0	0.7311	0.7242
4	banglabert	Classification	0	0.7345	0.7236
5	muril-large-cased	Classification	0	0.7303	**0.7307**
6	muril-base-cased	Classification	0	0.7179	0.7081
7	xlm-roberta-base	Classification	0	0.7076	0.7033
8	mt5-large	Generation	0	0.7095	0.7070
In-Context Learning Data Format					
9	mt5-large	Generation	0	0.7189	0.7082
10	banglabert_large	Classification	0	0.7256	0.6675

Table 2: Performance comparison of the submitted solution (shaded) and alternative approaches.

ID	Base Model	Training Objective	# of Pseudo-Labeling Rounds	F1-Micro on Dev Set	F1-Micro on Test Set
1	banglabert_large	Classification	0	0.7311	0.7242
2	banglabert_large	Classification	1	0.7376	**0.7267**
3	banglabert_large	Classification	2	**0.7384**	0.7224
4	xlm-roberta-base	Classification	0	0.7076	0.7093
5	xlm-roberta-base	Classification	1	0.7141	0.7155
6	xlm-roberta-base	Classification	2	**0.7225**	**0.7246**
7	muril-large-cased	Classification	0	0.7303	0.7307
8	muril-large-cased	Classification	1	0.7353	0.7355
9	muril-large-cased	Classification	2	**0.7397**	**0.7401**

Table 3: Pseudo-labeling performance from different models

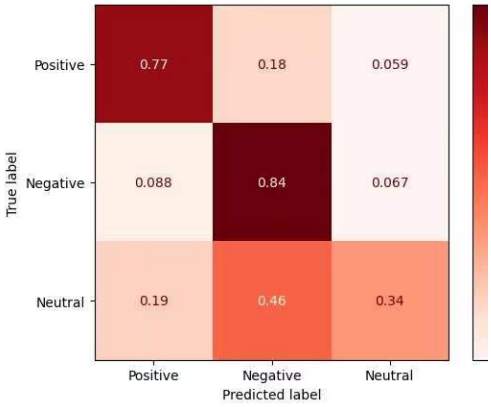


Figure 1: Test set confusion matrix

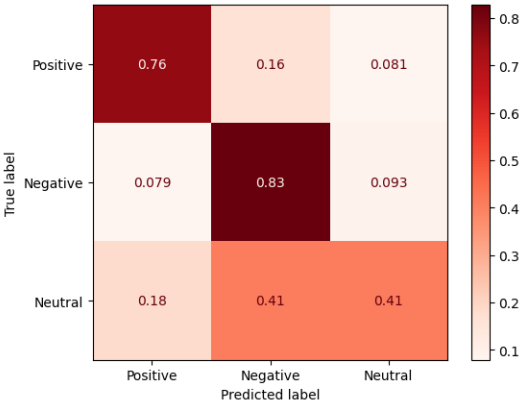


Figure 2: Test set confusion matrix after Pseudo-Labeling

tation. However, increasing from 1 to 2 rounds brought marginal gains on dev, marginal losses on test. This suggests 1 round sufficiently improves banglabert_large on this dataset, while additional rounds may lead to overfitting.

In order to validate the effectiveness of the pseudo-labeling method using ensemble models, we conducted experiments on three models - banglabert_large, xlm-roberta-base and muril-large-cased. The detailed experimental results are shown in the table 3. The results show that for most models, 1 to 2 rounds of pseudo-labeling led to improved performance on both dev and test sets. The banglabert_large model, the model that we submitted to the leaderboard during the evaluation period, achieved the best F1-Micro of 0.7384 on the dev set after 2 rounds of pseudo-labeling. Overall, the experimental results validate that the pseudo-labeling method can effectively improve pretrained language models' performance on downstream tasks.

We also visualized the results on the test set using confusion matrices. Figure 1 shows the confusion matrix for the predictions of the ensemble banglabert_large model on the test set. Figure 2 presents the confusion matrix for the banglabert_large ensemble model after pseudo-label training. Through comparing the two confusion matrices, it can be observed that the model performed relatively poorly on the neutral class - the banglabert_large model achieved an F1 of only 0.34 for the neutral category. After applying the model ensemble pseudo-labeling algorithm, the F1 for the neutral class improved to 0.41. The visualization via confusion matrices and comparison between the banglabert_large model before and after pseudo-labeling provides insight into the performance gain on the challenging neutral sentiment class through utilizing model ensembling and pseudo-labeling.

4 Conclusion and Future Work

In this work, we presented our approach and results for the Sentiment Analysis Shared Task. Our proposed solution using banglabert_large achieved strong performance, ranking 2nd out of 30 submitted systems with an F1-micro score of 0.7267. Through post-competition analysis, we found that larger transformer models designed specifically for Indian languages, such as BanglaBert and Muril, lead to better performance on this multi-class sentiment analysis task.

For low-resource languages like Bangla, pretrained models tailored to the specific language are crucial, as our results demonstrated the superior performance of Bangla-focused models over multilingual models. However, when languages have limited resources, starting with multilingual models from related language families can provide an initial boost, as evidenced by the strong test set results of muril-large-cased pretrained on Indian languages.

As resources grow, continued pre-training of language-specific models on larger and more diverse corpora for that language can further improve adaptation. Additionally, leveraging semi-supervised approaches and generative data augmentation techniques to expand limited labeled datasets will become more viable. Techniques like consistency training, backtranslation, and synthetic data generation can help in low-resource scenarios but require a certain data baseline to be effective.

References

Firoj Alam, Md Arid Hasan, Tanvir Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer.

M1437 at BLP-2023 Task 2: Harnessing Bangla Text for Sentiment Analysis: A Transformer-based Approach

Majidur Rahman and **Özlem Uzuner**
George Mason University, Virginia, USA
{mrahma37, ouzuner}@gmu.edu

Abstract

Analyzing public sentiment on social media is helpful in understanding the public's emotions about any given topic. While numerous studies have been conducted in this field, there has been limited research on Bangla social media data. Team M1437 from George Mason University participated in the Sentiment Analysis shared task of the Bangla Language Processing (BLP) Workshop at EMNLP-2023. The team fine-tuned various BERT-based Transformer architectures to solve the task. This article shows that $BanglaBERT_{large}$, a language model pre-trained on Bangla text, outperformed other BERT-based models. This model achieved an F1 score of 73.15% and top position in the development phase, was further tuned with external training data, and achieved an F1 score of 70.36% in the evaluation phase, securing the fourteenth place on the leaderboard. The F1 score on the test set, when $BanglaBERT_{large}$ was trained without external training data, was 71.54%.

1 Introduction

Social networking platforms have emerged as avenues where people share their thoughts and feelings on diverse subjects such as entertainment, politics, and education (Chen et al., 2022). Natural Language Processing (NLP) can effectively evaluate the sentiment of a text (Medhat et al., 2014) and explore the information discussed in social networking platforms. However, most research in this field has focused on English as the primary language; many other languages (e.g., Bangla) have remained largely unexplored (Sazzed, 2020; Islam et al., 2020).

Despite being the seventh most commonly spoken language worldwide, as well as the sixth in terms of native speakers (Babbel, 2023), Bangla is regarded as a low-resource language (Alam et al., 2021). The inaugural Bangla Language Processing (BLP) Workshop (Hasan et al., 2023a) sought to address sentiment analysis of Bangla social media posts. Within the scope of this workshop's sentiment analysis shared task, two datasets were utilized: the **MUltiplatform BAngla SEntiment (MUBASE)** (Hasan et al., 2023b) dataset, which features tweets and Facebook posts paired with their corresponding sentiment polarity, and the **Sentiment on Noisy Bangla texts (SentNoB)** (Islam et al., 2021) dataset, which consists of user comments on news articles and social media videos in various domains, such as education, politics, etc.

This paper presents our solution to sentiment analysis in Bangla on the workshop datasets. Our experiments with various Bidirectional Encoder Representations from Transformers (BERT)-based models (Devlin et al., 2019) indicated that $BanglaBERT$ (Bhattacharjee et al., 2022), a BERT language model that is pretrained on more than 27 GB of Bangla data is effective for classifying Bangla text sentiment. This system achieved an F1 score of 73.15% during the development phase. To further improve performance, we supplemented the training set with the CogniSenti dataset (Hasan et al., 2020) containing Facebook posts and tweets authored by Bangla speakers. This updated system achieved the best F1 score of 70.36% on the test set, securing the fourteenth place on the evaluation leaderboard. Without training data from CogniSenti Dataset, the F1 score was 71.54%. Our code is publicly available on GitHub[1].

2 Related Work

Extensive research has been carried out regarding sentiment analysis in languages with abundant resources, such as English. Traditional sentiment analysis approaches on resource-abundant languages relied heavily on syntactic parsing (Nasukawa and Yi, 2003). The advent of Transformer-based architectures (Vaswani et al., 2017), such as

[1] https://github.com/majidurrahman1437/blp-shared-task2

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 279–285

BERT (Devlin et al., 2019), greatly improved the state-of-the-art (Socher et al., 2013) on sentiment classification (Munikar et al., 2019).

Low-resource languages have traditionally lagged behind these advancements. In recent years, however, the NLP community has turned its attention to low-resource languages like Bangla. Sentiment analysis for low-resource languages became one of the tasks to receive attention. The availability of high-quality datasets, such as aspect-based sentiment analysis (ABSA) of Bangla text (Rahman et al., 2018) dataset, has supported sentiment analysis in Bangla. Example approaches to sentiment analysis on Bangla primarily utilized long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997; Tripto and Ali, 2018; Rezaul Karim et al., 2020). The SentNoB dataset (Islam et al., 2021) was introduced in 2021, which consists of noisy Bangla texts. Islam et al. showed that combining lexical features resulted in better performance than neural models for SentNoB. Hasan et al. developed the CogniSenti dataset (Hasan et al., 2020), which leverages Transformer models like XLM-RoBERTa (Conneau et al., 2020) to predict sentiment polarity in Bangla text, with promising results.

In a recent comparative study of various Bangla sentiment classification datasets using different Transformer-based architectures, XLM-RoBERTa outperformed all models (Alam et al., 2021). These results demonstrate the growing potential of Transformer-based architectures to improve language processing even in low-resource languages such as Bangla. BanglaBERT (Bhattacharjee et al., 2022) is a language model based on BERT, pre-trained on a large dataset of 27.5 GB of Bangla text. It has yielded state-of-the-art results in Bangla sentiment classification. While there are some promising research directions for Large Language Models (LLM) to perform Bangla sentiment analysis (Hasan et al., 2023b), existing pre-trained language models, such as BanglaBERT, can outperform them. Although there has been a sentiment analysis shared task for Indian languages, including Bangla, in the past (Patra et al., 2015), there has been a lack of initiatives to organize such a task for the Bangla language specifically. The Bangla sentiment analysis shared task at the first BLP workshop (Hasan et al., 2023a) aims to highlight the research efforts of Bangla researchers from around the world.

3 Methods

3.1 Data

The dataset used in this shared task consists of samples from the MUltiplatform BAngla SEntiment (MUBASE) (Hasan et al., 2023b) and SentNoB (Islam et al., 2021) datasets. The former contains Bangla language posts from social media platforms like Twitter and Facebook, which have undergone manual annotation for sentiment analysis. The latter comprises comments from multiple social media domains; each has also been manually annotated for sentiment.

The dataset comprises three sentiment classes: Negative, Neutral, and Positive. The proportion of Negative, Neutral, and Positive examples is kept uniform across the training and validation splits, whereas in the test split, the ratio is almost similar to the train and validation split. The distribution of labels across data splits is illustrated in Figure 1.

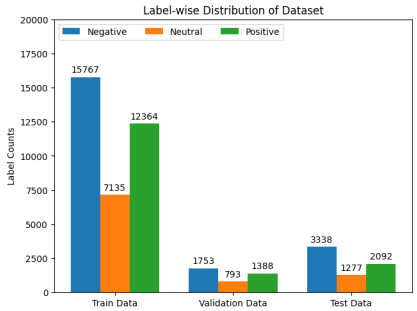


Figure 1: Label-wise Distribution of the Dataset

External Training Data: As can be observed from the class distribution of the training data, the "Neutral" class is under-represented compared to the other two sentiment classes. In order to boost the representation of the "Neutral" class and to reduce the class imbalance, we integrated external training data from the CogniSenti dataset (Hasan et al., 2020) to the original training set. The CogniSenti dataset consists of 6570 instances, encompassing three sentiment categories (Negative, Neutral, Positive) extracted from Tweets and Facebook posts written by native Bangla speakers. It features examples from diverse domains, including politics, current affairs, etc. By merging the complete dataset with the provided training set, we create a new training set of more than 41k samples. The distribution of the dataset across various categories is presented in Table 1.

Dataset	Class	Number of Samples
CogniSenti	Negative	1333
	Neutral	3749
	Positive	1488
	Total	6570
Merged (BLP Train Set + CogniSenti)	Negative	17100
	Neutral	10884
	Positive	13852
	Total	41836

Table 1: Data Distribution of External Training Data (CogniSenti Dataset), Along With Merged Training Data Per Class

3.2 BanglaBERT

BanglaBERT language model utilizes ELECTRA (Clark et al., 2020) as its foundation due to ELECTRA's superior computational efficiency compared to BERT. BanglaBERT is pre-trained on 27.5 GB of Bangla text from various sources such as news, encyclopedias, and blogs. The $BanglaBERT_{base}$ model includes 12 Transformer Encoder layers with 768 hidden units, 12 attention heads, and 110M parameters, while the $BanglaBERT_{large}$ model boasts 24 Transformer Encoder layers with 1024 hidden units, 16 attention heads, and 335M parameters (Bhattacharjee et al., 2022).

3.3 Evaluation

The official evaluation metric for the Sentiment Analysis shared task is the micro-F1 score (Pedregosa et al., 2011).

3.4 Experimental Setup

We utilized BanglaBERT with the aid of HuggingFace transformers library (Wolf et al., 2019). Our model is trained on NVIDIA DGX-A100 GPU nodes, with a maximum input sequence length of 512. We conducted hyperparameter tuning on the learning rate, seed, training batch size, and number of training epochs to achieve optimal performance. The model undergoes ten epochs of training, with a training batch size of 32 and a seed value of 18. We set the learning rate 3e-5 utilizing the Adam (Kingma and Ba, 2014) optimizer and a linear warmup with a warmup ratio of 0.001. We develop our models on the provided development set and validate utilizing the development-test (dev-test) and test sets during the development and evaluation phases, respectively.

4 Results and Discussions

During the development phase, our system attains the top position in the leaderboard, which is evaluated using the dev-test split. During the evaluation phase, our model ranks as the fourteenth-best model evaluated using the test split, as illustrated in Table 2.

Model	F1 Score (%)
Development Phase	
M1437	**73.15**
MoFa_Aambela	73.03
yangst	72.88
Hari_vm	72.48
amlan107	72.24
Evaluation Phase	
MoFa_Aambela	**73.10**
yangst	72.67
amlan107	71.79
Hari_vm	71.72
PreronaTarannum	71.64
ShadmanRohan	71.55
M1437 (latest submission)	70.36
M1437 (best submission)	71.54

Table 2: Performance Comparison on the Dev-Test Set and Test Set of Our System Submissions

4.1 Performance with External Data

Upon merging the CogniSenti dataset with the BLP sentiment analysis shared task train set, we analyze our latest submission, which utilizes $BanglaBERT_{large}$. Unfortunately, we discovered that incorporating external data did not improve the performance of our model. Following an in-depth investigation into our model's inaccuracies, we uncovered that 331 instances were classified as "Positive" when they should have been labeled as "Negative". Upon further analysis of these predictions, including phrases such as 'চ্যালেঞ্জের মুখে নার্সারি ব্যবসায়ীরা" (The nursery traders are facing challenges), "কেন ঝুঁকি থাকলেও এখনো মশক কর্মী নিয়োগ হচ্ছে না?" (Why the mosquito workers are still not recruited despite having risks?), we observed that our model struggled to detect the "Negative" sentiment in these samples accurately. On the contrary, the model that was trained without incorporating CogniSenti data accurately identified 91 of the 331 "Negative" samples.

Our further analysis discovered that 50% of the incorrect predictions were originally labeled as "Negative" but fell under the "Neutral" category. Likewise, 33.26% of mispredictions were

previously labeled as "Positive" but were classified as "Neutral". Examples of "Negative" sentiments that were misclassified as "Neutral" were found in the CogniSenti data-trained model, such as "এদিকে শহরের মানুষের বিদ্যুৎ অপচয় ও বিদ্যুৎ নির্ভরতা বাড়তেই আছে ।" (Meanwhile, the city's electricity consumption and electricity dependence continues to increase!), "আপনার কাছ থেকে এমন বক্তব্য আশা করি না" (I do not expect such a statement from you). However, these examples were predicted correctly by the model trained without CogniSenti data. The merged dataset had a higher proportion of "Neutral" to "Negative" samples, resulting in a more effective prediction of "Neutral" sentiment examples but leading to a higher number of mispredictions for the "Positive" and "Negative" sentiment examples compared to the model trained without CogniSenti data. This is supported by the fact that the model trained without CogniSenti data made 501 mispredictions for the "Negative" sentiment category, while the model trained with CogniSenti data made 663 mispredictions for the same category.

4.2 Performance Without External Data

Model	F1 Score (%)
Random Baseline	33.56
Majority Baseline	49.77
n-gram Baseline	55.14
Logistic Regression	55.05
Decision Tree Classifier	48.68
multi-lingual BERT-cased	64.20
XLM-RoBERTa_large	68.21
MuRIL_base	68.39
IndicBERT	70.82
$BanglaBERT_{base}$	71.49
$BanglaBERT_{large}$	**71.54**

Table 3: Performance Comparison on the Test Set Across Various BERT Models

Comparative Study Across Baselines: Prior to the commencement of the shared task, the organizers released three baseline scores for the dev-test set and the test set. The initial score, referred to as the random baseline, randomly predicts a label from the three likely class labels. The second score, known as the majority baseline, employs the "DummyClassifier" from the sklearn library (Pedregosa et al., 2011) and predicts the most frequent class label for each instance. Lastly, the third baseline, named the n-gram baseline, employs the TF-IDF vectorization (Salton and Buckley, 1988) technique to generate feature vectors and the Support Vector Machine classifier (Noble, 2006) to provide predictions on the test set. Moreover, we have conducted a comparison of our model's performance, $BanglaBERT_{large}$, trained only on the BLP sentiment analysis train set by utilizing the test set specified in Table 3, with that of conventional machine learning classifiers, namely Logistic Regression (Wright, 1995) and Decision Tree Classifier (Swain and Hauska, 1977). To extract features, we utilized a similar TF-IDF vectorization technique and independently applied Logistic Regression and Decision Tree Classifier to generate predictions on the test set. Our assessment demonstrates that the baselines and traditional machine learning classifiers were not able to develop a robust model due to their inability to grasp the intricacies of the input text and context.

Comparative Study Across BERT models: We further assess the performance of our top-performing model as specified previously. Our findings reveal that the BERT-based models exceed the performance of other models chosen for comparison. One of these models is the multi-lingual BERT-cased (mBERT) (Devlin et al., 2019), which is trained in 104 languages, including Bangla. However, it's worth noting that multi-lingual models typically yield better results for high-resource languages and may not perform as well on lower-resource languages like Bangla (Wu and Dredze, 2020). Multilingual language models such as MuRIL (Khanuja et al., 2021) and IndicBERT (Doddapaneni et al., 2023) have undergone pretraining on a range of Indian languages, including Bangla, through the use of monolingual, translated, and transliterated text. These models have demonstrated superior performance in comparison to mBERT, a similar multilingual language model. However, it is worth highlighting that although these models are multilingual, this is also the primary reason for their inability to surpass our model's performance. Research has shown that XLM-RoBERTa (XLM-R) (Conneau et al., 2020), despite having more model parameters (550M), is unable to outperform $BanglaBERT_{large}$ due to its limited pretraining knowledge of Bangla text (8.7 GB). In contrast, $BanglaBERT_{large}$ has access to a vast amount of pretraining knowledge (27.5 GB) specific to the Bangla language. This highlights the importance of having a substantial amount of language-specific pretraining knowledge, which aids in generating robust context-

No. of Input Tokens	No. of Train Samples	No. of Test Samples	Prediction Correctness (%)
1 to 20	28108	5490	72.91
21 to 40	5612	895	67.82
41 to 60	994	174	65.52
61 to 80	241	69	53.62
81 to 100	101	29	48.28
101 to 150	109	28	46.43
151 to 200	40	12	33.33
201 or higher	170	10	60.00

Table 4: Performance Comparison of Test Set According to Input Token Length

aware embedding vectors and ultimately improves model performance.

Error Analysis: Based on our findings, it appears that the model trained without CogniSenti data exhibits higher true positive rates for the "Negative" and "Positive" classes at 84.99% and 75.76%, respectively, compared to only 29.44% for the "Neutral" class. Our model is more proficient at learning examples from the "Negative" and "Positive" classes while struggling with the "Neutral" class due to the data imbalance in our training set. In fact, 69.59% of the mispredictions regarding the "Neutral" class actually belong to the "Negative" class, which can be attributed to the larger number of "Negative" examples in our training set. To ensure unbiased outcomes, a well-balanced dataset with comparable sample sizes in each class is essential for optimal performance.

Examining FP and FN: We thoroughly analyze the mispredictions made by the model trained without CogniSenti data, specifically when it predicts a "Positive" sentiment instead of a "Negative" one, or vice versa, according to the gold label. We examine texts such as "উনার রেস্ট দরকার । । ।" (They need rest...), "একাদশ জাতীয় সংসদের ৯ম অধিবেশন শুরু ৬ সেপ্টেম্বর" (The 9th session of the eleventh National Parliament begins on September 6), "আস্তাগফিরুল্লাহ!" (God forgive us!), and "গ্রাহকের কাঁধে আর থাকছে না বাড়তি বিলের বোঁঝা" (The customer is no longer burdened with additional bills). While these texts are labeled as "Negative" in the gold label, the model may not have enough background knowledge to accurately label them as "Negative" instead of "Positive". Similar cases have been observed in texts such as "দেশে আরও ৩ জন করোনাভাইরাসে আক্রান্ত" (3 new people infected with coronavirus in the country), "ছিল না ফরম ফিলাপের টাকা, অভাব ছিল নিত্যসঙ্গী" (Neither there was money for filling up the form nor a daily companion) where the model predicts a text to be "Negative" whereas the actual label is "Positive". It has been observed that a significant number of erroneous predictions can be attributed to political and national affairs, which are over-represented in the dataset. It is imperative to acknowledge the potential biases that can result from such imbalances and to devise strategies for mitigating their impact to ensure the accuracy and reliability of the predictions. This issue highlights the importance of careful data curation and analysis in the context of predictive modeling, particularly when dealing with sensitive or complex domains.

Performance Comparison by Text Length: To assess the performance of the model trained without CogniSenti data on texts of varying lengths, we closely monitor its predictions on the test set. Our evaluation reveals that our model accurately predicts approximately most of the 5.5k samples with up to 20 tokens. However, as the input text length increases, the F1 score declines. Notably, the model's F1 score is highest (72.91%) for texts with up to 20 tokens, dropping to 33.33% for texts with 151 to 200 tokens. This suggests that the model learns to predict shorter texts more precisely, possibly due to more training examples with 20 tokens or less as per table 4. In order to facilitate the learning process for longer inputs, it may be advantageous to consider augmenting the training data with lengthier texts.

5 Conclusion

Team M1437 had the privilege of participating in the Bangla Sentiment Analysis challenge during the inaugural BLP workshop at EMNLP-2023. For this task, we prefer the $BanglaBERT_{large}$ as our language model due to its exceptional pre-trained proficiency in the Bangla language. During development, our system ranked first on the leaderboard. Although we achieved a comparable F1 score during the evaluation phase, we remain committed to exploring a range of Large Language Models (LLMs) to improve the true positive rates for longer input sequences.

Limitations

In an effort to enhance our model's ability to generalize across all labels, we integrated the CogniSenti dataset into the training set. Unfortunately, the model's performance did not meet our expectations in this particular scenario. However, this can be due to the specific dataset chosen and leaves open the question of whether other datasets would yield similar results. We, therefore, remain committed to examining other relevant datasets that can not only supplement the training data but also enhance the model's performance across all sentiment classes.

Ethics Statement

The dataset used in this research complies with a non-commercial share-alike international license by Creative Commons [2], which is taken under careful consideration. The research does not use this dataset for any commercial purpose.

References

Firoj Alam, Md Arid Hasan, Tanvir Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Babbel. 2023. The 10 most spoken languages in the world.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Junhan Chen, Yumin Yan, and John Leach. 2022. Are emotion-expressing messages more shared on social media? a meta-analytic review. *Review of Communication Research*, 10.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Md. Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020. Sentiment classification in bangla textual content: A comparative study. In *23rd International Conference on Computer and Information Technology (ICCIT)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Khondoker Ittehadul Islam, Md Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5. IEEE.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja

[2] https://creativecommons.org/licenses/by-nc-sa/4.0/

Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77.

William S Noble. 2006. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *Proc. of MIKE*, pages 650–655. Springer.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Md Rahman, Emon Kumar Dey, et al. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2):15.

Md Rezaul Karim, Bharathi Raja Chakravarthi, Mihael Arcan, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced Bengali language based on multichannel convolutional-lstm network. *arXiv*, pages arXiv–2004.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Salim Sazzed. 2020. Cross-lingual sentiment classification in low-resource Bengali language. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 50–60, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Philip H Swain and Hans Hauska. 1977. The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3):142–147.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *Proc. of ICBSLP*, pages 1–6. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Raymond E Wright. 1995. Logistic regression.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

nlpBDpatriots at BLP-2023 Task 2: A Transfer Learning Approach to Bangla Sentiment Analysis

Dhiman Goswami*, **Md Nishat Raihan***, **Sadiya Sayara Chowdhury Puspo***, **Marcos Zampieri**
George Mason University
{dgoswam, mraihan2, spuspo, mzampier}@gmu.edu

Abstract

In this paper, we discuss the nlpBDpatriots entry to the shared task on Sentiment Analysis of Bangla Social Media Posts organized at the first workshop on Bangla Language Processing (BLP) co-located with EMNLP. The main objective of this task is to identify the polarity of social media content using a Bangla dataset annotated with positive, neutral, and negative labels provided by the shared task organizers. Our best system for this task is a transfer learning approach with data augmentation which achieved a micro F1 score of 0.71. Our best system ranked 12^{th} among 30 teams that participated in the competition.

1 Introduction

NLP has become a major domain of modern computational research, offering a lot of applications from machine translation to chatbots. However, much of this research has been concentrated on English and other high-resource languages like French, German, and Spanish.

Bangla, despite being the seventh most spoken language in the world with approximately 273 million speakers (Ethnologue, 2023), has not received similar attention from the NLP community. This gulf is not just an academic oversight; it has real-world implications. Bangla is a language of significant cultural heritage and economic activity. The development of NLP technologies for Bangla is both a scientific necessity and a practical imperative. The limited availability of Bangla NLP resources has led to a reliance on traditional machine learning techniques like SVMs and Naive Bayes classifiers for classification tasks such as sentiment analysis. The advent of deep learning models has opened new avenues. Models like BERT (Devlin et al., 2019) have shown promising results in languages other than English and has been recently trained to support Bangla (Kowsher et al., 2022).

Sentiment analysis is increasingly becoming a vital tool for understanding public opinion and people's behavior (Rosenthal et al., 2017). It has found applications in various sectors, including finance, where it helps investors to leverage social media data for better investment decisions (Mishev et al., 2020). In the context of Bangla, the utility of sentiment analysis extends beyond mere academic interest. It can serve as a powerful tool for businesses to gauge customer satisfaction, for policymakers to understand public sentiment, and even for social scientists studying behavioral trends.

In this paper, we evaluate several models and implement transfer learning for the shared task on Sentiment Analysis of Bangla Social Media Posts organized at the first workshop on Bangla Language Processing (BLP) (Hasan et al., 2023a). Moreover, an ensemble model consisting of three transformer-based models generates a superior performance over the other approaches.

2 Related Work

Initiating Sentiment Analysis in Bangla Sentiment analysis, which was mainly focused on English (e.g. Yadav and Vishwakarma 2020, Saberi and Saad 2017), is now becoming popular in other low resource languages like Urdu (e.g. Noor et al. 2019, Muhammad and Burney 2023), Pashto (e.g. Iqbal et al. 2022, Kamal et al., Kamal et al.), Bangla (e.g. Islam et al. 2020, Akter et al. 2021). Researchers are actively working to improve how people analyze and modify Bangla online comments using different methods and datasets. They are doing a variety of tasks, from classifying documents to mining opinions and analyzing sentiment, all while adapting their techniques to the specifics of the Bangla language. For example, for document classification, Rahman et al. (2020) presented

*These three authors contributed equally to this work.
WARNING: This paper contains examples that are offensive in nature.

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 286–292

an approach using the transformer-based models BERT and ELECTRA with transfer learning. The models were fine-tuned on three Bangla datasets. Similarly, Rahman et al. (2020) explored character-level deep learning models for Bangla text classification, testing Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models. On the other hand, for opinion mining, Haque et al. (2019) analyzed Bangla and Phonetic Bangla restaurant reviews using machine learning on a dataset of 1500 reviews. SVM achieved the highest accuracy of 75.58%, outperforming prior models.

Advancements of Sentiment Analysis in Bangla Islam et al. (2020) presented two new Bangla sentiment analysis datasets which achieved state-of-the-art results with multi-lingual BERT (71% accuracy for 2-class, 60% for 3-class), and notes sentiment differences in newspaper comments. Tuhin et al. (2019) proposed two Bangla sentiment analysis methods: Naive Bayes and a topical approach, aiming at six emotions, which achieved over 90% accuracy for sentence-level emotion classification, outperforming Naive Bayes. Similarly, Al Kaiser et al. (2021) discussed research focused on sentiment analysis and hate speech detection in Bangla language Facebook comments; compiling a dataset of over 11,000 comments, categorized by polarity (positive, negative, neutral) and various sentiment types, including gender-based hate speech. Furthermore, there are researches conducted on sentiment analysis in the field of online Bangla reviews. For example, Khan et al. (2020) detected depression in Bangla social media using sentiment analysis. They preprocessed a small dataset and employed machine learning classifiers, but faced limitations due to the dataset's size and basic classifiers.

Akter et al. (2021) used machine learning for Bangla e-commerce review sentiment analysis, with KNN achieving 96.25% accuracy, outperforming other classifiers. This highlighted machine learning's potential in analyzing Bangla e-commerce reviews. Whereas, Banik and Rahman (2018) introduced a Bangla movie review sentiment analysis system using 800 annotated social media reviews. (Hasan et al., 2023b) introduced a significant dataset of 33,605 manually annotated Bangla social media posts and examined how different language models perform in zero- and few-shot learning situations. Thus, the research of sentiment analysis is continuously growing, and it's helping us better understand sentiment in Bangla online content.

3 Dataset

The dataset provided for the shared task (Hasan et al., 2023a), consists of a training set, a development set, and a blind test set. For each set, the texts have been annotated using three labels - 'Positive', 'Neutral', or 'Negative' (Islam et al., 2021). The label distribution for each set is provided in Table 1.

Label	Train	Dev	Test
Positive	35%	35%	31%
Neutral	20%	20%	19%
Negative	45%	45%	50%

Table 1: Distribution of instances and labels across training, development, and test sets.

The dataset is imbalanced across the labels, hence it is challenging for the models to learn well.

4 Experiments

We conduct a wide range of experiments with several models and data augmentation strategies. Our experiments include statistical models, transformer-based models; data augmentation strategies like back-translation, multilinguality and also prompting proprietary LLMs.

Statistical ML Classifiers In our experiments, we use statistical machine learning models like Logistic Regression and Support Vector Machine using TF-IDF vectors. We implement both models and some hyperparameter tuning. While SVM performs better with a 0.55 F1 score (Micro) the overall results do not improve much.

Transformers We also test several transformer-based models which are pre-trained on Bangla data. Our initial experiments include Bangla-BERT (Kowsher et al., 2022) which is only pre-trained on bangla corpus. We finetune the model on the train set and evaluate it on the dev set with empirical hyperparameter tuning. We get 0.64 as the best micro F1 using Bangla-BERT. We then use multilingual transformer models like multilingual-BERT (Devlin et al., 2019) and xlm-roBERTa (Conneau et al., 2020), which are pre-trained on 104 and 100 different languages respectively, including Bangla.

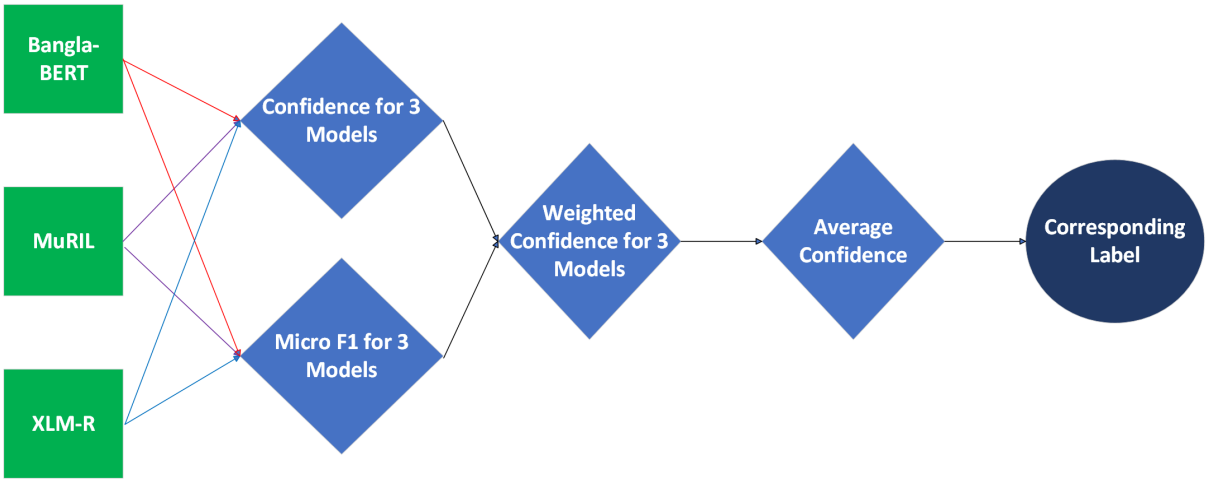


Figure 1: Workflow of the Ensemble Model

We also do the same hyperparameter tuning with both models. While mBERT gets a 0.60 Micro F1 score, xlm-roBERTa does better with 0.71 on the dev set and 0.70 on the test set. Lastly, we use MuRIL (Khanuja et al., 2021), another transformer pre-trained in 17 Indian languages including Bangla. It has a test micro F1 score of 0.67. While experimenting with these models, we observe the losses while fine-tuning to make sure the models do not overfit.

Prompting Next, we try prompting with gpt-3.5-turbo model (OpenAI, 2023) from OpenAI for this classification task. We use the API to prompt the model, while providing a few examples for each label and ask the model to label the dev and test set. The model does not do well with a micro F1 of 0.57 on the dev and 0.51 on the test set.

Transfer Learning on Augmented Data Finally, we augment the data of the Bangla YouTube Sentiment and Emotion dataset by Hoq et al. (2021). The dataset has highly positive (2), positive (1), neutral (0), negative (-1) and highly negative (-2) labels. We merge the highly positive and positive labels to Positive, negative and highly negative labels to Negative and keep the neutral label unchanged. This is how we get three labels out of five and merge it with our train data. Following this procedure, we get 0.71 micro F1 score for test dataset.

Ensemble After finding the results of transformer-based models, we perform an ensemble approach on BanglaBERT, MuRIL, and XLM-R. We then find the weighted average confidence of these three models. For Negative, the confidence interval is fixed 0.0 - 0.33, for Neutral between 0.33 to 0.66 exclusive and for Positive 0.66 - 1.0. The weights are their corresponding test F1 scores found in Table 3. With that confidence interval, we predict the test labels. We get a 0.72 micro F1 score by this approach. However this result is not reported to the shared task test phase as we get this result by additional experiments. The detailed label prediction procedure is given in Table 2 and the workflow of the whole ensemble method is given in Figure 1. For the first instance, the example is indeed Neutral but BanglaBERT predicts it borderline Negative and XLM-R predicts it Positive. But the power of ensemble approach bring it to the confidence interval of Neutral and thus predicts the label correctly. Similarly, for the second one, a corrected Neutral label is predicted from a Negative, Neutral and borderline Positive confidence. For the last two cases, Negative and Positive labels are determined correctly even with the presence of two Neutral confidence.

5 Results and Analysis

At the start of the share task competition, 3 baseline micro F1 scores are provided by the organizers. For random selection the provided baseline is 0.34, for majority selection 0.50, and n-gram 0.55. The results of different models are given in Table 3.

Amongst the statistical machine learning models, we use logistic regression and support vector machine. For logistic regression, we achieve a micro F1 score of 0.45 and for the support vector machine, the F1 is 0.55.

For transformer-based models, we use mBERT,

Text Example	BanglaBERT conf.	MuRIL conf.	XLM-R conf.	Average conf.	Label
আজ স্কুল গেলে ফুল পেতাম	0.32	0.51	0.99	0.61	Neutral
প্রধানমন্ত্রীর সাথে আমি একমত	0.01	0.52	0.68	0.41	Neutral
রাজধানীতে বালতির পানিতে পড়ে শিশুর মৃত্যু	0.49	0.35	0.01	0.28	Negative
মা মানেই আগলে রাখা	0.65	0.99	0.51	0.71	Positive

Table 2: Ensemble with Three Transformer Based Models based on Confidence Score

BanglaBERT, MuRIL and XLM-R where we get the best F1 score of 0.70 by XLM-R.

A few shot learning procedure is used by using GPT3.5 Turbo. We give a few instances of each label as prompt and got 0.51 F1 which is significantly lower than our other attempted approaches except logistic regression. It is because GPT3.5 is still not efficient enough for any downstream classification problem in bangla like this shared task.

Moreover, we augment the data of Bangla YouTube Sentiment and Emotion dataset by Hoq et al. (2021). The dataset has highly positive, positive labels which we consider as positive and negative, highly negative labels which we consider negative. We keep the neutral label unchanged. This is how we get three labels out of five labels and merge it with our train data. Following this procedure, we finally achieve micro F1 score of 0.71 which we this shared task's leader board.

Additionally, we perform ensemble method over the test micro F1 score of BanglaBERT, MuRIL and XLM-R. Instead of doing majority voting on the predicted test label, we find weighted average of confidence interval for the each instances of the test set for the three transformer based models shown in Table 3. With that confidence interval, test labels are predicted with 0.72 F1 score which is the best among all our experiments. A comparison bar chart for different models' performance is shown in Figure 2.

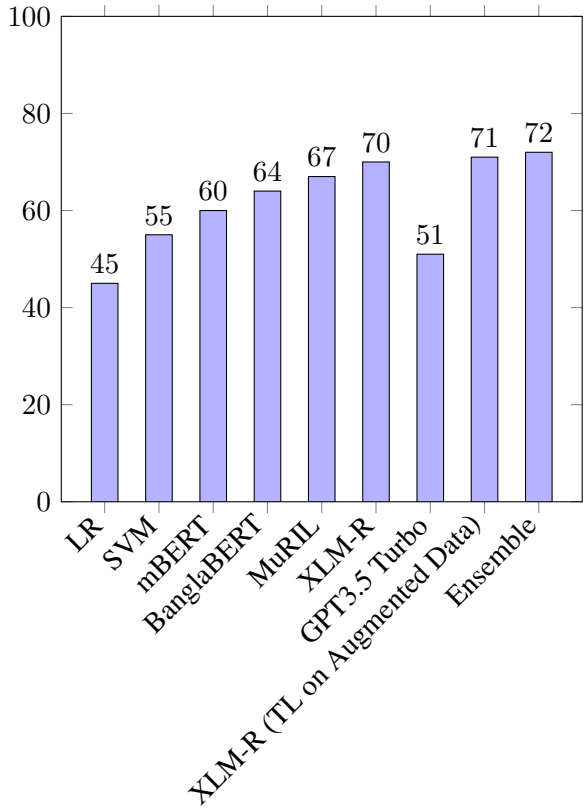


Figure 2: Models vs. Test Micro-F1 score (in percentage)

Models	**Dev**	**Test**
Logistic Regression	0.47	0.45
Support Vector Machine	0.56	0.55
mBERT	0.60	0.60
BanglaBERT	0.66	0.64
MuRIL	0.70	0.67
XLM-R	0.71	0.70
GPT 3.5 Turbo	0.57	0.51
XLM-R (Transfer Learning on Augmented data)	0.71	0.71
Ensemble	-	**0.72**

Table 3: Dev and Test micro F-1 score for different models and procedures

6 Error Analysis

The classification report provides a comprehensive understanding of our model's performance across the three classes. The overall accuracy of the model is 0.71. The 'Positive' class has the highest F1-score of 0.78, driven by a precision of 0.75 and a recall of 0.80. The 'Neutral' class, on the other hand, shows a relatively weaker performance with an F1-score of 0.42, a result of its lower precision and recall, 0.51 and 0.37 respectively. The 'Negative' class offers a competitive performance with an F1-score of 0.74, a precision of 0.72, and a recall of 0.76.

On a macro level, the average values indicate a precision of 0.66, recall of 0.64, and an F1-score of 0.65. When weighted by support, the averages show a slightly better picture with precision at 0.69, recall identical to the overall accuracy at 0.71, and an F1-score of 0.70.

Further dissecting the errors by text length offers more insights. Texts with lengths in the range of 50 to 100 characters contribute the most to the dataset, constituting 43.73% of the samples, and have an F1-score of 0.74. The second largest group, texts ranging from 20 to 50 characters, contribute 26.64% to the dataset with a slightly better F1-score of 0.70. It is also worth noting that the performance drastically reduces for texts with lengths between 500 and 1000 characters, yielding the lowest F1-score of 0.39, albeit they only make up 0.73% of the samples. Few misclassified examples are given in Figure 4.

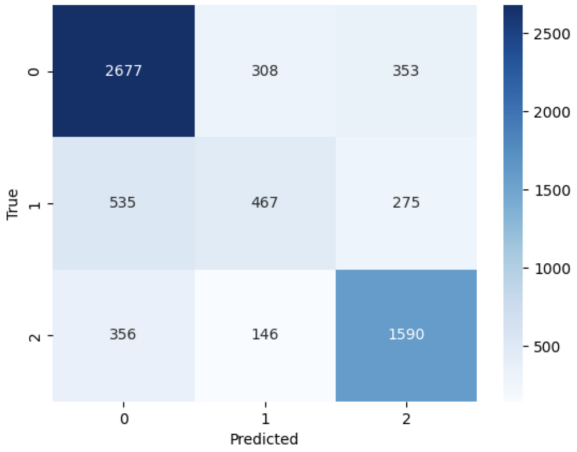


Figure 3: Confusion Matrix

Test Data Instance	Actual Label	Predicted Label
সিরিজে চতুর্থবার নাসুমের শিকার গ্র্যান্ডহোম	Positive	Negative
ইয়াছিন আহমদ কি নিয়ে কাজ করেন ?	Positive	Neutral
নরসিংদীতে ধর্ষণ ও হত্যা মামলার আসামি গ্রেপ্তার	Negative	Positive
মুসলিম হিসেবে সুবিধাগুলো নিবা কিন্তু অসুবিধা নিবা না এটা হয় না ।	Negative	Neutral
আমার ছেলের দুর্ভাগ্য না সৌভাগ্য জানিনা জ্বর এর জন্য স্কুল এ যেতে পারেনি !?	Neutral	Negative
প্রহসনের স্কুল খোলা । অধিকাংশ ক্লাসের শিক্ষার্থীর সপ্তাহে ১ দিন ক্লাস ।	Neutral	Positive

Figure 4: Few examples of misclassified labels

Text_Length	Micro_F1	Count	%
(0, 10]	0.67	69	1.03
(10, 20]	0.64	250	3.73
(20, 50]	0.70	1787	26.64
(50, 100]	0.74	2933	43.73
(100, 200]	0.69	1288	19.20
(200, 300]	0.64	202	3.01
(300, 500]	0.59	119	1.77
(500, 1000]	0.39	49	0.73
(1000, 5000]	0.80	10	0.15

Table 4: Performance Analysis Based on Text Length.

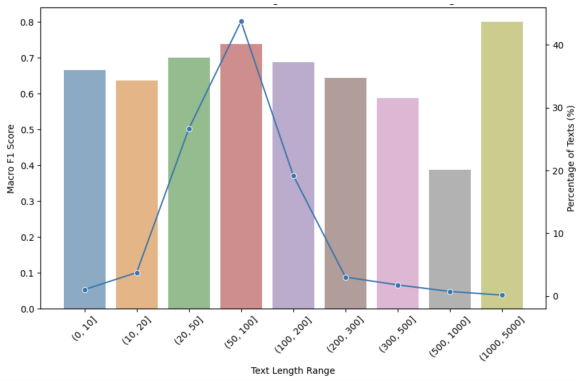


Figure 5: Performance Analysis

7 Conclusion

In this shared task, we use statistical machine learning models, transformer-based models, a few shot prompting, some customization with transformer-based models with transfer learning, data augmentation, and an ensemble-based approach. The transfer learning and data augmentation procedure is reported as the most successful approach in terms of a micro F1 score of 0.71. But additional experiments by doing an ensemble over three transformer-based models provide a 0.72 F1 score. Overall, this paper can be treated as a holistic experimental outcome for this shared task.

Limitations

Our transfer learning approach towards solving the problem presented for this shared task shows promising results. However, in most cases, our models keep overfitting. We use dropouts and weight decaying to handle the issue. Even though we perform a lot of hyper-parameter tuning with all the models, it might still be the case that we are not able to find the optimal set of parameters for a few models in our experiments.

Ethics Statement

The present study, which centers on the analysis of sentiment in Bangla text, rigorously adheres to the ACL Ethics Policy and seeks to make a valuable contribution to the realm of online safety. The dataset was supplied to us by the organizers and has undergone anonymization to secure the privacy of the users. The technology in question possesses the potential to serve as a beneficial instrument for the moderation of online content, thereby facilitating the creation of safer digital environments. However, it is imperative to exercise caution and implement stringent regulations to prevent its potential misuse for purposes such as monitoring or censorship.

References

Mst Tuhin Akter, Manoara Begum, and Rashed Mustafa. 2021. Bengali sentiment analysis of e-commerce product reviews using k-nearest neighbors. In *2021 International conference on information and communication technology for sustainable development (ICICT4SD)*, pages 40–44. IEEE.

Shad Al Kaiser, Sudipta Mandal, Ashraful Kalam Abid, Ekhfa Hossain, Ferdous Bin Ali, and Intisar Tahmid Naheen. 2021. Social media opinion mining based on bangla public post of facebook. In *2021 24th International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Nayan Banik and Md Hasan Hafizur Rahman. 2018. Evaluation of naïve bayes and support vector machines on bangla textual movie reviews. In *2018 international conference on Bangla speech and language processing (ICBSLP)*, pages 1–6. IEEE.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Ethnologue. 2023. The most spoken languages worldwide 2023.

Fabliha Haque, Md Motaleb Hossen Manik, and MMA Hashem. 2019. Opinion mining from bangla and phonetic bangla reviews using vectorization methods. In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–6. IEEE.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Muntasir Hoq, Promila Haque, and Mohammed Nazim Uddin. 2021. Sentiment analysis of bangla language using deep learning approaches. In *International Conference on Computing Science, Communication and Security*, pages 140–151. Springer.

Saqib Iqbal, Farhad Khan, Hikmat Ullah Khan, Tassawar Iqbal, and Jamal Hussain Shah. 2022. Sentiment analysis of social media content in pashto language using deep learning algorithms. *Journal of Internet Technology*, 23(7):1669–1677.

Khondoker Ittehadul Islam, Md Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5. IEEE.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Uzair Kamal, Imran Siddiqi, Hammad Afzal, and Arif Ur Rahman. 2016. Pashto sentiment analysis using lexical features. In *Proceedings of the Mediterranean Conference on Pattern Recognition and Artificial Intelligence*, pages 121–124.

Md Rafidul Hasan Khan, Umme Sunzida Afroz, Abu Kaisar Mohammad Masum, Sheikh Abujar, and Syed Akhter Hossain. 2020. Sentiment analysis from bengali depression dataset using machine learning. In *2020 11th international conference on computing, communication and networking technologies (ICCCNT)*, pages 1–5. IEEE.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages.

M Kowsher, Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, 10:91855–91870.

Kostadin Mishev, Ana Gjorgjevikj, Irena Vodenska, Lubomir T Chitkushev, and Dimitar Trajanov. 2020. Evaluation of sentiment analysis in finance: from lexicons to transformers. *IEEE access*, 8:131662–131682.

Khalid Bin Muhammad and SM Aqil Burney. 2023. Innovations in urdu sentiment analysis using machine and deep learning techniques for two-class classification of symmetric datasets. *Symmetry*, 15(5):1027.

Faiza Noor, Maheen Bakhtyar, and Junaid Baber. 2019. Sentiment analysis in e-commerce using svm on roman urdu text. In *Emerging Technologies in Computing: Second International Conference, iCETiC 2019, London, UK, August 19–20, 2019, Proceedings 2*, pages 213–222. Springer.

OpenAI. 2023. Gpt-3.5 turbo fine-tuning and api updates. Accessed: 2023-08-28.

Md Mahbubur Rahman, Md Aktaruzzaman Pramanik, Rifat Sadik, Monikrishna Roy, and Partha Chakraborty. 2020. Bangla documents classification using transformer based deep learning models. In *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pages 1–5. IEEE.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Bilal Saberi and Saidah Saad. 2017. Sentiment analysis or opinion mining: A review. *Int. J. Adv. Sci. Eng. Inf. Technol*, 7(5):1660–1666.

Rashedul Amin Tuhin, Bechitra Kumar Paul, Faria Nawrine, Mahbuba Akter, and Amit Kumar Das. 2019. An automated system of sentiment analysis from bangla text using supervised learning techniques. In *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, pages 360–364. IEEE.

Ashima Yadav and Dinesh Kumar Vishwakarma. 2020. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385.

Ushoshi2023 at BLP-2023 Task 2: A Comparison of Traditional to Advanced Linguistic Models to Analyze Sentiment in Bangla Texts

Sharun Akter Khushbu
Daffodil International University
sharun.cse@diu.edu.bd

Nasheen Nur
Florida Institute of Technology
nurn@fit.edu

Mohiuddin Ahmed
University of North Carolina at Charlotte
mahmed27@uncc.edu

Nashtarin Nur
United International University
nashtarin.nur@gmail.com

Abstract

This article describes our analytical approach designed for BLP Workshop-2023 Task-2: in Sentiment Analysis. During actual task submission, we used DistilBERT. However, we later applied rigorous hyperparameter tuning and preprocessing, improving the result to 68% accuracy and a 68% F1 micro score with vanilla LSTM. Traditional machine learning models were applied to compare the result where 75% accuracy was achieved with traditional SVM. Our contributions are a) data augmentation using the oversampling method to remove data imbalance and b) attention masking for data encoding with masked language modeling to capture representations of language semantics effectively, by further demonstrating it with explainable AI. Originally, our system scored 0.26 micro-F1 in the competition and ranked 30th among the participants for a basic DistilBERT model, which we later improved to 0.68 and 0.65 with LSTM and XLM-RoBERTa-base models, respectively.

1 Introduction

Sentiment analysis and opinion-mining techniques determine a text's sentiment or emotional polarity and then analyze it (Medhat et al., 2014). Throughout diverse fields, such as marketing, customer feedback analysis, and social media monitoring, sentiment analysis has gained significant attention in recent years. While sentiment analysis has been extensively studied in languages like English, there is a growing interest in applying this technique to other languages, including Bangla. Analyzing sentiment in Bangla text presents unique challenges due to its complex grammar, script, and nuances. This article aims to explore sentiment analysis in the Bangla language with an example dataset provided for the BLP workshop competition for task 2 using sequential data analysis models, such as LSTM and large language models, along with traditional models. This multi-class classification task determines whether the sentiment expressed in the text is positive, negative, or neutral.

Even though LSTM provides the highest performance among the deep learning models, XLM-RoBERTa-base (Singh et al., 2022) uses Masked Language Modeling (MLM) to handle multilingual and cross-lingual tasks, making it a powerful tool for understanding and generating text in multiple languages. MLM is a pre-training objective used in models like XLM-RoBERTa-base. Using MLM, a fraction of input tokens are replaced with unique [MASK] tokens, and the model is trained to predict the original tokens from the context provided by the surrounding tokens. MLM is a self-supervised learning task where a model learns to understand the statistical properties of the language by making predictions. We provide the competition results on the GitHub[1] which was implemented with DistillBERT. The final implementation with the higher accuracy and comparative analysis on different models is available in the GitHub[2].

Our rigorous experiments on a dataset and with various models have resulted in the following observations in addition to designing the system.

- Observation 1: Classifiers with no boosting, oversampling, or undersampling gave lower recall with a lower false positive rate (FPR). Without techniques like boosting, oversampling, or undersampling, a classifier tends to be biased toward the majority class. For example, after applying these techniques and masking, we get 66% accuracy for the XLM-RoBERTa-base, which was previously 41.45% on the XLM-RoBERTa-base.The classifier is conservative when clas-

[1] https://github.com/blp-workshop/blp_task2#leaderboard

[2] https://github.com/sharunakter/BLPWorkshop_2023_SentimentAnalysisInBangla

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 293–299

Figure 1: Data Distribution for Different Categories

sifying instances as the minority class. It generates too many false positive predictions (i.e., predicting the minority class when it is the majority class), which keeps the FPR low. Moreover, oversampling with boosting combats the data skew for all the models.

- Observation 2: XAI on XLM-RoBERTa-base's output shows how the MLM approach captures the nuanced sentiment expressed in Bangla text, even in the presence of code-mixing, sarcasm, or subtle linguistic cues. By understanding sentiment polarity and the context in which sentiments are expressed, it is possible to gain a deeper understanding. The randomly masked tokens were replaced with the special [MASK] token, creating partially masked sequences. The partially masked sequences were fed into the pre-trained XLM-RoBERTa-base model, which has been fine-tuned for sentiment analysis and language understanding tasks.

2 Background

2.1 Dataset Description

The dataset contains tweets or news-related public comments (Hasan et al., 2023a) to identify multi-class classification. Bangla data on various topics, such as political issues, incidents, COVID-19 facts, and country news from various online sources, are manually collected. The distribution of three classification labels, "negative," "positive," and "neutral," for training, dev, and test datasets are "19612", "17090", and "9205" respectively with a total datapoints of 45907.

2.2 Related Work

Interpreting implicit and underspecified phrases in instructional texts is vital to elicit plausible clarification and understanding (Roth et al., 2022; Islam et al., 2021). Researchers are increasingly focusing on sentiment analysis for low-resource languages like Bangla using traditional supervised machine learning such as multinomial Naïve Bayes (Sharif et al., 2019), SVM, Random forest and decision tree, and deep learning approaches such as deep recurrent neural network (Hassan et al., 2016), Glove word embedding with convolutional neural network (Mahmud et al., 2022), transfer learning using multilingual BERT (M-BERT) (Islam et al., 2020), transformer-based approach (Bhowmick and Jana, 2021; Hasan et al., 2023b). The lack of sufficient labeled data and domain and gender agnostic data limit the performance of those approaches (Islam et al., 2023a). Considering the scarcity of annotated data and the problem of predicting the lexical complexity of single-word and multi-word expressions, (Taya et al., 2021) used an ensemble model over a set of transformer-based model with hand-crafted features to increase the model generalization and robustness. To improve the quality of the sentiment analysis task of low-resource languages such as Bangla, the authors (Rahman and Kumar Dey, 2018; Sultana et al., 2022) proposed aspect-based sentiment analysis using BOW and supervised machine learning techniques and provided two datasets for aspect-based sentiment analysis. Many researchers claimed that transfer learning with adaptive pre-training effectively improves sentiment prediction tasks in low-resource languages by selecting appropriate source languages (Wang et al., 2023). Candidate source language selection through forward and backward strategies will increase compute requirements. To discover the effectiveness of semantic and syntactic parsing and the effects of subjective aspects on sentiment analysis, the authors at (Morio et al., 2022) performed a graph-based and seq2seq-based analysis with the help of a pre-trained language model and discovered that both research approaches perform well in extracting structured sentiment.

Considering the challenges for the Bangla dataset, the sentiGold (Islam et al., 2023a) developed a comprehensive dataset for sentiment analysis and provides a word embedding method, BanglaBERT, which performs well on formal Bangla text. However, the performance degrades for controversial text because of the need to be trained on informal data.

3 System Overview: Experiment and Setup

This section describes our data preprocessing steps for traditional machine learning models, vanilla deep learning models, and transformers. Next, we discuss the training and hyperparameter tuning of each model group.

3.1 Preprocessing and Data Augmentation

Bangla sentiment annotation is a challenging task because of its diversified syntaxes. Our task is to detect sentiment with three polarities: positive, negative, and neutral. We filtered out duplicate text if structural and semantic similarity were high (Islam et al., 2023b). Several syntaxes have been removed from the text, including punctuation marks, links, emoji, hashtag signs, and usernames (Mukta et al., 2021). We removed all non-Bangla characters and stop words and implemented Porterstemmer (Budiasih et al., 2009) to identify the root words and suffixes. Following preprocessing, boosting is applied with oversampling. There is a lack of balance in the class distribution of the Bangla dataset provided. Therefore, to balance the class distribution, we used oversampling techniques (Tahir et al., 2023) on the dataset. We merged the train and dev-test set to train the model. We applied the upsampling technique to the combined dataset with a ratio of 1.0 for the negative class.

3.2 Training and Hyperparameter Tuning

We used an 80-20 training-validation split for training all the classifiers: complex deep learning models, pre-trained transformers, and traditional machine learning algorithms.

Deep Learning and MLM: We experimented with following vanilla deep learning models: LSTM (Bhowmik et al., 2022), LSTM CNN (Chowdhury et al., 2022) and pretrained transformer models such as multilingual-BERT (M-BERT) (Tarannum et al., 2022), XLM-RoBERTa-base (Singh et al., 2022), DistilBERT (Suri, 2022; Fröbe et al., 2023), BanglaBERT (Sarker, 2020).

After the first round of analysis, we continue with both multilingual models BERT and XLM-RoBERTa-base and train our datasets with rigorous hyperparameter tuning and masked language modeling. The number of parameters and network size are responsible for the computation time and performance of the learning.

The number of labels determines the size of the last fully-connected dense layer. To predict the likelihood of the label, softmax activation with sparse categorical cross-entropy is applied on top of the model. The total parameter size for XLM-RoBERTa-base was 278045955, which took approximately 2 hours to complete the training on 8 GB RAM. We use a transformer toolkit for transfer learning in Bangla language (Hasan et al., 2019). The hyperparameters for hidden and feed-forward sizes are 768 and 3072, with 12 heads and 12 transformer blocks, regularized by a dropout of 10%, and the vocabulary size is 250002. XLM-RoBERTa-base model and other transformer models were fine-tuned with a batch size ranging from [16, 32], learning rate (Adam) range [3e-5, 2e-5], and number of epochs is 3. Tokenizer tools in the Huggingface (Zhang et al., 2019) repository were used to tokenize and preprocess the dataset.

For LSTM training, the parameters are maximum features = 500, embedding_dimension = 128, input length = 300, vocabulary size = 5000, and learning rate 0.01 with a decay value 1e-6. For 3 class labels, the batch size is 64, and the epoch number is 50. Additionally, there is one dense layer for sequential learning, 2 units of 1D MaxPooling layers, and a dropout of 0.2. Relu and Softmax were used for embedding. We used Adam optimization and sparse categorical cross-entropy as loss function. Table 1 reports the output for evaluation metrics and individual class labels on the test dataset for all deep learning models.

Traditional Machine Learning Models: We experimented with traditional approaches such as (I) Linear Regression (LR), (ii) Decision Tree (DT), (iii) Random Forest (RF), (iv) Multinomial Naïve Bayes (MNB), (v) K-Nearest Neighbour (KNN), (vi) Support Vector Machine (SVM) (Sazzed, 2021) and (vii) Stochastic Gradient Descent (SGD). We first transformed the preprocessed data into TF-IDF vectors with weighted n-gram (unigram, bigram, and trigram) to use contextual information. Table 2 reports the output for the traditional machine learning models.

4 Evaluations and Discussion on Results

In the original competition, we generated the results using a basic DistillBERT model without any preprocessing and fine-tuning. DistillBERT can process maximum 10k data - even batch-

Table 1: Evaluation of Top Deep Learning Models based on Individual Class Labels

Class Label	Model	Accuracy	Precision	Recall	F1	Micro F1	Macro F1
Negative	LSTM	0.68	0.70	0.64	0.67	0.68	0.62
Neutral			0.70	0.78	0.74		
Positive			0.63	0.63	0.63		
Negative	XLM-RoBERTa-base	0.66	0.71	0.76	0.73	0.65	0.58
Neutral			0.51	0.26	0.34		
Positive			0.62	0.74	0.67		
Negative	BanglaBERT	0.64	0.71	0.72	0.71	0.64	0.59
Neutral			0.44	0.38	0.41		
Positive			0.63	0.67	0.65		
Negative	Multilingual BERT	0.64	0.68	0.77	0.72	0.64	0.57
Neutral			0.46	0.29	0.36		
Positive			0.65	0.66	0.66		
Negative	DistilBERT	0.55	0.54	0.54	0.54	0.55	0.51
Neutral			0.60	0.64	0.61		
Positive			0.20	0.33	0.24		

Table 2: Evaluation Metrics: Traditional ML Models

Traditional Models	Accuracy	Precision	Recall	F1-Score
LR	71.91	72.54	71.91	71.52
DT	64.81	64.31	64.81	64.18
RF	72.66	73.55	72.66	72.00
MNB	71.22	72.51	71.22	70.83
KNN	53.69	54.79	53.69	53.64
SVM	75.02	75.26	75.02	74.85
SGD	60.40	65.94	60.40	58.69

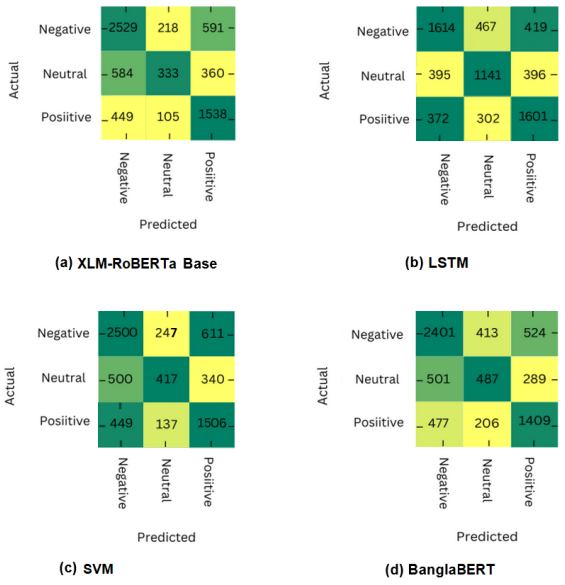


Figure 2: Confusion Matrix for Deep Learning Models

wise processing and averaging the output scores couldn't give a good result. We improved with a rigorous comparative analysis with vanilla deep learning, transformer-based LLMs, and traditional machine learning models that can handle large datasets. SVM achieved the highest accuracy and F1-score of 75.02% and 74.85% (Table 2). Unlike transformer-based models, LSTM and traditional models require extensive preprocessing, data cleaning, and oversampling. Moreover, upsampling and boosting improves all of the models. For example, before oversampling and boosting, XLM-RoBERTa-base reported 41% accuracy, where it improved to 66% after applying them (Table 1).

XLM-RoBERTa-base better predicts actual positive labels (Figure 2). However, it reports higher false negative (FN) values for negative classes and more false positive (FP) values for positive classes. In contrast, BanglaBERT reports fewer FP and FN values for each class but fails to predict TP with about 103 data points deviation. Therefore, XLM-RoBERTa-base and BanglaBERT performed well on the test dataset. FP and FN for each class in LSTM made minimal impact on accuracy because their values show a slightly equal distribution. Though LSTM generates better accuracy than the transformer model, transformers produce more correct instances for negative and other classes. In Figure 4, the learning curve backs up the finding of the unstable nature of the LSTM model, showing how it is underfitting. We saw the similar pattern for traditional ML models such as SVM. Therefore, models like LSTM and SVM may not generalize to another dataset with new

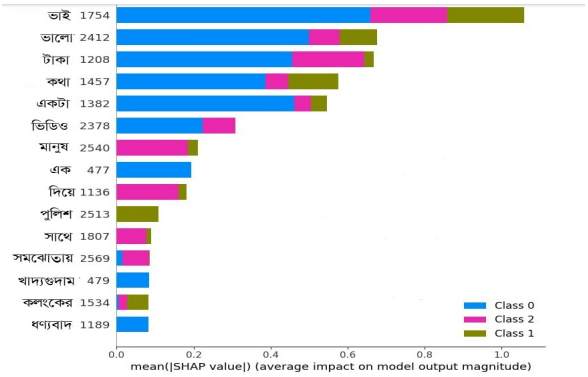


Figure 3: SHAP on XLM-RoBERTa-base output- blue (Positive), green (Negative) and pink (Neutral)

Figure 4: Learning Curve for Underfit LSTM

test instances. Since the class distribution is imbalanced in the dataset, we also calculated other metrics such as F1-score, precision, and recall, which basically signifies if the model is doing a better job for lowered-numbered classes. For example, none of the models did a great job with the "neutral" class showing a lower f1-score, precision, and recall, which also syncs with the confusion matrix.

We used SHAP (Lundberg and Lee, 2017) - a state-of-the-art explainable AI (XAI) tool, to interpret the classification results of the XLM-RoBERTa-base transformer model's output, in our case "accuracy" (Figure 3). This SHAP plot combines the significance of the features with their impacts. The Y-axis lists the features from top to bottom or most important to least important order. The labels on the Y-axis represent the most influential word features for XLM-RoBERTa-base and their associated indexing in the word vector. The x-axis shows the Shapely values from 0 to 1. Blue, green, and pink spectrum are representations of Shapley values for "positive," "negative," and "neutral" classes. Not only the length of the spectrum but also the color has significance. For example, the "পুলিশ" feature correlates less than 20% with the model output accuracy. However, this word influences a post's identification as only negative (green color). Another good example is the "ভাই" feature, the most influential feature in the predictions with XLM-RoBERTa-base. The Shapely value for blue (positive) is 70%, whereas for pink (neutral) and green (negative) is 20%. That means having a "ভাই" word in a post mostly co-related to a positive post, which is also intuitively correct since it is a respectful salutation. The Shapely values of the features are more positively correlated with the positive class (labeled with blue) since blue spectrums are larger than the others. The neutral class (labeled with pink) has the lowest correlation with the model output. This result also aligns with the confusion matrix (Figure 3), where prediction accuracy for positive classes is higher with XLM-RoBERTa. Therefore, the positive class operated on a higher accuracy scale with a higher correlation of approximately 70% with the most influential feature (feature 1754). The plot also shows that the impact of the "negative class" is very low- it does not frequently appear as the positive or neutral classes.

5 Limitations and Conclusion

In summary, we compared multiple ML approaches to discuss the multi-class sentiment analysis. We analyzed and compared results based on preprocessing techniques, rigorous output analysis, and XAI. Our analysis shows that the XLM-RoBERTa-base generates a stable model even with lower accuracy regarding confusion matrix, evaluation metrics, and XAI than LSTM and traditional models. The first challenge we faced is that vector assembler on huge data made the dimensions of the feature very large and computationally expensive, difficult to address with low computing resources. Secondly, the highly imbalanced dataset has only 20% "Neutral" labels, which skewed the prediction against this class and caused some models to underfit. Developing MLM-based masked models with oversampled datasets improved the quality of the classification tasks for XLM-RoBERTa. It understands contextual relationships between words better and effectively predicts missing or masked words within a sentence. Our future work will focus on mitigating the challenge of Bangla sentiment analysis for lacking high-quality datasets, generalizable tools, comprehensive sentiment lexicons, and standardized evaluation metrics.

Ethics Statement

All the authors are trained in the ethical conduct of research. Ethical usage of data, analysis, writing, and transparency of implementation have been maintained by sharing the implementation.

References

Anirban Bhowmick and Abhik Jana. 2021. Sentiment analysis for Bengali using transformer based models. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 481–486, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Nitish Ranjan Bhowmik, Mohammad Arifuzzaman, and M Rubaiyat Hossain Mondal. 2022. Sentiment analysis on bangla text using extended lexicon dictionary and deep learning algorithms. *Array*, 13:100123.

Ni Nyoman Budiasih, TAB Wirayuda, and RN Dayawati. 2009. Analisis dan implementasi stemming teks berbahasa indonesia dengan menggunakan porter stemmer. *Tugas Akhir Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom: Bandung*.

Pallab Chowdhury, Ettilla Mohiuddin Eumi, Ovi Sarkar, and Md Faysal Ahamed. 2022. Bangla news classification using glove vectorization, lstm, and cnn. In *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021*, pages 723–731. Springer.

Maik Fröbe, Benno Stein, Tim Gollub, Matthias Hagen, and Martin Potthast. 2023. Semeval-2023 task 5: Clickbait spoiling. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2275–2286.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md Arid Hasan, Firoj Alam, Shammur Absar Chowdhury, and Naira Khan. 2019. Neural machine translation for the bangla-english language pair. In *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Asif Hassan, Mohammad Rashedul Amin, Abul Kalam Al Azad, and Nabeel Mohammed. 2016. Sentiment analysis on bangla and romanized bangla text using deep recurrent models. In *2016 International Workshop on Computational Intelligence (IWCI)*, pages 51–56.

Khondoker Ittehadul Islam, Md Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Md. Ekramul Islam, Labib Chowdhury, Faisal Ahamed Khan, Shazzad Hossain, Md Sourave Hossain, Mohammad Mamun Or Rashid, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Sentigold: A large bangla gold standard multi-domain sentiment analysis dataset and its evaluation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 42074218, New York, NY, USA. Association for Computing Machinery.

Md Ekramul Islam, Labib Chowdhury, Faisal Ahamed Khan, Shazzad Hossain, Md Sourave Hossain, Mohammad Mamun Or Rashid, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023b. Sentigold: A large bangla gold standard multi-domain sentiment analysis dataset and its evaluation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4207–4218.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Md. Shihab Mahmud, Md. Touhidul Islam, Afrin Jaman Bonny, Rokeya Khatun Shorna, Jasia Hossain Omi, and Md. Sadekur Rahman. 2022. Deep learning based sentiment analysis from bangla text using glove word embedding along with convolutional neural network. In *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

Gaku Morio, Hiroaki Ozaki, Atsuki Yamaguchi, and Yasuhiro Sogawa. 2022. Hitachi at SemEval-2022

task 10: Comparing graph- and Seq2Seq-based models highlights difficulty in structured sentiment analysis. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1349–1359, Seattle, United States. Association for Computational Linguistics.

Md Saddam Hossain Mukta, Md Adnanul Islam, Faisal Ahamed Khan, Afjal Hossain, Shuvanon Razik, Shazzad Hossain, and Jalal Mahmud. 2021. A comprehensive guideline for bengali sentiment annotation. *Transactions on Asian and Low-Resource Language Information Processing*, 21(2):1–19.

Md. Atikur Rahman and Emon Kumar Dey. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2).

Michael Roth, Talita Anthonio, and Anna Sauer. 2022. SemEval-2022 task 7: Identifying plausible clarifications of implicit and underspecified phrases in instructional texts. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1039–1049, Seattle, United States. Association for Computational Linguistics.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Salim Sazzed. 2021. Abusive content detection in transliterated bengali-english social media corpus. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 125–130.

Omar Sharif, Mohammed Moshiul Hoque, and Eftekhar Hossain. 2019. Sentiment analysis of bengali texts on online restaurant reviews using multinomial naïve bayes. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–6.

Sumit Singh, Pawankumar Jawale, and Uma Tiwary. 2022. silpa_nlp at semeval-2022 tasks 11: Transformer based ner models for hindi and bangla languages. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1536–1542.

Nasrin Sultana, Rehena Sultana, Risul Islam Rasel, and Mohammed Moshiul Hoque. 2022. Aspect-based sentiment analysis of bangla comments on entertainment domain. In *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pages 953–958.

Manan Suri. 2022. Pickle at semeval-2022 task 4: Boosting pre-trained language models with task specific metadata and cost sensitive learning. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 464–472.

Maryam Tahir, Ahmad Naeem, Hassaan Malik, Jawad Tanveer, Rizwan Ali Naqvi, and Seung-Won Lee. 2023. Dscc_net: Multi-classification deep learning models for diagnosing of skin cancer using dermoscopic images. *Cancers*, 15(7):2179.

Prerona Tarannum, Firoj Alam, Md Arid Hasan, and Sheak Rashed Haider Noori. 2022. Z-index at checkthat! lab 2022: Check-worthiness identification on tweet text. *arXiv preprint arXiv:2207.07308*.

Yuki Taya, Lis Kanashiro Pereira, Fei Cheng, and Ichiro Kobayashi. 2021. OCHADAI-KYOTO at SemEval-2021 task 1: Enhancing model generalization and robustness for lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 17–23, Online. Association for Computational Linguistics.

Ming Wang, Heike Adel, Lukas Lange, Jannik Strotgen, and Hinrich Schütze. 2023. Nlnde at semeval-2023 task 12: Adaptive pretraining and source language selection for low-resource multilingual sentiment analysis. *ArXiv*, abs/2305.00090.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

EmptyMind at BLP-2023 Task 2: Sentiment Analysis of Bangla Social Media Posts using Transformer-Based Models

Karnis Fatema, Udoy Das, Md. Ayon Mia, Md Sajidul Mowla, Mahshar Yahan, MD Fayez Ullah, Arpita Sarker, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
u1804{052, 109, 128, 100, 007, 094, 099}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

With the popularity of social media platforms, people are sharing their individual thoughts by posting, commenting, and messaging with their friends, which generates a significant amount of digital text data every day. Conducting sentiment analysis of social media content is a vibrant research domain within the realm of Natural Language Processing (NLP), and it has practical, real-world uses. Numerous prior studies have focused on sentiment analysis for languages that have abundant linguistic resources, such as English. However, limited prior research works have been done for automatic sentiment analysis in low-resource languages like Bangla. In this research work, we are going to finetune different transformer-based models for Bangla sentiment analysis. To train and evaluate the model, we have utilized a dataset provided in a shared task organized by the BLP Workshop co-located with EMNLP-2023. Moreover, we have conducted a comparative study among different machine learning models, deep learning models, and transformer-based models for Bangla sentiment analysis. Our findings show that the BanglaBERT (Large) model has achieved the best result with a micro F1-Score of 0.7109 and secured 7^{th} position in the shared task 2 leaderboard of the BLP Workshop in EMNLP 2023.

1 Introduction

Nowadays, social media platforms produce a large amount of text data by posting, commenting, and messaging. Finding the sentiment of social media data is an active research area among practitioners due to its numerous practical applications. However, conducting sentiment analysis on social media data is a difficult task due to the natural variation of writing patterns among users.

A significant amount of effort has been devoted to analyzing sentiment in social media data for resourced enriched languages like English (Babu and Kanaga, 2022). However, we have found a limited number of relevant studies focused on sentiment analysis in the Bangla language due to the lack of a standardized annotated dataset of Bangla text sourced from social media platforms (Pran et al., 2020).

The main objective of this research work is to analyze sentiment on Bangla social media posts. Moreover, we conduct a comparative analysis among different ML, DL, and transformer-based models for Bangla sentiment analysis. To train and evaluate different models, we have utilized a dataset provided in a shared task named Sentiment Analysis of Bangla Social Media Posts organized by the First Workshop on Bangla Language Processing co-located with EMNLP-2023 (Hasan et al., 2023a,b; Islam et al., 2021).

Various ML models and DL models have been deployed for Bangla sentiment analysis. We have utilized three popular transformer-based model architectures named Bangla BERT Base, BanglaBERT, and BanglaBERT (Large) for the sentiment analysis model.

Among the machine learning models, SVM utilizing TF-IDF yields the best performance, achieving a micro F1-Score of 0.57. In the realm of deep learning, the BiLSTM + CNN model with Word2Vec attains the highest micro F1-Score at 0.61. The transformer-based BanglaBERT (Large) models (Bhattacharjee et al., 2022) outperform the rest, achieving an impressive micro F1-Score of 0.7109.

The main contributions of our research works are as follows -

- We have finetuned the transformer-based BanglaBERT and BanglaBERT (Large) models for Bangla sentiment analysis.
- We have conducted a comparative analysis among different ML, DL, and transformer-based models for sentiment analysis in the

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 300–304

Bangla language.

The implementation of our research work has been shared in the following GitHub repository - https://github.com/ML-EmptyMind/blp-task2.

2 Related Work

We divide all the previous works related to sentiment analysis into three different categories: ML approaches, DL approaches, and transformer-based approaches.

Machine learning techniques like SVM, Multinomial Naive Bayes, KNN, Logistic Regression, Decision Trees, and Random Forests are used for sentiment analysis. Among these, SVM and Multinomial Naive Bayes classifiers (Hassan et al., 2022) have demonstrated the best performance, with SVM achieving the highest accuracy scores. The dataset is subsequently transformed using a TF-IDF Vectorizer, and SVM is used as the classifier for data classification (Arafin Mahtab et al., 2018).

Numerous deep-learning techniques are employed for sentiment analysis as well. RNN with LSTM model is used (Wahid et al., 2019) for sentiment analysis to classify and categorize the sentiments of social media posts about cricket as positive, negative, or neutral. In order to analyze sentiment or opinion in Bangla, the attention mechanism is suggested (Sharmin and Chakma, 2020) in the study. It examines the difficulties with sentiment analysis and evaluation, particularly in the Bangla language.

Alongside a deep learning model that utilizes multilingual BERT and transfer learning, the research incorporates datasets for two-class and three-class sentiment analysis that have been manually annotated in Bangla, as mentioned in (Islam et al., 2020). This model surpasses the current state-of-the-art algorithms in terms of accuracy, attaining a 71% accuracy rate for two-class sentiment classification and a 60% accuracy rate for three-class sentiment classification. The approach is also used to examine the tone of reader comments in an online daily newspaper, demonstrating that while comments on religious articles tend to be more positive than those on political or sports news, the former are more numerous for those topics.

The objectives of the study include finetuning the transformer-based models for Bangla sentiment analysis and providing a comparison analysis with the baseline models using ML and DL approaches.

3 Dataset

During our research work, we have capitalized the dataset provided using the shared task 2 (Sentiment Analysis of Bangla Social Media Posts) organized by the BLP Workshop @ EMNLP 2023 (Hasan et al., 2023a). The dataset used for this shared task consists of MUltiplatform BAngla SEntiment (MUBASE) (Hasan et al., 2023b) and SentNoB (Islam et al., 2021). The MUBASE dataset is a cross-platform collection of Facebook and Twitter posts that has been manually annotated with sentiment polarity. The SentNob dataset comprises user comments sourced from social media platforms in response to news articles and videos. The dataset covers several fields, such as politics, education, and agriculture. The provided dataset has three sentiment categories: Positive, Negative, and Neutral. This dataset has train, dev, and test split containing 35266, 3934, and 6707 texts respectively. In Table 1, statistics about the dataset are given with class-wise samples.

Classes	Train	Dev	Test
Positive	12,364	1,388	2,092
Negative	15,767	1,753	3,338
Neutral	7,135	793	1,277
Total	35,266	3,934	6,707

Table 1: Class-wise distribution of sentiment analysis dataset

The provided dataset contains URLs, emojis, and other symbols which are removed in the preprocessing step.

4 Methodology

In this section, we outline the methodology of our research. We establish baseline models by employing both ML and DL techniques. Subsequently, we enhance performance by incorporating a transformer-based model. Figure 1 shows an overview of our methodology.

Machine Learning Models

For machine learning algorithms, Word2Vec and TF-IDF word embeddings have been applied to extract the feature vector (Mikolov et al., 2013). Word2Vec embedding has been implemented with

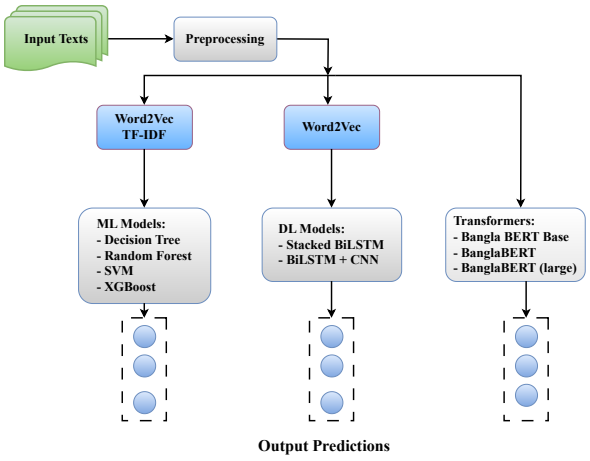


Figure 1: Conceptual process of sentiment analysis

a 100-dimension vector for each word in the vocabulary. We have explored different ML algorithms to set a baseline. We have trained the Decision Tree, Random Forest, SVM, and XGBoost models. Further, all the aforementioned algorithms have been investigated for both TF-IDF and Word2Vec word embedding.

Deep Learning Models

Three different DL models such as Stack of Bidirectional LSTM (BiLSTM), BiLSTM with CNN and dropout, and BiLSTM with CNN for Word2Vec embedding have been explored for Bangla sentiment analysis. We have padded the tokenized dataset by setting the max length value of text as 400 for three models. For the stacked BiLSTM model three consecutive layers of BiLSTM are stacked with 32, 16, and 8 neurons respectively.

Transformer Models

In recent ages, transformer models have gained popularity for their tremendous performance in NLP tasks. We use Bangla BERT Base (Sarker, 2020), BanglaBERT(Bhattacharjee et al., 2022), and BanglaBERT (large)(Bhattacharjee et al., 2022) pre-trained model to fine-tune on Bangla sentiment analysis dataset. The above three pre-trained models have been trained on the Bangla natural language dataset. Bangla BERT Base is a Bangla sentencepiece model containing vocab size 102025. BanglaBERT and BanglaBERT (large) are ELECTRA discriminator models that are pre-trained with the Replaced Token Detection (RTD) objective. Fine-tuned BanglaBERT (large) gives the best result for our case. We have used a pre-trained tokenizer and tokenized sample using 512 as the maximum length of the text. For training purposes, we have taken the help of trainer API.

5 Results and Analysis

In this section, we provide the outcomes obtained from our experimentation.

5.1 Parameter Settings

All parameters are kept identical for the TF-IDF and Word2Vec embedding. For random forest, we have selected *n_estimator* value 40. While training the SVM model, we have picked out *C* as 2 and kernel *rbf*. Lastly, *n_estimator* value 40 is chosen for XGBoost.

We have set *epochs* to value 30, *batch size* to value 32, *verbose* to value 1 along with callback having an accuracy threshold value of 0.99 for BiLSTM with CNN model which uses Word2Vec embedding. For all DL models, we have set *learning rate* as 0.001, *adam* as the optimizer, and *sparse_categorical_crossentropy* as a loss function. We have further investigated the DL model varying *epochs*, *batch size*, and *learning rate* to validate the consequence on the performance.

In our best performing transformer model, we set the 0.00005 as *learning rate*, 0.01 as *weight decay*, 0.1 as *warm-up ratio*, learning rate scheduler type to *linear*, 3 as training *epochs*, training *batch size* as 16, 2 as *radient accumulation steps* and *adafactor* as the optimizer. Moreover, we set the *dropout rate* to 50% to get the best result. We have evaluated the performance of the BanglaBERT (Large) model, training it without dropout and setting the learning rate to 0.01 for just 3 epochs, which has provided an F1 score of 0.7001. In another setup, we have introduced a 50% dropout rate and extended the training to 4 epochs, which has shown an F1 score of 0.7026.

5.2 Evaluation Metrics

We have applied micro F1-Score evaluation metrics according to guidelines set up by the organizer. Moreover, we also have evaluated precision and recall for all models.

5.3 Comparative Analysis

The performance of each model tested on the evaluation set is displayed in Table 2. We have determined the best-performing model based on the F1-score.

Approach	Classifier	Average		
		P	**R**	**F1**
ML	Decision Tree (TF-IDF)	0.48	0.48	0.48
	Random Forest (TF-IDF)	0.53	0.56	0.56
	SVM (TF-IDF)	0.54	0.57	0.57
	XGBoost (TF-IDF)	0.51	0.53	0.53
	Decision Tree (Word2Vec)	0.45	0.45	0.45
	Random Forest (Word2Vec)	0.50	0.52	0.52
	SVM (Word2Vec)	0.40	0.50	0.50
	XGBoost (Word2Vec)	0.50	0.52	0.52
DL	Stacked BiLSTM (Word2Vec)	0.57	0.57	0.57
	BiLSTM+ CNN (Word2Vec)	0.59	0.61	0.61
Transformer	Bangla BERT Base	0.63	0.63	0.63
	BanglaBERT	0.70	**0.71**	**0.71**
	BanglaBERT (large)	**0.71**	0.70	**0.7109**

Table 2: Performance of various systems on test set. Here P, R, and F1 denote weighted Precision, weighted Recall, and micro F1-Score respectively.

Among the ML models, SVM combined with TF-IDF word embedding has given the highest micro F1-score of 0.5688 while Decision Tree has provided a micro F1-score of 0.4839, Random Forest has shown an F1-score of 0.5555 and XGBoost has given an F1-score of 0.5264. In addition, using Word2Vec embedding, Decision Tree, Radom Forest, SVM and XGBoost model has given micro F1-score of 0.4471, 0.5167, 0.5008, and 0.5239 respectively.

The stacked BiLSTM model, which consists of an input layer with a text length of 400, a Word2Vec embedding layer, there BiLSTM layer, and finally one output layer, has provided a 0.5714 micro F1-score. The combination of BiLSTM along CNN has shown a micro F1-score of 0.6069 which surpasses all other DL and ML models in the evaluation.

Bangla BERT Base has provided a 0.63 micro F1-score which is better than the best performing DL model. In addition, BanglaBERT has shown a micro F1-score of 0.7100. Furthermore, BanglaBERT (large) pre-trained has archived the best score of 0.7109 for this task.

The findings suggest that the transformer-based models have delivered outstanding performance for the assigned task. By comparison, DL models have achieved better results than ML models. Moreover, in transformer-based models, BanglaBERT outperforms Bangla BERT Base. BanglaBERT (large) performs slightly better than BanglaBERT.

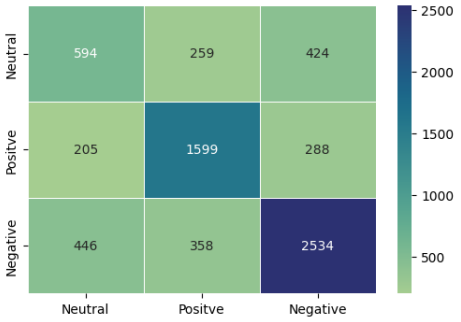


Figure 2: Confusion Matrix of best model

5.4 Error Analysis

Table 2 illustrates that BanglaBERT (large) has acquired the best performance for this task. An observational error analysis has been conducted for the best-performing model. From Figure 2, it has been observed that the model classifies 594 samples of Neutral class correctly and misclassifies 259 as Positive and 424 as negative. Furthermore, 446 samples of the negative class have been incorrectly classified as Neutral. The main reason behind this problem is due to the use of an imbalance dataset. Different size of text length has an impact on error. Sentences with just a few words are not classified correctly for all classes. In the case of neutral sentences, the model misclassifies as negative and for negative sentences model predicts as neutral on a large scale due to a rich set of inflections in the Bangla language, unable to capture all subword information.

6 Conclusion

In this research work, we have explored various transformer-based models for analyzing sentiment in the Bangla language. To train and evaluate different models, we have employed a dataset made available through the BLP Workshop in conjunction with EMNLP-2023. Additionally, we have conducted a comprehensive comparison among different ML, DL, and transformer-based approaches for Bangla sentiment analysis. We have found that the BanglaBERT (Large) model has outperformed the others, achieving the highest micro F1-Score of 0.7109.

In the future, we intend to investigate various architectures and employ ensemble methods to enhance model performance. Additionally, we will apply different techniques to address issues arising from the use of an imbalanced dataset.

Limitations

We have explored only 100-dimensional word embedding for ML and DL models. Other word embedding techniques and hyper-parameter tuning should be further analyzed. Hyper-parameter setting for the BERT model should be an option to investigate beyond. Removal of the impact of text length variation must be addressed.

Ethics Statement

In this paper, we have experimented with different models and techniques that have been ethically implemented. Our aim is to develop a system that finds the sentiment of Bangla text for the betterment of our society and culture. Moreover, we have shared the implementation details in a GitHub repository for reproducibility.

References

Shamsul Arafin Mahtab, Nazmul Islam, and Md Mahfuzur Rahaman. 2018. Sentiment analysis on bangladesh cricket with support vector machine. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4.

Nirmal Varghese Babu and E Grace Mary Kanaga. 2022. Sentiment analysis in social media data for depression detection using artificial intelligence: a review. *SN Computer Science*, 3:1–20.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Mahmudul Hassan, Shahriar Shakil, Nazmun Nessa Moon, Mohammad Monirul Islam, Refath Ara Hossain, Asma Mariam, and Fernaz Narin Nur. 2022. Sentiment analysis on bangla conversation using machine learning approach. *International Journal of Electrical and Computer Engineering (IJECE)*, 12(5):5562–5572.

Khondoker Ittehadul Islam, Md. Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.

Md. Sabbir Alam Pran, Md. Rafiuzzaman Bhuiyan, Syed Akhter Hossain, and Sheikh Abujar. 2020. Analysis of bangladeshi people's emotion during covid-19 in social media using deep learning. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Sadia Sharmin and Danial Chakma. 2020. Attention-based convolutional neural network for bangla sentiment analysis. *AI & SOCIETY*, 36:381 – 396.

Md. Ferdous Wahid, Md. Jahid Hasan, and Md. Shahin Alom. 2019. Cricket sentiment analysis from bangla text using recurrent neural network with long short term memory model. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4.

RSM-NLP at BLP-2023 Task 2: Bangla Sentiment Analysis using Weighted and Majority Voted Fine-Tuned Transformers

Pratinav Seth[*] **Rashi Goel**[†] **Komal Mathur**[† ‡] **Swetha Vemulapalli**[§ ‡]
Manipal Institute of Technology
Manipal Academy of Higher Education, Manipal, India
{seth.pratinav,rashigoel2017,komlixmathur,swetha.vemulapalli.3}@gmail.com

Abstract

This paper describes our approach to submissions made at Shared Task 2 at BLP Workshop - Sentiment Analysis of Bangla Social Media Posts(Hasan et al., 2023a; Islam et al., 2021; Hasan et al., 2023b). Sentiment Analysis is an action research area in the digital age. With the rapid and constant growth of online social media sites and services and the increasing amount of textual data, the application of automatic Sentiment Analysis is on the rise. However, most of the research in this domain is based on the English language. Despite being the world's sixth most widely spoken language, little work has been done in Bangla. This task aims to promote work on Bangla Sentiment Analysis while identifying the polarity of social media content by determining whether the sentiment expressed in the text is Positive, Negative, or Neutral. Our approach consists of experimenting and finetuning various multilingual and pre-trained BERT-based models on our downstream tasks and using a Majority Voting and Weighted ensemble model that outperforms individual baseline model scores. Our system scored 0.711 for the multiclass classification task and scored 10th place among the participants on the leaderboard for the shared task. Our code is available at https://github.com/ptnv-s/RSM-NLP-BLP-Task2

1 Introduction

In the era of a high influx of social media platforms, blogs, and online reviews, sentiment analysis has become the need of the hour. Also known as opinion mining, sentiment analysis is a computational linguistic task that is aimed at determining whether a text contains a positive, negative, or neutral sentiment behind it (Khan et al., 2020)

Text	Label
ভাই সোনাই ঘোষ এর দই খেয়ে যাইতেন , খুব ই মজার	Positive
এখানে আরো ভালো ভাবে দলীয় ও র এর অবস্থান পাকা হলো কি ? ?	Neutral
শাউয়ার মাগি তরে এত রিপোর্ট মারি তাও আসে কেন ভিডিও	Negative

Table 1: Text Samples from the Training dataset, with labels as either Positive, Neutral or Negative

Sentiment analysis has diverse uses, including preventing adolescent suicide by detecting cyberbullying and mitigating unjust actions that target specific communities through hate speech detection, among numerous other applications (Islam et al., 2020). Approximately 284.3 million people worldwide speak Bangla as their primary language. Individuals speaking Bangla increasingly engage in social media platforms like Instagram, Facebook, Reddit, and Twitter and express opinions on microblogging platforms, commenting on news portals and online shopping. However, analyzing vast volumes of rapidly generated data in the digital age is a very tedious job to do. This is where sentiment analysis can be applied (Hassan et al., 2016). Most sentiment analysis research predominantly focuses on English, leaving Bangla Sentiment analysis in its nascent stages. Recently, some works have addressed this issue. However, none of these studies have fully embraced the different perspectives of Bangla.

To address this problem, we present our contributions to Shared Task 2 at BLP Workshop - Sentiment Analysis of Bangla Social Media Posts. This task aims to detect the polarity associated with a given social media text. This multiclass classification task involves determining whether the sentiment expressed in the text is Positive, Negative, or Neutral. For this problem statement, we have conducted various experiments using multi-

[*] Dept. of Data Science & Computer Applications
[†] Dept. of Computer Science & Engineering
[‡] Authors have contributed equally to this work
[§] Dept. of Information and Communication Technology

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 305–311

lingual berts (Bhattacharjee et al., 2022; Sanh et al., 2019a; Das et al., 2022; Sarker, 2020) and various pre-trained transformers (Liu et al., 2019a) by fine-tuning them on downstream tasks. We also apply Majority Voting and Weighted ensembling on the top-k models to show how these methods affect the models' performance and how an ensemble of these models performs better than the individual baselines.

2 Background

2.1 Problem and Data Description

The EMNLP 2023 Bangla Workshop Task 2: Sentiment Analysis of Bangla Social Media Posts (Hasan et al., 2023a; Islam et al., 2021; Hasan et al., 2023b) aims to detect the polarity of the sentiment associated with a given text extracted from social media. From the entire set of labels, over 14,000 were classified as negative, approximately 12,000 as positive, and roughly 6,000 as neutral, as indicated in the distribution chart in Figure 1 and a few samples of the Dataset are shown in Table 1. The dataset includes the MUltiplatform BAngla SEntiment (MUBASE) dataset and the SentNob dataset (Islam et al., 2021). SentNob comprises public comments from social media on news and videos across 13 domains, such as agriculture, politics, and education. It is manually annotated with a moderate agreement score of 0.53. On the other hand, MUBASE is a sizable compilation of multi-platform data, including Facebook posts and tweets, each manually tagged for sentiment polarity. These datasets provide a comprehensive and diverse landscape for studying Bangla sentiment analysis.

2.2 Previous Works

2.2.1 Sentiment Analysis

Sentiment analysis is an NLP task that uses computational methods to determine and extract the emotional tone expressed in a piece of text (Hogenboom et al., 2014). There are several different approaches to sentiment analysis. Early sentiment analysis approaches primarily employed rule-based methods and lexicon-based techniques (Obaidat et al., 2015) to determine the sentiment context of texts. One of the significant areas of application of Sentiment Analysis is in Social Media Posts as in (Tang et al., 2014) and (Taboada et al., 2011), a sentiment lexicon with a linguistic rule-based approach was used to create a sentiment detection mechanism from tweets(Reckman et al., 2013). Following this, contemporary advancements have introduced machine learning and deep learning techniques that significantly boost accuracy by extracting intricate patterns from annotated datasets. Due to human language's complexity and sentiment expression nuances, it is a challenging task. The accuracy of the task may be improved by using larger datasets, more complex and fine-tuned models (Hassan et al., 2016), ensembling, etc. Modern approaches leverage large-scale Pre-trained Language Models (PLMs), such as Transformers, BERTs (Devlin et al., 2018), and NLUs (Bender and Koller, 2020), alongside refined fine-tuning mechanisms(Hasan et al., 2023b). They excel at capturing the intricate associations between words within the text and their corresponding polarity. In today's world, with the introduction of free-to-use models like ChatGPT, sentiment analysis has opened to new possibilities (Wang et al., 2023).

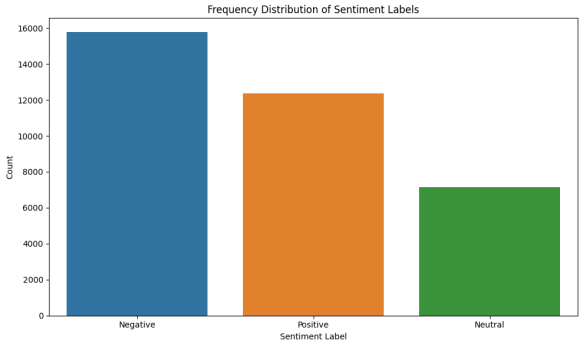


Figure 1: Frequency of Task 2 labels in training set

2.2.2 Bangla Language Processing

The Bangla language is the 7th most spoken language, with 265 million speakers worldwide (Sen et al., 2022). However, since English is the predominant language used for technical knowledge, journals, and documentation, many Bangla-speaking people face hurdles in utilizing these resources. Research on Bangla Natural Language Processing (BNLP) began in the early 1990s, focusing on rule-based lexical and morphological analysis (Alam et al., 2021). From the modeling perspective, most earlier endeavors are either rule-based, statistical, or classical machine learning-based approaches(Kudo and Matsumoto, 2001). As for the sequence tagging tasks, such as NER and G2P, the algorithms, including Hidden Markov Models (HMMs) (Brants, 2000), Conditional Random Fields (CRFs) (Lafferty et al., 2001), Maximum

Entropy (ME) (Ratnaparkhi, 1996) and Maximum Entropy Markov Models (MEMMs) (McCallum et al., 2000) have been used successfully. It is only very recently that a small number of studies have explored deep learning-based approaches. As depicted in (Alam et al., 2021), there has been significant work in resource and model development in Bangla sentiment analysis. In (Das and Bandyopadhyay, 2010), the authors proposed a computational technique of generating an equivalent SentiWordNet (Bangla) from publicly available English sentiment lexicons and an English-Bangla bilingual dictionary with few easily adaptable noise reduction techniques. However, with the Introduction of BERTs many works focused on fine-tuning multilingual BERTs (Ashrafi et al., 2020; Das et al., 2021), but BanglaBERT (Sarker, 2020) being the first model pre-trained on Bangla text corpus.

2.2.3 Bangla Sentiment Analysis

Sentiment analysis is a tool to extract the emotional tone of the text. It is used for cyberbullying detection, hate speech mitigation and market research. Bangla is the 7th most spoken language, and sentiment analysis for Bangla is still in its early stages. The first attempt to perform sentiment analysis in the context of Indian Languages, including Bangla, was done as recently as in 2015 (Patra et al., 2015). The lack of accurately annotated data is one of the biggest bottlenecks to advancing Bangla Sentiment Analysis. (Islam et al., 2021) and (Rahman et al., 2018) describe the creation of datasets for this purpose. A word2vec model was tuned with word co-occurrence scores for sentiment analysis in (Al-Amin et al., 2017), achieving an accuracy of 75.5%. In (Wahid et al., 2019), aspect-based sentiment analysis data was examined, boasting a remarkable 95% accuracy. However, challenges were encountered when rephrasing common and proper nouns in Bangla. Among most studies, however, transformer models have consistently outperformed other algorithms and models, inciting a significant amount of research into the area. In (Chowdhury et al., 2019), Opinion Mining was conducted on a dataset of 4,000 manually translated Bangla movie reviews, with the objective of classifying them as positive or negative. The LSTM approach had achieved an accuracy of 82.42%. A Bi-LSTM architecture was applied by (Sharfuddin et al., 2018) to a labeled dataset of 10,000 Facebook comments in Bangla, resulting in an accuracy of 85.67%. However, the study faced significant data preprocessing difficulties. In (Tripto and Ali, 2018), a combination of CNN and LSTM was employed to extract six distinct emotions from various types of Bangla YouTube video comments. The reported accuracies were 65.97% and 54.24% for three and five-label sentiment classification, respectively. A common issue faced by authors while using CNNs was that proper tuning between layers could not be achieved. In another study (Hossain et al., 2020), 1000 online restaurant reviews were collected from the Foodpanda website for performing SA and deployed, thus combining CNN with LSTM architecture with a 300 dimensional Word2Vec pretrained model having validation accuracy of 75.01%. (Rezaul Karim et al., 2020) developed a novel word embedding system for Bangla texts, BanglaFastText, incorporating it into a Multichannel Convolutional LSTM (MConv-LSTM). In (Islam et al., 2020) authors performed SA on 1002 public comments from newspapers with the help of the BERT pretrained model and achieved accuracy on GRU at 71% on 2 class sentiments. In (Hasan et al., 2020a), the performance of multiple classical machine learning algorithms and deep learning models were compared on several sentiment-labeled datasets, showing that pre-trained transformer models such as BERT and XLM-RoBERTa yielded the highest scores.

3 System Overview

We conducted extensive experiments for the given task involving Bangla Sentiment analysis. We fine-tuned various multilingual and pre-trained transformer architectures, including BERT (Kenton and Toutanova, 2019), DistillBERT (Sanh et al., 2019b), RoBERTa (Liu et al., 2019b), and Various Pre-Trained BERT models (Das et al., 2022; Sarker, 2020) on our downstream task of polarity classification. We shortlist the top-k model based on the performance metrics and ensemble the predictions using Majority Voted and Weighted Ensemble.

3.1 Fine-Tuning Transformers

We used multiple transformer architectures to observe the effect of the model architecture and the pre-trained dataset on the downstream task. For multiclass classification, we added a linear layer acting as a classification head to fine-tune the models for the multiclass classification.

We have used various models for our experiments, including **BERT** (Kenton and Toutanova,

Model	Acc.	Pre.	Rec.	F1
RoBERTa (Base)	0.550	0.544	0.550	0.550
Distill BERT	**0.701**	**0.687**	**0.701**	**0.701**
HF-PT BERT-1	0.672	0.679	0.672	0.672
HF-PT BERT-2	0.639	0.630	0.639	0.639
HF-PT BERT-3	0.669	0.671	0.669	0.669
Bangla BERT (Small)	0.657	0.649	0.657	0.657
Bangla BERT (Large)	0.693	0.684	0.693	0.693
Bangla BERT (Base)	**0.701**	**0.687**	**0.701**	**0.701**
Banglish BERT	0.684	0.672	0.684	0.684

Table 2: Results of Base-Models on Test-Set of Shared-Task Dataset where Acc. is Accuracy, Pre. is Precision, Rec. is Recall & F1 refers to F1-Score

2019), a transformer-based language model that creates representations of text by combining both left and right contexts with Masked Language Modeling and Next Sentence Prediction being pre-training tasks. **RoBERTa** (Liu et al., 2019b) is a faster variation of BERT. **DistilBERT (multi-lingual cased)** (Sanh et al., 2019b) is a distilled version of the multilingual Bert with pretraining on Wikipedia data in 104 languages. **BanglaBERT** (Sarker, 2020) referred to as HF-PT-BERT-2 in Table 1 is a pretrained BERT trained on the Bangla common crawl dataset and the Bangla Wikipedia Dump Dataset. **Indic-abusive-allInOne-MuRIL** (Das et al., 2022) is a model finetuned from the MuRIL (Khanuja et al., 2021) and multilingual BERT models, trained to detect abusive speech using multiple datasets in 8 Indian languages. **Bengali-abusive-MuRIL** (Das et al., 2022) is also finetuned from MuRIL (Khanuja et al., 2021), trained specifically on the Bangla abusive speech dataset. These have been referred to as HF-PT-BERT-1 and HF-PT-BERT-3 in Table 1, respectively. **BanglaBERT** (Bhattacharjee et al., 2022)is a fine-tuned ELECTRA (Clark et al., 2020) model which is trained on Bangla Wikipedia dump dataset as well as data from 110 Bangla websites. **BanglishBERT**(Bhattacharjee et al., 2022)is similar to BanglaBERT; instead, it was trained on both English and Bangla data to allow zero-shot cross-lingual transfer.

3.2 Ensembling Predictions

To increase the overall performance of the predictions and robustness of the predictive model, models were first individually tuned on the downstream task dataset. The predictions from these models were combined using the two ensembling methods on top-3,top-5, and all model predictions:

Majority Voting: The most frequently occurring prediction from all the models for each training instance was chosen as the final label.

Weighted: Each model was assigned a weight based on its accuracy score on the training dataset. Each model voted on the prediction class with its weight, and the prediction with the highest final vote was chosen as the final label.

$$y_i = argmax(\sum_{j=1}^{k} a_j.p_{ij}) \quad (1)$$

Here, y_i denotes the Weighted ensemble prediction of the ith sample, p_{ij} the ith probabilistic prediction for each polarity made by the jth model, a_j the accuracy of the jth model on the training set and k is the number of models being considered for the ensemble.

4 Experiments & Results

The dataset used for the task is organized in 3 columns, with id, text, and label. It has also been partitioned into a train set with 35266 samples, a dev set with 3935 samples, and a dev-test set with 3427 samples. The distribution in the training set is shown in Figure 1.

The preprocessing pipeline before model training included padding, tokenizing, and truncating text data to ensure uniformity and manage lengthy inputs. We used the AdamW optimizer, a learning rate of $2x10^{-5}$ and a batch size of 32 over 32 epochs was chosen to strike a balance between convergence speed and stability with a maximum sequence length of 512 tokens used with Huggingface AutoTokenizer to tokenize the data.

We evaluated models using four metrics: accuracy, precision, recall, and F1-score. F1-score is

Method	Top	Acc.	Prec.	Rec.	F1
Majo -rity Voted	3	0.706	0.692	0.706	0.706
	5	0.707	0.694	0.707	0.707
	All	**0.711**	**0.695**	**0.711**	**0.711**
Weig -hted	3	0.703	0.691	0.703	0.703
	5	0.703	0.692	0.703	0.703
	All	0.708	**0.695**	0.708	0.708

Table 3: Results of ensemble models on Test-Set of Shared-Task Dataset where Method is the method of ensembling, Top refers to top-k models chosen, Acc. is Accuracy, Pre. is Precision, Rec. is Recall & F1 refers to F1-Score

a good metric for imbalanced datasets because it takes into account both precision and recall.

The results of our experiments over the official Test set are shown in Table 2 & 3. For Individual Models as shown in Table 2 we observe DistilBERT and BanglaBERT(Base) show the best performance on the test data, with an F1-Score of 0.701.

We did an ensemble of both types (Majority-Voted and Weighted) with the top 3 (BanglaBERT (Sarker, 2020), BanglishBERT, HF-PT-BERT-1 (Das et al., 2022)), top 5 (HF-PT-BERT-2, BanglishBERT, HF-PT-BERT-1 (Das et al., 2022) , BanglaBERT(Base), HF-PT-BERT-3 (Das et al., 2022)) and lastly using all the models. As in Table 3 for ensembles, we observe that the majority ensemble shows a better performance in general as compared to the weighted models. The majority voted ensemble using predictions from all the models had the highest F1 score of 0.711. Furthermore, an ensemble of 3 models yielded almost optimal results. The use of more than three models resulted in a marginal increase in performance but significantly increased resource utilization. Thus, the use of more than three models seems unproductive.

5 Conclusion

In this work, we benchmarked various multilingual and pre-trained BERT-based models - RoBERTa(Liu et al., 2019a), DistillBERT(Sanh et al., 2019a), BanglaBERT(Bhattacharjee et al., 2022), BanglishBERT(Hasan et al., 2020b) and Various Pre-Trained BERT models (Das et al., 2022; Sarker, 2020) for Bangla Sentiment Analysis (Hasan et al., 2023a; Islam et al., 2021; Hasan et al., 2023b) while identifying the polarity of social media content by determining whether the sentiment expressed in the text is Positive, Negative, or Neutral as our downstream tasks and using a Majority Voting and Weighted ensemble model that outperforms individual baseline model scores.

Our system achieved a micro F1-Score of 0.711 for the multiclass classification task and scored 10th among the participants on the leaderboard for the shared task.

6 Acknowledgments

We would like to thank the Research Society MIT, an undergraduate interdisciplinary technical society of Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India, for supporting our research.

References

Md Al-Amin, Md Saiful Islam, and Shapan Das Uzzal. 2017. Sentiment analysis of bengali comments with word2vec and sentiment information of words. In *2017 international conference on electrical, computer and communication engineering (ECCE)*, pages 186–190. IEEE.

Firoj Alam, Arid Hasan, Tanvirul Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Imranul Ashrafi, Muntasir Mohammad, Arani Shawkat Mauree, Galib Md. Azraf Nijhum, Redwanul Karim, Nabeel Mohammed, and Sifat Momen. 2020. Banner: A cost-sensitive contextualized model for bangla named entity recognition. *IEEE Access*, 8:58206–58226.

Emily M Bender and Alexander Koller. 2020. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Thorsten Brants. 2000. Tnt-a statistical part-of-speech tagger. *arXiv preprint cs/0003055*.

Rumman Rashid Chowdhury, Mohammad Shahadat Hossain, Sazzad Hossain, and Karl Andersson. 2019. Analyzing sentiment of movie reviews in bangla by applying machine learning techniques. In *2019 international conference on bangla speech and language processing (ICBSLP)*, pages 1–6. IEEE.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Amitava Das and Sivaji Bandyopadhyay. 2010. Sentiwordnet for bangla. *Knowledge Sharing Event-4: Task*, 2:1–8.

Avishek Das, Omar Sharif, Mohammed Moshiul Hoque, and Iqbal H. Sarker. 2021. Emotion classification in a resource constrained language using transformer-based approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 150–158, Online. Association for Computational Linguistics.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. 2022. Data bootstrapping approaches to improve low resource abusive language detection for indic languages. *arXiv preprint arXiv:2204.12543*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Md. Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020a. Sentiment classification in bangla textual content: A comparative study. In *23rd International Conference on Computer and Information Technology (ICCIT)*.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020b. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623, Online. Association for Computational Linguistics.

A. Hassan, M. R. Amin, N. Mohammed, and A. K. A. Azad. 2016. Sentiment analysis on bangla and romanized bangla text (brbt) using deep recurrent models.

Alexander Hogenboom, Bas Heerschop, Flavius Frasincar, Uzay Kaymak, and Franciska de Jong. 2014. Multi-lingual support for lexicon-based sentiment analysis guided by semantics. *Decision Support Systems*, 62:43–53.

Naimul Hossain, Md Rafiuzzaman Bhuiyan, Zerin Nasrin Tumpa, and Syed Akhter Hossain. 2020. Sentiment analysis of restaurant reviews using combined cnn-lstm. In *2020 11th International conference on computing, communication and networking technologies (ICCCNT)*, pages 1–5. IEEE.

Khondoker Ittehadul Islam, Md Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5. IEEE.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Md Rafidul Hasan Khan, Umme Sunzida Afroz, Abu Kaisar Mohammad Masum, Sheikh Abujar, and Syed Akhter Hossain. 2020. Sentiment analysis from bengali depression dataset using machine learning. In *2020 11th international conference on computing, communication and networking technologies (ICCCNT)*, pages 1–5. IEEE.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *Second meeting of the North American chapter of the Association for Computational Linguistics*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew McCallum, Dayne Freitag, Fernando CN Pereira, et al. 2000. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598.

Islam Obaidat, Rami Mohawesh, Mahmoud Al-Ayyoub, Al-Smadi Mohammad, and Yaser Jararweh. 2015. Enhancing the determination of aspect categories and their polarities in arabic reviews using lexicon-based approaches. In *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6. IEEE.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *Proc. of MIKE*, pages 650–655. Springer.

Md Rahman, Emon Kumar Dey, et al. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2):15.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Conference on empirical methods in natural language processing*.

Hilke Reckman, Cheyanne Baird, Jean Crawford, Richard Crowell, Linnea Micciulla, Saratendu Sethi, and Fruzsina Veress. 2013. teragram: Rule-based detection of sentiment phrases using sas sentiment analysis. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 513–519.

Md Rezaul Karim, Bharathi Raja Chakravarthi, Mihael Arcan, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced Bengali language based on multichannel convolutional-lstm network. *arXiv*, pages arXiv–2004.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019a. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019b. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Ovishake Sen, Mohtasim Fuad, Md. Nazrul Islam, Jakaria Rabbi, Mehedi Masud, Md. Kamrul Hasan, Md. Abdul Awal, Awal Ahmed Fime, Md. Tahmid Hasan Fuad, Delowar Sikder, and Md. Akil Raihan Iftee. 2022. Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning-based methods. *IEEE Access*, 10:38999–39044.

Abdullah Aziz Sharfuddin, Md Nafis Tihami, and Md Saiful Islam. 2018. A deep recurrent neural network with bilstm model for sentiment classification. In *2018 International conference on Bangla speech and language processing (ICBSLP)*, pages 1–4. IEEE.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers*, pages 172–182.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6. IEEE.

Md Ferdous Wahid, Md Jahid Hasan, and Md Shahin Alom. 2019. Cricket sentiment analysis from bangla text using recurrent neural network with long short term memory model. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4. IEEE.

Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023. Is chatgpt a good sentiment analyzer? a preliminary study. *ArXiv*, abs/2304.04339.

Semantics Squad at BLP-2023 Task 2: Sentiment Analysis of Bangla Text with Fine Tuned Transformer Based Models

Krishno Dey[1], **Md. Arid Hasan**[1], **Prerona Tarannum**[2], **Francis Palma**[1]
[1]SE+AI Research Lab, University of New Brunswick, Fredericton, Canada
[2]Daffodil International University, Dhaka, Bangladesh
krishno.dey@unb.ca, arid.hasan@unb.ca,
prerona15-14134@diu.edu.bd, francis.palma@unb.ca

Abstract

Sentiment analysis (SA) is a crucial task in natural language processing, especially in contexts with a variety of linguistic features, like Bangla. We participated in BLP-2023 Shared Task 2 on SA of Bangla text. We investigated the performance of six transformer-based models for SA in Bangla on the shared task dataset. We fine-tuned these models and conducted a comprehensive performance evaluation. We ranked 20th on the leaderboard of the shared task with a blind submission that used BanglaBERT Small. BanglaBERT outperformed other models with 71.33% accuracy, and the closest model was BanglaBERT Large, with an accuracy of 70.90%. BanglaBERT consistently outperformed others, demonstrating the benefits of models developed using sizable datasets in Bangla.

1 Introduction

Social networking sites' widespread use in the digital age has produced an unheard-of influx of user-generated content. These sites act as gathering places where people can publicly express their opinions and feelings. It has become popular to identify and measure the emotional tone in textual data through sentiment analysis (SA), a key component of Natural Language Processing (NLP).

While SA has been extensively studied for *resource-rich* languages like English, it is still largely unexplored for many *low-resource* languages like Bangla. Understanding public opinion is crucial for making well-informed decisions in democratic countries. Developing efficient SA tools for the Bangla language has not been possible due to the lack of SA resources, such as datasets and evaluation benchmarks.

This study is devoted to SA and focuses specifically on Bangla being the 7th most spoken language globally (Ethnologue, 2023), and its use on social media sites, particularly Facebook, X, and YouTube, has increased significantly. While much research has been conducted in SA, most attempts have been based on traditional machine learning (ML). Traditional ML techniques have drawbacks in feature engineering, representation learning, scalability, and handling sequential data. They perform best when working with structured data that has clearly defined features. In contrast, deep learning (DL) models like Transformers have excelled at a variety of tasks, especially when dealing with unstructured data like natural language text. Despite the enormous amount of data generated on social media platforms, not many Bangla benchmark datasets are available.

This study addresses this gap by concentrating on the SA of Bangla text in the context of social media. We employ multiple state-of-the-art pretrained transformer models: multilingual BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018), XLM-RoBERTa (Conneau et al., 2019), BanglaBERT (Bhattacharjee et al., 2022), BanglishBERT (Bhattacharjee et al., 2022), fine-tuned for SA in Bangla.

We use the dataset provided in the shared task 2 of BLP-2023 (Hasan et al., 2023b) of Bangla text for SA in order to thoroughly assess the efficacy of these models. We measure and report the accuracy, precision, recall, and F1-score as important performance metrics used for SA evaluation. According to the performance matrices, monolingual models such as BanglaBERT BanglaBERT large outperform other transformer-based models.

We secured the 20th position with the submission of BanglaBERT Small of micro F1 score of 67.42%. The shared task followed a blind submission process, meaning the last submission was considered the final submission. Later, we reran the experiment with models including multilingual BERT, XLM-RoBERTa base, BanglishBERT, BanglaBERT Small, BanglaBERT and BanglaBERT large. The best-performing system

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 312–316

on the leaderboard achieved a micro F1 score of 73.1%, while BanglaBERT in our work achieved a micro F1 score of 70.62%, which is in close proximity to the best system's performance and significantly exceeds the baseline. BanglaBERT large also achieved an F1 score of 70.34%, and closely approaches the performance of the best system. Other models in our study did not achieve the same performance level as BanglaBERT and BanglaBERT large.

2 Related Work

There have been many attempts to address NLP tasks with traditional machine learning (ML). However, it has limitations related to feature engineering, representation learning, scalability, and handling unstructured data. In contrast, Transformer-based models can capture contextual information, rely on pre-trained representations, and can be applied to various languages and domains. Hence, we focus on NLP tasks that were addressed with deep learning (DL) and Language Models (LM).

Several attempts have been made by the researchers to develop resources for SA(Rahman et al., 2018; Tripto and Ali, 2018; Rezaul Karim et al., 2020; Patra et al., 2015), One of the most comprehensive and rigorous overviews of Bangla NLP tasks was conducted by (Alam et al., 2021) and (Hasan et al., 2020). They provided a comparative analysis of Bangla NLP tasks using both classical machine learning algorithms and transformer-based pre-trained models. Their study demonstrates that transformer-based pre-trained models outperform traditional machine learning algorithms.

(Bhowmik et al., 2022) used both DL and transformer-based models for SA of Bangla text. They used a domain-based categorical weighted lexicon data dictionary (LDD) (Bhowmik et al., 2021), which was developed for analyzing sentiments in Bangla from the original dataset (Rahman, 2018). They found that attention-based LSTM (HAN-LSTM), Dynamic routing-based capsule neural network with Bi-LSTM (D-CAPSNET-Bi-LSTM) and bidirectional encoder representations from Transformers (BERT) with LSTM (BERT-LSTM) outperformed other learning models. This study emphasized transformer models improve NLP tasks for languages with limited resources.

(Aurpa et al., 2022) addressed the growing issue of abusive comments in the Bangla language on social media platforms like Facebook. Using transformer-based models like BERT and ELECTRA (Clark et al., 2020), the study achieved a high accuracy of around 85% in identifying and classifying abusive comments from a novel dataset with more than 44k comments. (Rahman et al., 2020) conducted a study on Bangla text document classification using two transformer models, BERT and ELECTRA. The study highlighted the effectiveness of these models for accurately categorizing Bangla text documents, indicating their potential in NLP tasks. (Bhowmick and Jana, 2021) investigated the potential of multilingual BERT and fine-tuned XLM-RoBERTa for SA in Bangla as a low-resource language. The study demonstrated promising results, achieving a maximum accuracy of 95% across three different Bangla datasets, establishing itself as a valuable benchmark for this task. (Aurpa et al., 2022) addressed the growing issue of abusive comments in the Bangla language on social media platforms like Facebook. Using transformer-based models like BERT and ELECTRA, the study achieved a high accuracy of 85%.

In order to address the lack of high-quality Bangla SA datasets, (Hasan et al., 2023a) developed a dataset that focuses on attitudes toward the conflict between Russia and Ukraine. They fine-tuned various transformer-based models and achieved the best performance with 86% accuracy and 82% F-1 score using BanglaBERT. (Islam et al., 2020) introduced two manually tagged SA datasets and a DL model called BERTBSA.

3 Experimental Methodology

This section outlines our experimental methodology. We begin with an overview of the dataset, followed by a discussion of our pre-processing procedures, and conclude by presenting detailed descriptions of the models employed in our study.

Data: We used the dataset that was offered in the BLP shared task 2. The dataset employed for this shared task is a combination of Bangla text data from two distinct sources: MUBASE (Hasan et al., 2023c) and SentNob (Islam et al., 2021). SentNob is a compilation of public comments extracted from various social media platforms, spanning 13 domains such as politics, education, and agriculture, and manually annotated. The level of agreement among annotators for this dataset is moderate, with an agreement score of 0.53. On the other hand, the MUBASE dataset comprises a comprehensive col-

Split	# of Samples	Pos	Neg	Neu
Train	35,266	35%	45%	20%
Dev	3,934	35%	45%	20%
Test	6,707	31%	50%	19%

Table 1: Data Description and Split. Pos: Positive, Neg: Negative, Neu: Neutral

Model	Epochs	LR	Par
m-BERT	3	2e-5	180M
XLM-RoBERTa base	3	2e-5	270M
BanglishBERT	3	2e-5	110M
BanglaBERT Small	3	2e-5	13M
BanglaBERT	3	2e-5	110M
BanglaBERT large	3	2e-5	335M

Table 2: Training Parameters of Models. LR: Learning Rate, Par: Parameters

lection of multi-platform data, featuring manually labeled Tweets and Facebook posts, each categorized based on their sentiment polarity. This dataset presents a multi-class sentiment analysis (SA) challenge with three categories: *positive*, *negative*, and *neutral*. Table 1 show the overview of the data and splitting procedure.

Data Pre-Processing and Cleaning: Preprocessing for the Bangla text dataset offered in the shared task 2 of BLP-2023 entails several steps to ensure that the data is prepared for SA. First, standard text cleaning techniques like removing special characters, punctuation, extra white space, and URLs should be applied to the text data. Tokenization is then used to separate the text into tokens or single words. If stop words are present, they are typically eliminated to lower data noise. For modeling, it is crucial to convert the class labels into numerical values, such as 0 for negative, 1 for neutral, and 2 for positive.

Transformer-Based Models: We employed a variety of transformer-based models to conduct SA on the dataset provided for Shared Task 2. Our approach involved taking our pre-processed dataset and fine-tuning it using multiple transformer models, including m-BERT, XLM-RoBERTa base, BanglishBERT, and BanglaBERT. To optimize model performance, batch size of 32 was employed to expedite the training process, meaning that gradient accumulation was computed after every 32 data samples. The choice of a learning rate of $2e^{-5}$ was predicated on the rationale that this rate allows the algorithm to more effectively learn parameter estimates. Three epochs were sufficient for the models to converge on the dataset and avoid model overfitting and under-fitting. These experiments were conducted to explore the effectiveness of different transformer models in capturing sentiment patterns within the dataset and achieve the most accurate SA results. Batch size 32 was used to speed up the training process, and we set gradient accumulation count set 1 which means the gradient accumulation was calculated after 32 data samples. The learning rate of a $2e^{-5}$ was due to the fact that at this pace algorithms learn the values of a parameter estimate in a better way. Table 2 provides an overview of the model parameters.

4 Results Analysis and Discussion

To determine which models were most effective and could be applied to real-life SA problems, we fine-tuned and applied the m-BERT, XLM-RoBERTa base, BanglishBERT, BanglaBERT Small, BanglaBERT and BanglaBERT large models. In particular, BanglaBERT consistently outperformed the other models in terms of various performance metrics.

Table 3 presents a comprehensive breakdown of the performances of all these models. From the table, we can see that BanglaBERT achieved the highest accuracy with 71.33% on the test set, and among other Bangla pre-trained models, BanglaBERT large was also quite close with an accuracy of 70.9%. The other two models, namely BanglaBERT Small and BanglishBERT, achieved 67.23% and 68.81%, respectively. On the other hand, the multilingual model XLM-RoBERTa achieved an accuracy of 68.81%, and m-BERT achieved an accuracy of 65.56%. From the perspective of accuracy, BanglaBERT outperforms the other models. However, in terms of precision, BanglaBERT and BanglaBERT large are very close, averaging 70.22% and 70.07%, respectively. Regarding the F1 score, BanglaBERT and BanglaBERT large also exhibit similar performance, with average F1 scores of 70.62% and 70.4%, respectively. Another pattern that emerges from the table is that the performance measures for all models in the neutral class are lower than those for both the positive and negative classes. In fact, the performance measures for the negative

CL	Acc	P	R	F1
Multi-lingual BERT(m-BERT)				
Negative		0.71	0.75	0.73
Neutral	0.6556	0.45	0.37	0.41
Positive		0.67	0.68	0.68
XLM-RoBERTa base				
Negative		0.73	0.78	0.75
Neutral	0.6826	0.49	0.35	0.41
Positive		0.69	0.73	0.71
BanglishBERT				
Negative		0.76	0.76	0.76
Neutral	0.6881	0.49	0.36	0.42
Positive		0.67	0.78	0.72
BanglaBERT Small				
Negative		0.72	0.79	0.75
Neutral	0.6723	0.47	0.28	0.35
Positive		0.67	0.73	0.70
BanglaBERT				
Negative		0.76	0.80	0.78
Neutral	0.7133	0.48	0.38	0.43
Positive		0.74	0.77	**0.76**
BanglaBERT large				
Negative		0.76	0.80	0.78
Neutral	0.7090	0.48	0.40	0.44
Positive		0.74	0.76	0.75

Table 3: Comprehensive Breakdown of the Classification Results. Bold numbers indicate the best F1 score with respect to positive class. CL: Class Label, Acc: Accuracy, P: Precision, R: Recall, F1: F1 Score

.

class are superior to those of the other two classes for all models. This likely stems from the significantly higher number of samples in the negative class. Nearly 50% of the samples in the training, development, and test sets belong to the negative class.

However, we were unable to extract insights into why BanglaBERT exhibited superior performance compared to m-BERT and XLM-RoBERTa models. It is possible that BanglaBERT's training on a substantial Bangla dataset provided a slight advantage over the other multi-lingual models. The superior performance of BanglaBERT indicates that models specifically trained on a sizable Bangla dataset have a natural advantage when identifying subtle sentiment nuances in Bangla text. This may be attributed to the fine-tuning process used by BanglaBERT, which allowed it to better comprehend the nuances of Bangla language and sentiment expression. However, despite being intended to be multi-lingual models, m-BERT and XLM-RoBERTa may not have fully adapted to the nuances of the Bangla language, which resulted in their comparatively poorer performance.

Although BanglaBERT outperformed the other models, our study could not pinpoint the precise causes of this performance disparity. For example, despite being larger and having three times more parameters than BanglaBERT, BanglaBERT large could not perform as expected. The observed behavior may be attributed to several potential factors within the context of the data provided for Shared Task 2 of BLP-2023. One likely contributor could be the inadequacy of the data structure for the models to perform optimally. Another possibility is that the pre-processing steps applied to the data may not have been sufficient to enable the models to achieve their expected levels of performance. Additionally, the choice of hyper-parameters for the models, including the fine-tuning process, might not have been optimal, potentially impacting their overall performance.

5 Conclusion and Future Work

This study conducted a comprehensive evaluation of fine-tuned transformer-based models for sentiment analysis (SA) in Bangla text. The importance of models specifically trained on large Bangla datasets for SA tasks is highlighted by BanglaBERT's consistent and superior performance across a variety of performance metrics. The advantage that BanglaBERT showed over the multi-lingual models, m-BERT and XLM-RoBERTa, suggests that a deeper comprehension of the Bangla language and sentiment expression is crucial for obtaining accurate SA results. The precise linguistic and contextual factors contributing to BanglaBERT's superior SA abilities need to be further investigated. In our future research endeavors, we aim to delve deeper into why transfer-based multi-lingual models struggled to compete with BanglaBERT, further enhancing our understanding of their performance disparities.

References

Firoj Alam, Md Arid Hasan, Tanvir Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Tanjim Taharat Aurpa, Rifat Sadik, and Md Shoaib Ahmed. 2022. Abusive bangla comments detection on facebook using transformer-based deep learning models. *Social Network Analysis and Mining*, 12(1):24.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Mubasshir, Md. Saiful Islam, Wasi Ahmad Uddin, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Lagnuage model pretraining and benchmarks for low-resource language understanding evaluation in bangla. In *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL 2022*.

Anirban Bhowmick and Abhik Jana. 2021. Sentiment analysis for bengali using transformer based models. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 481–486.

Nitish Ranjan Bhowmik, Mohammad Arifuzzaman, and M Rubaiyat Hossain Mondal. 2022. Sentiment analysis on bangla text using extended lexicon dictionary and deep learning algorithms. *Array*, 13:100123.

Nitish Ranjan Bhowmik, Mohammad Arifuzzaman, M Rubaiyat Hossain Mondal, and MS Islam. 2021. Bangla text sentiment analysis using supervised machine learning with extended lexicon dictionary. *Natural Language Processing Research*, 1(3-4):34–45.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Ethnologue. 2023. The most spoken languages worldwide in 2023. https://www.ethnologue.com/insights/ethnologue200/. [Online; accessed 09-September-2023].

Mahmud Hasan, Labiba Islam, Ismat Jahan, Sabrina Mannan Meem, and Rashedur M Rahman. 2023a. Natural language processing and sentiment analysis on bangla social media comments on russia–ukraine war using transformers. *Vietnam Journal of Computer Science*, pages 1–28.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023b. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023c. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Md. Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020. Sentiment classification in bangla textual content: A comparative study. In *23rd International Conference on Computer and Information Technology (ICCIT)*.

Khondoker Ittehadul Islam, Md Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5. IEEE.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *Proc. of MIKE*, pages 650–655. Springer.

Atik Rahman. 2018. Bangla absa datasets for sentiment analysis. `https://github.com/atik-05/Bangla_ABSA_Datasets`.

Md Rahman, Emon Kumar Dey, et al. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2):15.

Md Mahbubur Rahman, Md Aktaruzzaman Pramanik, Rifat Sadik, Monikrishna Roy, and Partha Chakraborty. 2020. Bangla documents classification using transformer based deep learning models. In *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pages 1–5. IEEE.

Md Rezaul Karim, Bharathi Raja Chakravarthi, Mihael Arcan, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced Bengali language based on multichannel convolutional-lstm network. *arXiv*, pages arXiv–2004.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *Proc. of ICBSLP*, pages 1–6. IEEE.

Aambela at BLP-2023 Task 2: Enhancing BanglaBERT Performance for Bangla Sentiment Analysis Task with In Task Pretraining and Adversarial Weight Perturbation

Md Fahim
Center for Computational & Data Sciences
Independent University, Bangladesh
Dhaka-1229, Bangladesh
fahimcse381@gmail.com

Abstract

This paper introduces the top-performing approach of "Aambela" for the BLP-2023 Task 2: "Sentiment Analysis of Bangla Social Media Posts". The objective of the task was to create systems capable of automatically detecting sentiment in Bangla text from diverse social media posts. My approach comprised fine-tuning a Bangla Language Model with three distinct classification heads. To enhance performance, we employed two robust text classification techniques. To arrive at a final prediction, we employed a mode-based ensemble approach of various predictions from different models, which ultimately resulted in the **1st place** in the competition.

1 Introduction

In recent years, Natural Language Processing (NLP) has advanced significantly, highlighting the importance of sentiment analysis. This application provides insights into public opinion and social media trends. In the context of Bangla text, sentiment analysis is crucial, aiding businesses in interpreting customer feedback, assisting policymakers in understanding public sentiment, and boosting media engagement. Concerning the importance of sentiment analysis, the organziers of BLP-Shared Task 1 (Hasan et al., 2023a) provide one of the largest manually annotated datasets for sentiment analysis which encompasses sentiment across multiple platforms.

The proposed sentiment analysis approach involves fine-tuning the Bangla Language Model, such as BanglaBERT(Bhattacharjee et al., 2022), and utilizing three distinct classification heads to enhance model performance. To address overfitting and ensure robust generalization, strategies like cross-validation and adversarial perturbation techniques are employed. Task-specific pretraining of BanglaBERT on both the train and train+validation datasets is explored, yielding performance improvements. Different classification heads in various techniques focus on distinct aspects of sentiment classification reasoning. To capture these diverse perspectives, a mode-based ensemble technique is applied. The ensemble predictions prove to be the best-performing model in the experiments, securing the top position on the leaderboard.

2 Background

2.1 Task & Dataset Description

The primary aim of this task (Hasan et al., 2023a) is to conduct sentiment analysis on Bengali textual data, focusing on multi-class sentiment classification. In essence, it involves categorizing text into one of three distinct sentiment classes: Positive, Negative, or Neutral. The overarching objective is to create a model that can effectively and precisely assign text to these sentiment categories by discerning its emotional context.

Data Splits	Total Samples	Class wise Samples		
		Negative	Positive	Neutral
Train	35266	15767	12364	7135
Dev	3934	1753	1388	793
Test	6707	3338	2092	1277

Table 1: Dataset Statistics for Shared Task 2 (Sentiment Analysis Task).

The dataset for this shared task is a fusion of two distinct sources: MUBASE (Hasan et al., 2023b) and SentNob (Islam et al., 2021). SentNob encompasses public comments sourced from diverse social media platforms, spanning 13 domains including politics, education, and agriculture. Conversely, the MUBASE dataset comprises an extensive collection of multi-platform data, featuring manually labeled Tweets and Facebook posts. The dataset statistics along with class wise sample size is provided in Table 1.

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 317–323

2.2 Observations and Baselines

Upon analyzing the dataset, several key observations emerged. Firstly, despite the presence of numerous URLs, they appeared to have no substantial influence on the dataset's attributes. Additionally, there was an absence of class dependency linked to these URLs. Moreover, emojis within the dataset did not appear to significantly impact the analysis. And also, the dataset exhibited a notable prevalence of error words, a common feature in text collected from YouTube comments. These observations offer valuable insights into the dataset's nature and characteristics.

The organizers have also provided baseline results for this task on both the Dev-Test and Test Dataset. Three different methods were employed: the Random Baseline, Majority Baseline, and the n-gram Baseline. Notably, the n-gram Baseline demonstrated better performance, surpassing the other two methods by a good margin. In the test dataset, the n-gram Baseline achieved an impressive 55.14% micro F1 score, while on the Dev-Test dataset, it reached 57.36%.

3 Method Description

3.1 ITPT

withIn Task PreTraining (ITPT) is a popular approach while solving text classification problem. It was proposed by (Sun et al., 2019). We also use this ITPT techniques in our task. BanglaBERT undergoes training in a broad domain, characterized by a distinct data distribution when compared to the target domain. So we perform additional pretraining on BanglaBERT using data specific to the target domain. Actually, we futher perform Masked Langugae Modeling (MLM) (Devlin et al., 2019) using the pretrained BanglaBERT on our training corpus.

3.2 AWP

Adversarial Weight Perturbation (AWP) (Wu et al., 2020) is a regularization technique that encourages neural networks to have stable and robust weights by penalizing sensitivity to parameter perturbations. This regularization improves the model's generalization and robustness, making it less susceptible to adversarial attacks.

In the neural network, the loss function is denoted as $\mathcal{L}(\Theta)$, where Θ represents the model parameters. The objective is to minimize this loss on a training dataset. AWP introduces a regularization term to penalize the sensitivity of the model's output to small perturbations in its parameters. This is added to the loss function:

$$\mathcal{L}_{\text{AWP}}(\Theta) = \mathcal{L}(\Theta) + \lambda \cdot \text{AWP}(\Theta)$$

Here, λ controls the regularization strength, and $\text{AWP}(\Theta)$ is the AWP term. The AWP term is designed to adversarially perturb the model's weights and is formulated as:

$$\text{AWP}(\Theta) = \frac{1}{2} \sum_i \left\| \frac{\partial \mathcal{L}}{\partial \Theta_i} \right\|_2^2$$

This term quantifies the sensitivity of the loss function to changes in each parameter Θ_i and encourages stable and robust weight values. During training, the combined loss function $\mathcal{L}_{\text{AWP}}(\Theta)$ is optimized. AWP's regularization helps prevent overfitting and enhances the model's resistance to adversarial attacks.

3.3 Classification Heads

For an input sentence S, we obtain $S = \{t_1, t_2, \ldots, t_n\}$ after processing the sentence with the BanglaBERT tokenizer, where t_i represents the i-th token. Then the sentence S through a BanglaBERT model, we obtain contextual representations of last layer for each token t_i, denoted as $H = \{h_1, h_2, \ldots, h_n\}$, where h_i represents the contextual representation of token t_i.

3.3.1 FFN Head on CLS Token

In order to obtain a fixed-size representation for the entire sentence to use in classification, we utilize the special [CLS] token representation, denoted as h_{CLS} which is fed into a two-layer Feed Forward Neural Network (FFN). The resulting representation z is employed for the classification process by following method.

$$z = W_2 \cdot (\text{ReLU}(W_1 \cdot h_{\text{CLS}} + b_1)) + b_2$$

3.3.2 Mean, Max, Min Pooling

As our model does batchwise operations, so the sequence may contain padded values for equal length. BanglaBERT model provides an input mask vector M for a sentence in a batch where $M = [m_1, m_2, \ldots, m_n]$ to indicate valid tokens. $m_i = 1$ for valid tokens and 0 for padded values. Then we apply *MeanPooling*, *MinPooling*, *MaxPooling* (Minaee et al., 2021) as followings:

Techniques	Classification Head	CV Score	Performance Metrics			
			Dev Set		Test Set	
		Micro F1	Accuracy	Macro F1	Accuracy	Macro F1
Without AWP	CLS + MLP	72.36	72.67	68.82	71.05	66.29
	Dropouts Enhanced MLP	72.24	73.00	69.19	70.81	65.65
	[Mean, Max, Min] Pooling	72.38	71.73	68.72	71.15	67.33
	Reinit Last Two Layers	72.39	72.32	68.50	71.48	66.74
With AWP	CLS + MLP	73.21	74.12	70.05	72.64	67.58
	Dropouts Enhanced MLP	72.90	73.87	69.29	72.72	67.30
	[Mean, Max, Min] Pooling	73.24	72.34	69.83	71.72	68.42
	Reinit Last Two Layers	73.47	72.52	69.62	71.58	68.00
Including Dev Dataset	CLS + MLP	73.83	-	-	72.40	67.32
	Dropouts Enhanced MLP	73.77	-	-	72.42	67.41
	[Mean, Max, Min] Pooling	73.76	-	-	71.28	67.41
	Reinit Last Two Layers	73.91	-	-	71.50	67.34
ITPT on Training Data	CLS + MLP	73.49	74.17	70.26	72.76	67.97
	Dropouts Enhanced MLP	73.47	74.17	70.23	72.89	67.83
	[Mean, Max, Min] Pooling	73.60	72.88	70.18	71.76	68.29
	Reinit Last Two Layers	73.42	72.47	69.82	70.85	67.60
ITPT on Train + Validation Data	CLS + MLP	73.74	74.07	70.12	72.66	67.84
	Dropouts Enhanced MLP	73.79	74.10	70.07	72.51	67.63
	[Mean, Max, Min] Pooling	73.94	73.31	70.52	71.48	67.99
	Reinit Last Two Layers	73.59	72.93	70.27	71.39	68.03
	Ensemble	-	-	-	**73.10**	**68.74**

Table 2: Performance of BanglaBERT in Sentiment Analysis in Shared Task 2 with different Techniques. While experiments were done with including validation (dev) dataset the measurement on dev set were skipped. **Ensemble** model was the final model which place first in the leaderboard. Scores with underline can also be the top scorer.

$$
\begin{aligned}
\textit{Mean\_Pool} &= \text{MeanPooling}(X, M) \\
&= \frac{1}{\sum_{i=1}^{n} m_i} \sum_{i=1}^{n} m_i \cdot h_i \\
\textit{Min\_Pool} &= \text{MinPooling}(X, M) \\
&= \min_{i=1}^{n} (m_i \cdot h_i) \\
\textit{Max\_Pool} &= \text{MaxPooling}(X, M) \\
&= \max_{i=1}^{n} (m_i \cdot h_i)
\end{aligned}
$$

Then we concat those pooling and passed them into a two layer MLP for finding class logits for classification as follows:

$$
\begin{aligned}
z = W_2 \cdot (\text{PReLU}(W_1 \cdot [&Mean\_Pool, \\
&Min\_Pool, Max\_Pool]) + b_1)) + b_2
\end{aligned}
$$

3.3.3 Dropout-Enhanced CLS Token Head

In this case, an expanded classification head is incorporated, which is an enhancement of the CLS_MLP head discussed in Section 3.3.1. In this variation, we apply dropout to the FFN layer. We explore a range of distinct dropout rates denoted as $D = d_1, d_2, \ldots, d_k$, where d_i signifies the dropout rate for the i-th rate. For a specific dropout rate d_i, we calculate class representations z_i using the subsequent equation:

$$z_i = W_2 \cdot (\text{DropOut}(d_i)(\text{ReLU}(W_1 \cdot h_{\text{CLS}}) + b_1)) + b_2$$

After acquiring m unique class representations (logits), we calculate the final representation z by taking the average of these representations, following the equation:

$$z = \frac{1}{m} \sum_{i=1}^{m} z_i$$

3.3.4 Re-initialization of last 2 layers of BanglaBERT

In this case, we re-initialize the last two layers for BanglaBERT like (Zhang et al., 2020) did. If the original BERT model as $M_{BanglaBERT}$, which comprises multiple layers. When we say we are re-initializing the last two layers, it means we are modifying these layers to create a new model, which we'll call M_{New}. The re-initialized model M_{New} can be defined as

$$M_{\text{New}} = M_{\text{BanglaBERT}}[: L-2] + \text{Reinitialize}(M_{\text{BanglaBERT}}[L-2 :])$$

For involving classification task in this case, we use MLP Head on CLS token similar to we describe in Section 3.3.1.

4 Result and Analysis

Different models and experiments were done during the development phase which are reproted in Appendix B. The experiment set up and hyper parameters details are described in Appendix A. Another experiments for model choice encompassed machine learning models (SVM, RandomForest, XGBoost) using TF-IDF feature extraction, deep learning models (LSTM, LSTM+Attention), and multilingual Transformer models (mBERT, mDeBerta, XLMRoberta base), with mDeBerta showing superior performance. Additionally, two Bangla Language Models were considered, with the *csebuetnlp-BanglaBERT* model emerging as the top performer. Table 3 summarizes the experimental results for model selection.

Table 3 displays key experiments using the *csebuetnlp/banglabert* model backbone for contextualized word representations, coupled with various classification heads as discussed in Section 3.3.

Model Name	Acc ↑	F1 ↑
TF-IDF + SVM	55.74	44.41
TF-IDF + RandomForest	58.41	50.65
TF-IDF + XGBoost	60.99	53.95
LSTM	65.91	61.88
LSTM + Attention	67.82	63.76
mBERT-case	66.29	62.19
mDeBerta-v3 base	70.84	64.35
XLM-Roberta-base	69.67	61.58
SagorSarker-BanglaBERT	67.08	61.30
csebuetnlp-BanglaBERT	**72.57**	**66.42**

Table 3: Different Types of Model Performance in Validation (Dev) Dataset.

These experiments employ 5-fold cross-validation. The inclusion of AWP (Section 3.2) enhances both cross-validation scores and generalization to validation and test datasets by approximately 1-2%. However, incorporating the validation data into training yields a slightly lower test set performance despite boosting the CV score. ITPT (Section 3.1) on training data significantly enhances performance across all classification heads. Conversely, including the validation data during ITPT yields mixed results, with slight improvements in some heads and minor reductions in others.

A final prediction is made by ensembling all classification heads from different techniques. The ensemble technique employed is a **Mode-based Ensemble**, aggregating predictions from all models across techniques and selecting the mode as the final prediction. This approach achieved an accuracy of **73.10%** (micro F1) and **68.74%** (macro F1) on the test set, placing it at the **top of the leaderboard**. Though the model has a highest score in the leaderboard, it has some limiations and scope for improvements which are describe in Limiation 5 section and Appendix C.

5 Conclusion

In this work,we have experimented with fine-tuning BanglaBERT in different aspects using different classification heads. The result showed that it gives a better score. Adverserial training and cross validation made the model more robust. In task pretraining helped the model to further investigate different classes of sentiment analysis. Our finding is that using adverserial training and in task pretraining we can improve our model further and build up a better model.

Limitations

The proposed models are struggled to predict the Neutral samples. Besides a good amount of sentences have token length larager than 512. To fit those sentences, we need to truncate the token length to 512. More on error analysis and scopes for improvement can be found at Appendix C.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. 2020. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*.

A Experimental Setup & Hyperparameters

In every dataset, we conducted critical preprocessing steps for text, encompassing the elimination of punctuation, emojis, and any existing URLs. We applied different types of models including TF-IDF+SVM, TF-IDF+RandomForest, TF-IDF+XGBoost, LSTM, LSTM+Attention, mBERT-case, mDeBerta-v3 base, XLM-Roberta-base and SagorSarker-BanglaBERT and csebuetnlp-BanglaBERT for Dev dataset. To extract hidden representations from the text, we employed two distinct models: LSTM and BERT, as the text encoding methods.

When using the LSTM-based models, an embedding layer with an embedding dimension of 128 was employed to convert the tokens into vector representations. The LSTM model's hidden dimension was set to 256. We used a learning rate of 10^{-3} and a batch size of 8 for this configuration.

On the other hand, for the BERT model, we utilized the *Bangla-bert* variants that enables us to extract contextual representations through fine-tuning. Bert model along with other transformers models include the hidden dimension 768. The learning rate for BERT was 2×10^{-5}, max length was 512 and a batch size of 8 were used for the models. This token length was consider because of its performance shown for Dev dataset from table 4, while from table 5 for Dev dataset showed batch size 8 performed better than other batch size configurations. Which encouraged the usage of the batch size 8 along with maximum length 512 in this study for transformer based models.

Both configurations employed the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. To ensure robustness, we performed five-fold cross-validation and three different random seeds. Additionally, we set $\lambda = 10$ for all experiments. An ablation study investigating the effect of different λ values is presented in Table. All experiments were conducted using Python (version 3.8) and PyTorch, leveraging the free NVIDIA Tesla K80 GPU available in

Google Colab, as well as a single NVIDIA Tesla P100 GPU provided by Kaggle.

B Ablation Study

In this section detailed ablation study was performed which contains max length effects, batch size effects and effects of different loss was measured and analyzed.

B.1 Token Length analysis

In table 4, for validation (Dev) dataset BanglaBert with 'csebuetnlp/BanglaBert' was applied for different max lengths. For 512 max length maximum accuracy and f1 score was achieved with outperforming other variation by 1%-2%.

Max Length	Dev Acc ↑	Dev F1 ↑
64	72.24	67.12
128	71.2	66.1
256	72.22	66.93
512	**73.34**	**68.52**

Table 4: Token Length Effect of *csebuetnlp/BanglaBert* in Validation (Dev) Dataset. Epoch Size 3

B.2 Batch Size Effects

Table 5 depicts the batch size for which the maximum accuracy and F1 score was achieved. For batch size 8 bested other variation by slight margin, ranging from 0.5% to 1%.

Batch Size	Dev Acc ↑	Dev F1 ↑
8	**72.52**	**67.74**
16	72.22	66.93
32	72.31	67.01

Table 5: Batch Size Effect of *csebuetnlp/BanglaBert* in Validation (Dev) Dataset while Max Length = 512 were considered. Epoch Size 3

B.3 Ablation On Losses

In this study, table 6 represent the effects of different loss variation were measured. From the table, Cross Entropy Loss(CE Loss) outperformed other variations including Weighted CE Loss, and Focal loss along with Cross Entropy Loss. Cross Entropy Loss showed improvement in its performance matrices by 1%-2% for all other losses.

Loss Name	Dev Acc ↑	Dev F1 ↑
CE Loss	**73.31**	**68.41**
Weighted CE Loss	72.5	67.12
0.5*Focal + 0.5*CE	72.72	67.41
0.3*Focal + 0.7*CE	71.35	66.36

Table 6: Batch Size Effect of *csebuetnlp/BanglaBert* in Validation (Dev) Dataset while Max Length = 128 were considered with Batch Size = 16. Epoch Size 5. Here *CE* indicates the *Cross Entropy Loss* and *Focal* means the *Focal Loss*

	precision	recall	f1-score	support
Negative	0.7803	0.8119	0.7958	3338
Neutral	0.5040	0.4957	0.4998	1277
Positive	0.7887	0.7457	0.7666	2092
accuracy			0.7310	6707
macro avg	0.6910	0.6844	0.6874	6707
weighted avg	0.7303	0.7310	0.7303	6707

Figure 1: The list of words that are considered as new tokens to the model.

B.4 Text Preprocessing Effects

Different preprocessing variations were also considered for this research endeavour. Removing URLs, Punctuation & Emoji's, Removing Punctuation only and Removing Emoji's only showed least improvement in the performance matrices, containing almost similar values. No preprocessing and Removing punctuation showed improvement by 1% from the previous variations. Applying BN-Unicode Normalizer after removing URLs + HTML Tag showed, or adding Normalizer after removing URLs + HTML Tag showed the most improvement by 1%-2% from aforementioned models. While, normalizing after removing URLs + HTML Tag bested all other preprocessing variations in terms of performance matrices.

C Error Analysis & Scope for Improvements

In Figure 1, the classification report for mode-base-ensembel, which gives the top performance score in leaderbaoard is reported. From the classification report it is easily seen that the proposed models are struggled when the class label is *Neutral*. One of reasons for the poor performance on the *Neutral* class labelled is that we have fewer samples for this class rather than the remaining classes 1. Besides there may overlapping words for the classses.

There are few scopes for improvements by which the model may be more efficient. Increasing class

Preprocessing	Dev Acc ↑	Dev F1 ↑
No Preprocessing	71.2	66.10
Removing URLs, Punctuation & Emoji's	70.67	66.56
Removing Punctuation Only	70.01	65.52
Removing URLs & HTML Tags	71.59	66.46
Removing Emoji's Only	70.87	66.84
Adding Normalizer after removing URLs + HTML Tag	**72.57**	**67.12**
Adding BN-Unicode Noramlizer after removing URLs + HTML Tag	72.22	66.93

Table 7: Effect of different preprocessing techniques in devset performance for BanglaBERT. Each experiment was trained for 3 epochs.

samples for *Neural* class may help. An external data can be used for this. As a good amount sentences have token length greater than 512, different techniques like (Chunking or Sliding Window, Document-Level Embeddings and so on) can be used. Besides different augmentation techniques can also be examined.

Z-Index at BLP-2023 Task 2: A Comparative Study on Sentiment Analysis

Prerona Tarannum[1]†, **Md. Arid Hasan**[2]†, **Krishno Dey**[2], **Sheak Rashed Haider Noori**[1]
[1]Daffodil International University, Dhaka, Bangladesh
[2]SE+AI Research Lab, University of New Brunswick, Fredericton, Canada
prerona15-14134@diu.edu.bd, arid.hasan@unb.ca,
krishno.dey@unb.ca, drnoori@daffodilvarsity.edu.bd

Abstract

In this study, we report our participation in Task 2 of the BLP-2023 shared task. The main objective of this task is to determine the sentiment (Positive, Neutral, or Negative) of a given text. We first removed the URLs, hashtags, and other noises and then applied traditional and pretrained language models. We submitted multiple systems in the leaderboard and BanglaBERT with tokenized data provided the best result and we ranked 5th position in the competition with an F1-micro score of 71.64. Our study also reports that the importance of tokenization is lessening in the realm of pretrained language models while the base models outperform the large models. In further experiments, our evaluation shows that BanglaBERT outperforms, and predicting the neutral class is still challenging for all the models.

1 Introduction

Sentiment Analysis is one of the most modern and sophisticated Natural Language Processing (NLP) applications. It is used for analyzing how people feel about the words they write in publicly accessible spaces like social media in the form of posts or comments. Social networking sites and other ways to use digital technology are commonly used to post a lot of information about feelings, ideas, and actions. Access to such a great amount of data provides the researchers the advantage to analyze the contents in order to help make decisions to process and understand the sentiment of a product and system, or views on social, international, cultural, and political agendas Hasan et al. (2020a).

The majority of current research is limited to resource-rich languages due to the availability of resources. The interest in low-resource languages is growing over time in sentiment analysis(Batanović et al., 2016; Nabil et al., 2015; Muhammad et al., 2023). Unlike other languages, a limited number of study has been done to develop resources for Bangla sentiment analysis (Hasan et al., 2020a; Alam et al., 2021; Islam et al., 2021; Hasan et al., 2023b; Islam et al., 2023). From the perspective of modeling, there have been studied both classical (i.e., SVM, RF, Naive Bayes) and deep learning (i.e., CNN, LSTM) models. Pretrained language models (i.e., BERT, XLM-RoBERTa, DistilBERT) have also been studied in recent years (Hasan et al., 2020a; Alam et al., 2021) for sentiment classification. Due to the availability of public data and inadequate information on annotation agreements (Alam et al., 2021), it is challenging for the researchers to focus on this area. This shared task provides a dataset by combining perfect and moderate agreement to shed light on sentiment analysis.

In this study, we participated in the Sentiment Analysis Shared Task at BLP-2023 and worked on a multiclass dataset where the class labels are Positive, Negative, and Neutral. We utilize both classical and transformer-based pretrained language models. For the classical model, we choose Support Vector Machine (SVM) and Random Forest (RF). We fine-tuned BERT multilingual, BanglaBERT base, and BanglaBERT large pretrained language models to train and evaluate models. Our findings from the study conclude as:

(i) The importance of tokenization before feeding into the models is diminishing in the presence of pretrained language models. There is little to no difference in performances between tokenized and non-tokenized data.

(ii) All the models are struggling to classify the neutral class.

(iii) Fine-tuned monolingual pretrained models outperform multilingual models.

(iv) Base model outperforms the large model.

The rest of the structure of this paper is as follows. We provide a brief overview of the literature in section 2. We discussed the data and approaches that we used for our experiments in section 3. Fol-

†The authors contributed equally to this work

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 324–330

lowing this in Section 4, we report results and discuss our findings. Finally, we conclude our work in Section 5.

2 Literature Review

Researchers are increasingly interested in investigating sentiment analysis utilizing social media data as a result of the rise of social media. The development of sentiment analysis began in the early 2000s (Pang et al., 2002). Early research includes rule-based and classical methodologies whereas recent studies include deep learning-based and pretrained language models. Researchers have been trying to develop resources over time and as a result, manual and semi-supervised approaches (Chowdhury and Chowdhury, 2014; Alam et al., 2021; Islam et al., 2021, 2023; Kabir et al., 2023) have been adopted in developing sentiment classification datasets. Chowdhury and Chowdhury (2014) used a semi-supervised technique to annotate data and train classical models. The study by Islam et al. (2021) constructs a dataset using manual annotations done by the annotators and presents 15,000 data in 13 domains. Rahman and Kumar Dey (2018) in their work, used the ABSA dataset consisting of human-annotated user comments on cricket and customer reviews of restaurants where SVM offered the maximum precision rate for both datasets.

Islam et al. (2016) developed a sentiment classification system utilizing SVM and Naive Bayes for textual movie reviews in Bangla and provided comparative results. Additionally, Naive Bayes with rules has been studied by Islam et al. (2016) for Bangla Facebook statuses sentiment classification. Hassan et al. (2016) worked on 10,000 post-processed text samples in both Bangla and Romanization of Bangla and by experimenting with LSTM, the authors achieved the maximum accuracy score of 55%. Hasan et al. (2020a) conducted comparison experiments using various datasets that existed in the literature to understand model performances, training difficulties, and consequences for real-world deployment. In this study, deep learning-based models outperform traditional models.

Furthermore, Alam et al. (2021) used the most sophisticated techniques currently available to compare datasets and conclude that XLM-RoBERTa exhibits the best performance over other deep learning approaches. Classifying the tweets of positive, negative, and neutral polarity was the major goal of SAIL-2015 Patra et al. (2015). Various well-known supervised classification methods have been studied in this study. Tripto and Ali (2018) used LSTM for identifying sentiment and emotions in Bangla writings achieving an accuracy of 65.97 and 54.24 for three and five classes respectively. Chowdhury et al. (2019) providing a method for conducting sentiment analysis on Bangla-language movie reviews that can automatically analyze viewer responses to a certain film or television program was the main work and the authors used social media websites' publicly accessible comments and posts serving as the source of the dataset that was manually compiled and labeled for this experiment.

Focusing on the largest publicly available dataset MUBASE (Hasan et al., 2023b) consolidated from social media data consisting of 33,605 tweets and Facebook comments about Bangla news and carried out experiments that went beyond traditional approaches and smaller transformer-based models. The authors focused on the efficiency of sophisticated algorithms in zero- and few-shot conditions, including Flan-T5, GPT-4, and Bloomz. The findings show that while LLMs are an interesting study area, smaller variations of precise pre-trained models perform better. In the context of sentiment analysis Cambria et al. (2022) provides a commonsense-based neurosymbolic framework that seeks to address these problems. They evaluated SenticNet 7 and concluded that of all 20 lexica, SenticNet 7 was the most effective. Ye et al. (2022) worked with the manually produced and labeled datasets that were obtained from social media. The accuracy achievement at the end of 140 epoch with the best performance using the NADAM optimizer.

3 Methodology

3.1 Data

Class	Train	Dev	Dev-Test	Test
Positive	12,364	1,388	1,126	2,092
Neutral	7,135	793	600	1,277
Negative	15,767	1,753	1,700	3,338
Total	**35,266**	**3,934**	**3,426**	**6,707**

Table 1: Class label distribution of the shared task dataset for each data split.

We utilized the dataset provided by the organizers of the BLP-2023 for task 2: Sentiment Analysis (Hasan et al., 2023a) . The goal is to iden-

Input text: **নতুন সেনাপ্রধান লে. জে. এস এম শফিউদ্দিন আহমেদ **২৪ জুন দায়িত্ব নেবেন এস এম শফিউদ্দিন আহমেদ

Tokenized Text: ['*', '*', 'নতুন', 'সেনাপ্রধান', 'লে', '.', 'জে', '.', 'এস', 'এম', 'শফি', '##উদ্দিন', 'আহমেদ', '*', '*', '২৪', 'জুন', 'দায়িত্ব', 'নেবেন', 'এস', 'এম', 'শফি', '##উদ্দিন', 'আহমেদ']

Encoded Text: [2, 14, 14, 1299, 21384, 1128, 18, 1683, 18, 1880, 1611, 28485, 4286, 3232, 14, 14, 3083, 3702, 2140, 7453, 1880, 1611, 28485, 4286, 3232, 3]

Figure 1: Representation of tokenized training text of id: 30960

tify the sentiment contained within a text. The dataset is consolidated from two distinct sources, i) MUBASE (Hasan et al., 2023b) and ii) Sent-NoB (Islam et al., 2021) consisting of social media tweets, posts, and comments. In this dataset, there are three columns, ID refers to sentence id, text refers to input text, and label containing Positive, Neutral, and Negative tags. In table 1, we present the class-wise official data distributions that are provided in the shared task.

3.2 Preprocessing

The dataset which is given for the Sentiment Analysis shared task at BLP-2023 was generated via social media, where it contains noise like emoticons, usernames, hashtags, URLs, invisible letters, and symbols. We went through numerous preprocessing stages to clear up these noisy data. We first removed unnecessary characters and URLs and then we removed the stopwords hashtags, and usernames from the data. We also used normalizer (Hasan et al., 2020b) before feeding into the pretrained language model.

3.3 Model

We run some traditional models and BERT-based models on the dataset. Several factors have been considered during the selection of these algorithms. The superior performance for the Bangla language is one of the main reasons for choosing BanglaBERT (Bhattacharjee et al., 2022) and BERT multilingual(Devlin et al., 2018) provides comparable results. We used two variants (base and large) of BanglaBERT. For the traditional models, we choose two popular algorithms such as Random Forest (RF) (Liaw et al., 2002) and SVM (Platt, 1998).

3.4 Experiments

BERT-based Models: Transformer toolkit (Wolf et al., 2020) is used in our study to fine-tune transformer-based models. We used a learning rate of $2e-5$ for optimizer Adam, batch size of 16, gradient accumulation of 1, and maximum sequence length of 256. BanglaBERT base version is trained on the BERT model, as a result, both BanglaBERT-base and BERT multilingual have 110M trainable parameters whereas BanglaBERT large is trained on the Electra model containing 335M parameters. For the transformer-based models, we run 3 epochs for all the models for better understanding. All models are trained on both tokenized and non-tokenized data and the change in performances is little to no on tokenized and non-tokenized data. To feed the non-tokenized data into the model, we added all the vocab of the dataset set to the pre-trained tokenizer which uses a Byte-Pair Encoding (BPE) tokenizer. As a result, we managed to ignore the default behavior of the BPE tokenizer, and the words were not tokenized by the BPE tokenizer. The representation of our tokenized and non-tokenized data is shown in Figure 1 and Figure 2 respectively.

Traditional Models: In order to train the traditional models, we first create tf-idf vectors with weighted n-gram from the preprocessed data. To use the contextual information, we utilized unigram, bigram, and trigram as part of weighted n-gram. We extract a fixed number of features (1,500) from the data and feed it to the models. Both models are trained on both tokenized and non-tokenized data and the performances remain the same on tokenized and non-tokenized data.

4 Results and Discussion

The official overall ranking and results determined by the lab organizers are presented in Table 2. The official evaluation metric for task 2 is F1-micro. We also presented the best system and baseline results (majority, random) including our system in Table 2. The last submission is considered for the leaderboard and our last submission is the BanglaBERT base model. In the competition, we officially ranked 5th position with an F1-micro score of 71.64 where the best system provides an

Input text: **নতুন সেনাপ্রধান লে. জে. এস এম শফিউদ্দিন আহমেদ **২৪ জুন দায়িত্ব নেবেন এস এম শফিউদ্দিন আহমেদ

Tokenized Text: ['**নতুন', 'সেনাপ্রধান', 'লে.', 'জে.', 'এস', 'এম', 'শফিউদ্দিন', 'আহমেদ', '**২৪', 'জুন', 'দায়িত্ব', 'নেবেন', 'এস', 'এম', 'শফিউদ্দিন', 'আহমেদ']

Encoded Text: [2, 32736, 21384, 32737, 32738, 1880, 1611, 32739, 3232, 32740, 3702, 2140, 7453, 1880, 1611, 32739, 3232, 3]

Figure 2: Representation of non-tokenized training text of id: 30960

Text: Pranoy Sen তখন পাকিস্তান ও আফগানিস্তান ভারতের হয়ে যাবে ।
Gold Label: Neutral
Predicted Label: Negative

Text: বিশ্বে উৎপাদিত করোনা টিকার বেশিরভাগই ব্যবহার করেছে... বিস্তারিত নিউজে
Gold Label: Neutral
Predicted Label: Positive

Figure 3: Example of sentences with wrong predictions for neutral class by BanglaBERT model.

F1 score of 73.10. Our system also performed better than both the majority and random baseline with a large margin of 21.87 and 38.08 respectively.

The detailed results of all the performed experiments are presented in Table 3. Once the submission period was over and the test set with labels became available, we conducted all the experiments again and reported the comprehensive findings. As shown in the reported results, we can state that the BanglaBERT approach with tokenized data outperforms other experiments by providing an accuracy of 71.64 with respect to the positive class F1 score of 75.59. With non-tokenized data, BanglaBERT gives an accuracy of 71.49 where the F1 score with respect to positive class is 75.18. Across the datasets, there is a definite tendency for the tokenized dataset to give better performances while evaluating than non-tokenized data. The performance between tokenized and non-tokenized data before feeding into networks for BERT-based models is little to none and for the traditional models, the performances remain the same. As a result of this, we can conclude that the importance of tokenization before feeding into the models is diminishing in the realm of the pretrained language models because each pretrained language model uses a model-specific tokenizer.

In table 3, all the models struggle to identify whether the data is in the neutral class because neutral class data are highly correlated with either positive class or negative class data, making it difficult for the models. Among all the models, both traditional models poorly perform to predict the neutral class. Although the BERT-based models perform well in comparison with traditional models on neutral class, the results are not comparable with the other two classes. We also explored the model performances for predicting neutral class and we came up with interesting findings which include if the text contains the words from frequently occurring words of the positive class, the text is classified as positive or negative if the frequently occurring words belong to negative class. We present two examples where our model couldn't predict neutral classes in Figure 3 for a better understanding of our findings.

Model	F1-micro	Rank
BanglaBERT	**71.64**	5^{th}
Best system	73.10	1^{st}
Baseline (Majority)	49.77	25^{th}
Baseline (Random)	33.56	29^{th}

Table 2: Official results on the test set and overall ranking of Task 2: Sentiment Analysis. **Bold** indicates our systems.

In our study, we found that monolingual pretrained language provides superior performance compared with the multilingual pretrained language model. We achieved an F1 score of 66.81 with respect to the positive class using BERT-multilingual while the BanglaBERT-base model has an F1 score of 75.59 with respect to the positive class which demonstrates the superior performance of the monolingual pretrained language model. We also observed that the base model outperforms the large model. The large model has more trainable parameters than the base model and the amount of

data is not sufficient to train and overfit the large model.

L	Model	Acc	P	R	F1
Neg	SVM*	54.76	58.72	73.46	65.26
Neu			37.08	09.55	15.19
Pos			49.91	52.53	51.19
Neg	SVM	54.76	58.72	73.46	65.26
Neu			37.08	09.55	15.19
Pos			49.91	52.53	51.19
Neg	RF*	55.42	58.65	76.18	66.28
Neu			41.33	12.69	19.41
Pos			51.14	48.37	49.72
Neg	RF	55.42	58.65	76.18	66.28
Neu			41.33	12.69	19.41
Pos			51.14	48.37	49.72
Neg	M1*	71.49	77.16	79.66	78.39
Neu			49.16	41.19	44.82
Pos			73.48	76.96	75.18
Neg	M1	71.64	78.77	80.77	78.39
Neu			48.88	37.59	42.50
Pos			73.44	77.87	**75.59**
Neg	M2*	70.61	77.30	78.16	77.73
Neu			48.53	38.84	43.15
Pos			70.61	77.96	74.10
Neg	M2	70.66	76.60	78.34	77.46
Neu			48.83	40.72	44.41
Pos			71.99	76.67	74.26
Neg	M3*	64.95	71.49	73.10	72.29
Neu			43.72	39.55	41.53
Pos			65.97	67.45	66.70
Neg	M3	65.01	71.50	73.07	72.28
Neu			44.24	38.76	41.32
Pos			65.50	68.16	66.81

Table 3: Detail results on the test set of **Task 2: Sentiment Analysis**. **Bold** indicates the best F1 score for positive class. * indicates the model trained and evaluated on non-tokenized data. L: Label, P: Precision, R: Recall, F1: F1-score, Neg: Negative, Neu: Neutral, Pos: Positive, M1: BanglaBERT, M2: BanglaBERT large, M3: BERT multilingual.

5 Conclusion

In this study, we run comparative experiments and analysis on the Bangla sentiment dataset provided by the task organizers of BLP-2023. We presented a detailed comparison of the fine-tuned models along with traditional models. Comparing the traditional model, we found that SVM outperforms RF with a margin of 1.47%. BanglaBERT outperforms all the models we used in our study. Our study also reveals that tokenization has little to no control over performance during the use of pretrained language models. In the submission of task 2 on the Sentiment Analysis dataset, we ranked 5^{th} position among all the participants. To extend this work, we will employ large language models (LLMs) and GPT-based models for comparative and in-depth sentiment analysis.

Limitations

The pretrained language models show promising performances toward tackling the sentiment analysis problem presented for this shared task. However, our models keep failing to predict neutral class, and we overfit the larger models (i.e., BanglaBERT large). Although we perform different hyperparameter tuning and dropouts for all the models, we are not able to find the optimal hyperparameters for each model. As a result, we decided to use the constant hyperparameter for all the models which causes overfitting the large model.

References

Firoj Alam, Arid Hasan, Tanvirul Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Vuk Batanović, Boško Nikolić, and Milan Milosavljević. 2016. Reliable baselines for sentiment analysis in resource-limited languages: The serbian movie review dataset. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2688–2696.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. 2022. Senticnet 7: A commonsense-based neurosymbolic ai framework for explainable sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3829–3839.

Rumman Rashid Chowdhury, Mohammad Shahadat Hossain, Sazzad Hossain, and Karl Andersson. 2019. Analyzing sentiment of movie reviews in bangla by applying machine learning techniques. In *2019 international conference on bangla speech and language processing (ICBSLP)*, pages 1–6. IEEE.

Shaika Chowdhury and Wasifa Chowdhury. 2014. Performing sentiment analysis in Bangla microblog posts. In *2014 International Conference on Informatics, Electronics Vision (ICIEV)*, pages 1–6.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero-and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv preprint arXiv:2308.10783*.

Md Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020a. Sentiment classification in bangla textual content: A comparative study. In *2020 23rd international conference on computer and information technology (ICCIT)*, pages 1–6. IEEE.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020b. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623, Online. Association for Computational Linguistics.

Asif Hassan, Mohammad Rashedul Amin, N Mohammed, and AKA Azad. 2016. Sentiment analysis on bangla and romanized bangla text (brbt) using deep recurrent models. *arXiv preprint arXiv:1610.00369*.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. Sentnob: A dataset for analysing sentiment on noisy bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271.

Md Ekramul Islam, Labib Chowdhury, Faisal Ahamed Khan, Shazzad Hossain, Sourave Hossain, Mohammad Mamun Or Rashid, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023. SentiGOLD: A large bangla gold standard multi-domain sentiment analysis dataset and its evaluation. *arXiv preprint arXiv:2306.06147*.

Md Saiful Islam, Md Ashiqul Islam, Md Afjal Hossain, and Jagoth Jyoti Dey. 2016. Supervised approach of sentimentality extraction from bengali facebook status. In *2016 19th international conference on computer and information technology (ICCIT)*, pages 383–387. IEEE.

Mohsinul Kabir, Obayed Bin Mahfuz, Syed Rifat Raiyan, Hasan Mahmud, and Md Kamrul Hasan. 2023. Banglabook: A large-scale bangla dataset for sentiment analysis from book reviews. *arXiv preprint arXiv:2305.06595*.

Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomforest. *R news*, 2(3):18–22.

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, Nedjma Ousidhoum, David Ifeoluwa Adelani, Seid Muhie Yimam, Ibrahim Sa'id Ahmad, Meriem Beloucif, Saif Mohammad, Sebastian Ruder, et al. 2023. Afrisenti: A twitter sentiment analysis benchmark for african languages. *arXiv preprint arXiv:2302.08956*.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2515–2519.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *Mining Intelligence and Knowledge Exploration: Third International Conference, MIKE 2015, Hyderabad, India, December 9-11, 2015, Proceedings 3*, pages 650–655. Springer.

J. Platt. 1998. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. MIT Press.

Md Atikur Rahman and Emon Kumar Dey. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2):15.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6. IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, EMNLP '20, pages 38–45, Online. Association for Computational Linguistics.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2022. Complementary explanations for effective in-context learning. *arXiv preprint arXiv:2211.13892*.

Team Error Point at BLP-2023 Task 2: A Comparative Exploration of Hybrid Deep Learning and Machine Learning Approach for Advanced Sentiment Analysis Techniques

Rajesh Kumar Das, Kabid Yeiad, Moshfiqur Rahman Ajmain,
Jannatul Maowa, Mirajul Islam, Sharun Akter Khushbu
Department of Computer Science and Engineering
Daffodil International University, Dhaka, Bangladesh
{rajesh15-13032, yeiad15-14440, moshfiqur15-14090, jannatul15-14095,
merajul15-9627, sharun.cse}@diu.edu.bd

Abstract

This paper presents a thorough and extensive investigation into the diverse models and techniques utilized for sentiment analysis. What sets this research apart is the deliberate and purposeful incorporation of data augmentation techniques with the goal of improving the efficacy of sentiment analysis in the Bangla language. We systematically explore various approaches, including preprocessing techniques, advanced models like Long Short-Term Memory (LSTM) and LSTM-CNN (Convolutional Neural Network) Combine, and traditional machine learning models such as Logistic Regression, Decision Tree, Random Forest, Multi-Naive Bayes, Support Vector Machine, and Stochastic Gradient Descent. Our study highlights the substantial impact of data augmentation on enhancing model accuracy and understanding Bangla sentiment nuances. Additionally, we emphasize the LSTM model's ability to capture long-range correlations in Bangla text. Our system scored 0.4129 and ranked 27th among the participants.

1 Introduction

Sentiment analysis, the process of extracting emotional information from textual data, has witnessed significant advancements in recent years. Our participation in the Sentiment Analysis Shared Task-2 at the BLP Workshop during EMNLP 2023 underscores our progress in Bangla Language Processing (BLP) and sentiment analysis (Hasan et al., 2023a). This study arises from the critical need to address sentiment expression issues specific to Bangla, a language with distinct linguistic nuances. Additionally, with the proliferation of Bangla content online, effective sentiment analysis tools are invaluable for applications ranging from social media monitoring to customer feedback analysis. (Jahan et al., 2021) Pronoun Replacement-Based Special Tagging System (PRS-TS) highlights context-specific language, improving Bangla sentiment analysis. The use of a Broad Multitask Transformer Network (BMT-Net) showed that multitask learning works in sentiment analysis (Zhang et al., 2022). (Zhang and Qian, 2020) Convolution over Hierarchical Syntactic and Lexical Graphs revealed ways to use syntactic and lexical information for aspect-level sentiment analysis. (Zhang et al., 2020) Convolutional multi-head self-attention on memory improved aspect sentiment categorization. The fusion strategy by (Zhou et al., 2020) for hate speech detection and the augmentation of BERT representations with context-aware embedding demonstrate contextual embeddings potential in sentiment analysis (Li and et al., 2020). (Hosain Sumit et al., 2018) Bangla Sentiment Analysis uses word embeddings to adapt to different languages. Long Short-Term Memory (LSTM) networks in hardware-accelerated sentiment analysis have also expanded this field (Wen and et al., 2021). Twitter is a popular social media tool for sentiment research. (Sigirci et al., 2020) use of heterogeneous multi-layer network representation and embedding shows new ways to look at unstructured textual data.

Our comprehensive study uses conventional preprocessing methods, advanced models like Long Short-Term Memory (LSTM) and LSTM-CNN Combine, and traditional machine learning models like Logistic Regression, Decision Tree, Random Forest, Multi-Naive Bayes, Support vector machine (SVM), and Stochastic gradient descent (SGD). Deliberate data augmentation is a hallmark of our study. Strategic augmentation has improved our dataset and sentiment analysis approaches, demonstrating data augmentation's ability to improve model accuracy and illuminate Bangla sentiment expression. We analyse LSTM and LSTM-CNN models with and without data augmentation as our main focus. We use dataset partition, performance evaluation criteria, and extensive per-class analysis in our experiments. The following discussion emphasises data augmentation's importance for model

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 331–335

efficacy. Comparing LSTM models to combined LSTM-CNN models shows that the former captures long-range correlations in Bangla text better, advancing Bangla sentiment analysis research. [1] final implementation with an anonymous GitHub link[2].

2 Literature Review

Recent studies in sentiment analysis, particularly in Bangla Language Processing (BLP), have catalysed the field (Hasan et al., 2023b). A key aspect of this progress lies in the development of specialised techniques for Bangla sentiment analysis. (Ritu et al., 2018) showed how word embeddings can be used in different linguistic settings. Another study by (Rahman et al., 2020) looked into more complex models, specifically how to group opinions in Bangla sentences. Considering structural aspects in sentiment analysis, (Tuhin et al., 2019) engineered an automated system for sentiment analysis from Bangla text using supervised learning techniques. (Abdalla and Özyurt, 2021) underscored the flexibility of deep learning techniques through a comprehensive sentiment analysis spanning various domains. Innovative methodologies are exemplified by (Zhu et al., 2018) bi-directional LSTM-CNN model, placing emphasis on fine-grained sentiment information extraction. (Wang et al., 2020) introduced an emotion-semantic-enhanced bidirectional LSTM with a multi-head attention mechanism for microblog sentiment analysis, showcasing the potential of attention mechanisms.(Luan and Lin, 2019) demonstrated the effectiveness of convolutional and recurrent neural network models for sentiment analysis tasks. (Hasan et al., 2023a) comparative study on modeling approaches for Bangla Sentiment Analysis yielded valuable insights. Moreover, (Islam et al., 2021) introduced SentNoB, a valuable resource for scrutinizing sentiment in informal and noisy textual data. Finally, (Zhou et al., 2016) integrated bidirectional LSTM with two-dimensional max pooling, showcasing the potential of amalgamating techniques for sentiment analysis tasks.

[1] https://github.com/blp-workshop/blp_task2#leaderboard

[2] https://anonymous.4open.science/r/EMNLP_2023_BLP_Workshop_Task2-46AE

3 Data and Methodology

Within the section, we provide a comprehensive overview of the data sources utilized and the rigorous research methodologies employed, ensuring transparency and credibility in our approach.

3.1 Dataset Description

Our study utilized the dataset sourced from BLP-2023 Task 2 (Hasan et al., 2023b) with the objective of discerning the sentiment expressed within textual content. This task involves the classification of sentiment into three categories: positive, negative, or neutral, thereby presenting a multi-class classification challenge. In Table 1, we present an overview of the dataset distribution used for experimentation in this shared task.

Table 1: Data splits and distributions of Shared Task-2

Class Label	Train	Dev	Test	Total
Negative	15767	1753	3338	20858
Positive	12364	1388	2092	15844
Neutral	7135	793	1277	9205
Total	35266	3934	6707	45907

Table 2: Dataset Split for Machine Learning Algorithms with and without Augmentation

Data Augmentation	Training Set Size	Testing Set Size	Total Dataset Size
No	20472	5118	25590
Yes	31379	7845	39224

Table 3: Dataset Split for Deep Learning Models with and without Data Augmentation

Data Augmentation	No	Yes
Training Set Size	16,377	19,433
Testing Set Size	5,118	6,073
Validation Set Size	4,095	4,859
Total Dataset Size	25,590	30,365

Table 2 presents the dataset partitioning for machine learning algorithms, highlighting distinctions between augmented and non-augmented data subsets. It offers a clear overview of the experimental design for model evaluation.

Table 3 shows a complete distribution of the deep learning dataset, separating augmented and non-augmented data segments. The academic setting relies on it to explain the experimental framework, especially for data augmentation. Figure 1 presents a word cloud representation for three sentiment categories: positive, negative, and neutral.

Figure 1: Word Cloud

3.2 Preprocessing

The BLP-2023 Task 2 dataset comprises two main components: the Multiplatform Bangla Sentiment (MUBASE) and SentNob datasets. The SentNob dataset encompasses public comments from various domains, including politics, education, and agriculture, sourced from news articles and videos. Meanwhile, the MUBASE dataset is a cross-platform compilation containing content from both Facebook and Twitter posts, all meticulously annotated to indicate sentiment polarity. As part of our preprocessing steps, we performed duplicate removal, filtered by text length, removed punctuation, links, emojis, non-character elements, and eliminated stopwords. We excluded very short or extremely long texts to focus on those that provide meaningful insights. Short texts might lack context, while overly long ones could introduce noise. In the process of removing stopwords, we systematically eliminate common, non-informative words to enhance the text's focus on meaningful content.

3.3 Algorithms

In our classification experiments, we employed a dual approach, encompassing both deep learning models and traditional machine learning algorithms like logistic regression (Nick and Campbell, 2007), decision trees (Kotsiantis, 2013), random forests (Rigatti, 2017), multi-naive bayes (Rish, 2001), SVM (Yang et al., 2012), and SGD (Chauhan et al., 2013). Specifically, within the domain of deep learning, we utilized the Long Short-Term Memory (LSTM) (Yu et al., 2019) model as well as a hybrid model combining LSTM and the Convolutional Neural Network (CNN) architecture (Li et al., 2021). This comprehensive approach allowed us to harness the strengths of both traditional and state-of-the-art methodologies, enhancing the depth and breadth of our analytical exploration.

3.4 Experimental Setup

In order to train the traditional models, we commenced by transforming the preprocessed data into TF-IDF vectors, integrating weighted n-grams, encompassing unigrams, bigrams, and trigrams. This approach was adopted to harness contextual information effectively. To address class imbalance, we implemented an up-sampling technique specifically focused on the neutral class within the merged dataset. We have used the train_test_split method from scikit-learn to organize the data for machine learning. This method divides the data into two parts: one for training (80%) and one for testing (20%). The parameters were selected to optimize model performance and ensure robustness in our deep learning-based classification approach listed in Table 6.

4 Results and Discussion

In this section, we present the outcomes of our experiments and engage in a comprehensive analysis of the findings.

Table 4: Performance scores for ML Models (With Augmentation)

Model Name	Accuracy	Precision	Recall	F1 Score
Logistic Regression	71.88	72.52	71.88	71.50
Decision Tree	65.29	64.79	65.29	64.67
Random Forest	72.36	73.36	72.36	71.79
Multi. Naive Bayes	71.22	72.51	71.22	70.83
SVM	75.02	75.26	75.02	74.85
SGD	60.84	65.69	60.84	59.34

Table 5: Performance scores for ML Models (Without Augmentation)

Model Name	Accuracy	Precision	Recall	F1 Score
Logistic Regression	64.20	66.81	64.20	59.55
Decision Tree	55.84	55.91	55.84	55.87
Random Forest	61.65	60.36	61.65	59.74
Multi. Naive Bayes	62.84	62.97	62.84	62.89
SVM	65.89	66.03	65.89	62.30
SGD	59.44	69.29	59.44	52.47

Table 4 displays machine learning model scores with data augmentation. SVM excels with 75.02% accuracy, showcasing its prowess in handling large datasets, clear separation, and noise robustness for

Table 6: Experimental setup for both DL models

Model	Data Augmentation	Embedding Dimension	Input Length	Vocabulary Size	Number of Classes	Batch Size	Number of Epochs
LSTM	No	128	300	5,000	3	64	50
LSTM	Yes	128	300	5,000	3	64	50
LSTM-CNN	No	128	300	5,000	3	64	50
LSTM-CNN	Yes	128	300	5,000	3	64	50

Table 7: Performance scores for Deep Learning Models

Model	Augmentation	Class	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
LSTM	With	Positive	70.94	64.45	67.54	68.43
		Negative	70.52	78.24	74.18	
		Neutral	63.04	63.07	63.06	
LSTM-CNN	With	Positive	67.85	64.79	66.29	67.59
		Negative	71.88	77.16	74.43	
		Neutral	62.25	60.97	61.60	
LSTM	Without	Positive	65.91	65.88	65.89	58.89
		Negative	36.88	30.64	33.47	
		Neutral	59.22	64.22	61.62	
LSTM-CNN	Without	Positive	64.01	67.90	65.90	57.74
		Negative	34.28	37.67	35.93	
		Neutral	62.94	55.03	58.72	

sentiment analysis. In contrast, SGD underperforms at 60.84% accuracy, indicating challenges with complex datasets or potential tuning requirements. Table 5 displays machine learning model performance metrics without data augmentation. SVM leads with 65.89% accuracy, validating its effectiveness in sentiment classification. In contrast, SGD underperforms with 59.44% accuracy, suggesting difficulties in handling dataset complexity without data augmentation. Table 7 summarizes deep learning model performance. "With Augmentation," LSTM excels in positive sentiment accuracy at 68.43%, and LSTM-CNN leads with 67.59% in negative sentiment accuracy. "Without Augmentation," LSTM's positive accuracy drops to 58.89%, and LSTM-CNN achieves 57.74% in negative sentiment, showing data augmentation's benefit.

5 Conclusion

This research offers a comprehensive examination of sentiment analysis in Bangla. It explores various models and techniques, traditional and advanced, with and without data augmentation. While not specifying accuracy rates, data augmentation notably boosts model effectiveness. Our study underscores the importance of addressing Bangla's unique challenges in sentiment analysis and the role of data augmentation. Comparative analysis between LSTM and LSTM-CNN models reveals LSTM's proficiency in capturing long-range correlations in Bangla text. These findings advance Bangla sentiment analysis and lay the groundwork for future research in this field.

References

G. Abdalla and F. Özyurt. 2021. Sentiment analysis of fast food companies with deep learning models. *The Computer Journal*, 64(3):383–390.

H. Chauhan, V. Kumar, S. Pundir, and E.S. Pilli. 2013. A comparative study of classification techniques for intrusion detection. In *2013 International Symposium on Computational and Business Intelligence*, pages 40–43. IEEE.

M. A. Hasan, S. Das, A. Anjum, F. Alam, A. Anjum, A. Sarker, and S. R. H. Noori. 2023a. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv [Cs.CL]*. Http://arxiv.org/abs/2308.10783.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023b. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

S. Hosain Sumit, M. Zakir Hossan, T. Al Muntasir, and T. Sourov. 2018. Exploring word embedding for bangla sentiment analysis. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5.

K. I. Islam, S. Kar, M. S. Islam, and M. R. Amin. 2021. Sentnob: A dataset for analysing sentiment on noisy bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271.

Busrat Jahan, Md. Ismail Emon, Sharmin Milu, Mohammad Hossain, and S. S. Mahtab. 2021. A pronoun replacement-based special tagging system for bengali language processing (blp). In *Proceedings of the Conference*, page 80.

S.B. Kotsiantis. 2013. Decision trees: a recent overview. *Artificial Intelligence Review*, 39:261–283.

X. Li and et al. 2020. Enhancing bert representation with context-aware embedding for aspect-based sentiment analysis. *IEEE Access*, 8:46868–46876.

Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.

Y. Luan and S. Lin. 2019. Research on text classification based on cnn and lstm. In *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*.

T.G. Nick and K.M. Campbell. 2007. Logistic regression. In *Topics in biostatistics*, pages 273–301. Springer.

Moqsadur Rahman, Summit Haque, and Zillur Rahman Saurav. 2020. Identifying and categorizing opinions expressed in bangla sentences using deep learning technique. *International Journal of Computer Applications (0975 – 8887)*, 176(17).

S.J. Rigatti. 2017. Random forest. *Journal of Insurance Medicine*, 47(1):31–39.

I. Rish. 2001. An empirical study of the naive bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, volume 3, pages 41–46.

Z.S. Ritu, N. Nowshin, M.M.H. Nahid, and S. Ismail. 2018. Performance analysis of different word embedding models on bangla language. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5.

I.O. Sigirci, H. Özgür, A. Oluk, H. Uz, E. Çetiner, H.U. Oktay, and K. Erdemir. 2020. Sentiment analysis of turkish reviews on google play store. In *2020 5th International Conference on Computer Science and Engineering (UBMK)*, pages 314–315.

R.A. Tuhin, B.K. Paul, F. Nawrine, M. Akter, and A.K. Das. 2019. An automated system of sentiment analysis from bangla text using supervised learning techniques. In *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, pages 360–364.

S. Wang, Y. Zhu, W. Gao, M. Cao, and M. Li. 2020. Emotion-semantic-enhanced bidirectional lstm with multi-head attention mechanism for microblog sentiment analysis. *Information*, 11(5):280.

S. Wen and et al. 2021. Memristive lstm network for sentiment analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(3):1794–1804.

Y. Yang, J. Wang, and Y. Yang. 2012. Improving svm classifier with prior knowledge in microcalcification detection. In *2012 19th IEEE International Conference on Image Processing*, pages 2837–2840. IEEE.

Y. Yu, X. Si, C. Hu, and J. Zhang. 2019. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–70.

Mi Zhang and Tieyun Qian. 2020. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *Proceedings of the Conference*, pages 3540–3549.

T. Zhang, X. Gong, and C. L. P. Chen. 2022. Bmt-net: Broad multitask transformer network for sentiment analysis. *IEEE Transactions on Cybernetics*, 52(7):6232–6243.

Y. Zhang, B. Xu, and T. Zhao. 2020. Convolutional multi-head self-attention on memory for aspect sentiment classification. *IEEE/CAA Journal of Automatica Sinica*, 7(4):1038–1044.

P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.

Y. Zhou, Y. Yang, H. Liu, X. Liu, and N. Savage. 2020. Deep learning based fusion approach for hate speech detection. *IEEE Access*, 8:128923–128929.

Y. Zhu, X. Gao, W. Zhang, S. Liu, and Y. Zhang. 2018. A bi-directional lstm-cnn model with attention for aspect-level text classification. *Future Internet*, 10(12):116.

UFAL-ULD at BLP-2023 Task 2 Sentiment Classification in Bangla Text

Sourabrata Mukherjee[1], Atul Kr. Ojha[2], Ondřej Dušek[1]
[1]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
[2]Insight SFI Centre for Data Analytics, DSI, University of Galway, Ireland
{mukherjee,odusek}@ufal.mff.cuni.cz
atulkumar.ojha@insight-centre.org

Abstract

In this paper, we present the UFAL-ULD team's system for the BLP Shared Task 2: Sentiment Analysis of Bangla Social Media Posts. The Task 2 involves classifying text into Positive, Negative, or Neutral sentiments. As a part of this task, we conducted a series of experiments with several pre-trained sequence classification models – XLM-RoBERTa, BanglaBERT, Bangla BERT Base and Multilingual BERT. Among these, our best-performing model was based on the XLM-RoBERTa-base architecture, which outperforms baseline models. Our system was ranked 19th among the 30 teams that participated in the task.

1 Introduction

Sentiment analysis, the task of determining the sentiment expressed in textual data, is a critical component of natural language processing (NLP). It plays a vital role in understanding public opinion, social media trends, and user sentiments in various languages. In the context of the Bangla language, sentiment analysis poses unique challenges due to the language's specific characteristics such as complex morphology, making it an intriguing area of research. Sentiment analysis classification in Bangla remains less explored in comparison to English for reasons ranging from the non-availability of the datasets to lack of development of models or shared tasks (Rosenthal et al., 2017; Patwa et al., 2020; Barnes et al., 2022).

This paper presents our system developed for the BLP Shared Task 2 (Hasan et al., 2023a,b). Our objective is to provide a comprehensive description of our approach and results. We aim to contribute valuable insights and techniques to the field of Bangla sentiment analysis. Our team conducted a series of experiments utilising several pre-trained sequence classification models where we contributed to fine-tuning and hyper-parameter tuning to optimize the performance of these models.[1] We further employed focal loss to counter class imbalance in the data. The resulting system placed 19th out of 30 submissions in the shared task.

2 Related Works

Several works on Bangla sentiment were released prior to the present shared task challenge. Ali et al. (2020) introduces "BanglaSenti", a dataset of 61,582 Bangla words for sentiment analysis, originally developed for social media, with potential uses in emotion detection and opinion mining. This dataset's polarity categorizations are crucial for understanding sentence sentiment. Kabir et al. (2023) provides "BanglaBook," a dataset of 158,065 Bangla book reviews categorized as positive, negative, or neutral. Pre-trained models outperform manually crafted features, underscoring the need for more training resources in Bangla sentiment analysis, and an error analysis reveals common classification mistakes in under-resourced languages. Patra et al. (2015) focuses on Sentiment Analysis in Twitter for Indian languages, including Bangla, Hindi, and Tamil. The paper presents the first sentiment analysis attempt for these languages and ranks participating teams based on accuracy, achieving a maximum accuracy of 55.67% for Hindi. Bhowmick and Jana (2021) explores sentiment analysis in Bangla using pre-trained transformer models, achieving state-of-the-art performance with a maximum accuracy of 95% for a two-class sentiment classification task, setting a benchmark for Bangla sentiment analysis.

3 Dataset

The data for the BLP Shared Task 2 (Hasan et al., 2023a) is curated from two primary sources (Islam et al., 2021; Hasan et al., 2023a): MUltiplatform BAngla SEntiment (MUBASE) and SentNob

[1]Our code is available at https://github.com/souro/classification_tasks_bangla

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 336–339

datasets. These datasets collectively contribute diverse textual data for sentiment analysis in the Bangla language. Each item of the data is annotated as having a positive, neutral or negative sentiment. Automating such annotation is the subject of BLP Shared Task 2.

The MUBASE dataset offers a substantial multiplatform collection. It includes a range of textual data, such as Tweets and Facebook posts, each meticulously annotated with their respective sentiment polarity labels. The SentNob dataset comprises public comments extracted from various social media platforms, specifically associated with news and video content. This dataset encompasses a broad spectrum of 13 distinct domains, spanning subjects such as politics, education, and agriculture.

The datasets were separated into sections for training, development, development-test and test (for final evaluation), consisting of 35,266, 3,934, 3,426 and 6,707 comments respectively.[2]

4 Experiments

This section provides a detailed description of our system's design, methodology, and the steps taken to achieve competitive results in the BLP Shared Task 2.

4.1 Data Preprocessing and Cleaning

Our system began with data preprocessing and cleaning using techniques provided by BNLP Shared Task 2 organisers (Hasan et al., 2023a). In addition to basic processing, we paid special attention to Unicode handling: We used the Bangla NLP toolkit to fix and normalize all Unicode characters into the NFKC normalization form.[3]

4.2 Model Selection

To develop our sentiment analysis system, we experimented with several pre-trained sequence classification models in a constrained setting.

We fine-tuned these models using various pretrained masked language models derived from the BERT (Devlin et al., 2019) architecture: XLM-RoBERTa (base and large versions) (Conneau et al., 2019), BanglaBERT '' (Bhattacharjee et al., 2022), Bangla BERT Base (Sarker, 2020) and BERT-base-multilingual-cased (Devlin et al., 2018). Among these, our best-performing model was based on the XLM-Roberta-base (Conneau et al., 2019) architecture. [4]

4.3 Hyperparameter Tuning

We conducted hyperparameter tuning on the development data to optimize our model's performance. The best hyperparameter settings we utilized are as follows: batch size 5, learning rate 1e-5, AdamW optimizer (Loshchilov and Hutter, 2019), 15 epochs gradient clipping (`clip_grad_norm`) 1.0, weight decay 0.01, dropout rate 0.1.

4.4 Handling Class Imbalance: Focal Loss

To address class imbalance issues, we experimented with oversampling and undersampling techniques. Although we obtained promising results using these methods, we eventually discovered an alternative approach that improved our results even further – focal loss.

Focal loss (Lin et al., 2017) is a specialized loss function designed to address class imbalance and focus on hard-to-classify examples. Specifically, we used the following parameters for the focal loss: $\alpha = 1, \gamma = 2$. Focal loss provided superior results compared to the simple cross-entropy loss, leading us to integrate it into our best-performing model.

5 Results

The official evaluation metric for the BLP Shared Task 2 is micro-F1 (Hasan et al., 2023a). This metric is commonly used in multi-class classification tasks, including sentiment analysis. It combines precision and recall to provide an overall assessment of the system's performance across all sentiment classes. The performance of our sentiment analysis system in the BLP Shared Task 2 is a testament to the effectiveness of our approach. We achieved a micro-F1 score of 0.6768 (see Table 1) on the evaluation dataset, substantially outperforming baselines, but placing 19th in the shared task out of 30 entries.

6 Conclusion

In this paper, we detailed the approach and methodologies we used for the BLP Shared Task 2: Senti-

[2] https://github.com/blp-workshop/blp_task2/tree/main/data

[3] We use the `fix_unicode=True`, `unicode_norm=True` and `unicode_norm_form="NFKC"` parameters.

[4] We use the models from HuggingFace: https://huggingface.co/xlm-roberta-base, https://huggingface.co/xlm-roberta-large, https://huggingface.co/csebuetnlp/banglabert, https://huggingface.co/sagorsarker/bangla-bert-base, https://huggingface.co/bert-base-multilingual-cased.

Model	micro-F1
Random Baseline	0.3356
Majority Baseline	0.4977
n-gram Baseline	0.5514
BLP Shared Task 2 winning system	0.7370
Our system	0.6768

Table 1: UFAL-ULD team and baseline systems results

ment Analysis of Bangla Social Media Posts. Our system demonstrated a strong performance, achieving a micro-F1 score of 0.6768, signifying its proficiency in classifying Bangla social media posts into Positive, Negative, and Neutral sentiments. Our team's system was ranked 19th among the 30th teams that participated in the task.

Through data preprocessing, model selection, hyperparameter tuning, and the incorporation of advanced techniques such as Focal Loss, we optimized our system to excel in sentiment analysis tasks for the Bangla language. Our results substantially outperformed baseline models, underscoring the effectiveness of our strategies.

Our work contributes to the advancement of sentiment analysis in Bangla social media, enabling a deeper understanding of user sentiment and trends in the Bangla-speaking community. We believe that our system's success marks a significant step toward improving sentiment analysis in underrepresented languages, and we look forward to further advancements in this field.

Limitations

While our system achieved competitive results, it is essential to acknowledge its limitations. First, our system's performance may vary depending on the specific characteristics of the social media posts and the domains they pertain to. Further fine-tuning and adaptation may be required for specialized applications.

Second, the dataset used for this shared task, although diverse, may not encompass the full breadth of Bangla social media discourse. As such, our system's performance may be influenced by potential biases in the training data.

Finally, while we have strived to optimize our system's performance, there may be room for further improvements through the exploration of alternative models, techniques, or additional linguistic resources.

Ethics Statement

We adhere to ethical guidelines in conducting our research and participating in the BLP Shared Task 2. We have respected privacy and data protection principles throughout our work, ensuring that any data used in our experiments adheres to appropriate consent and privacy regulations.

Furthermore, our system's output is intended for research and analysis purposes only. We emphasize responsible use and interpretation of sentiment analysis results, recognizing that automated sentiment analysis tools can influence decision-making and public perception.

We are committed to transparency and open collaboration in the field of natural language processing and sentiment analysis. We encourage ethical research practices and advocate for fairness, accountability, and transparency in AI and NLP technologies.

Acknowledgements

This research was supported by the European Research Council (Grant agreement No. 101039303 NG-NLG) and by Charles University projects GAUK 392221 and SVV 260575. We acknowledge of the use of resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).
Atul Kr. Ojha would like to acknowledge the support of the Science Foundation Ireland (SFI) as part of Grant Number SFI/12/RC/2289_P2 Insight_2, Insight SFI Centre for Data Analytics.

References

Hasmot Ali, Md. Fahad Hossain, Shaon Bhatta Shuvo, and Ahmed Al Marouf. 2020. Banglasenti: A dataset of bangla words for sentiment analysis. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–4.

Jeremy Barnes, Laura Oberlaender, Enrica Troiano, Andrey Kutuzov, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, and Erik Velldal. 2022. SemEval 2022 task 10: Structured sentiment analysis. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1280–1295, Seattle, United States. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya

Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Anirban Bhowmick and Abhik Jana. 2021. Sentiment analysis for Bengali using transformer based models. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 481–486, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, Minneapolis, MN, USA. ArXiv: 1810.04805.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mohsinul Kabir, Obayed Bin Mahfuz, Syed Rifat Raiyan, Hasan Mahmud, and Md Kamrul Hasan. 2023. BanglaBook: A large-scale Bangla dataset for sentiment analysis from book reviews. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1237–1247, Toronto, Canada. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR*, New Orleans, LA, USA. arXiv. ArXiv:1711.05101 [cs, math].

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (SAIL) tweets - an overview. In *Mining Intelligence and Knowledge Exploration - Third International Conference, MIKE 2015, Hyderabad, India, December 9-11, 2015, Proceedings*, volume 9468 of *Lecture Notes in Computer Science*, pages 650–655. Springer.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790, Barcelona (online). International Committee for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Embeddings at BLP-2023 Task 2: Optimizing Fine-Tuned Transformers with Cost-Sensitive Learning for Multiclass Sentiment Analysis

S.M Towhidul Islam Tonmoy
Islamic University of Technology, Gazipur , Bangladesh
towhidulislam@iut-dhaka.edu

Abstract

In this study, we address the task of Sentiment Analysis for Bangla Social Media Posts, introduced in first Workshop on Bangla Language Processing (Hasan et al., 2023a). Our research encountered two significant challenges in the context of sentiment analysis. The first challenge involved extensive training times and memory constraints when we chose to employ oversampling techniques for addressing class imbalance in an attempt to enhance model performance. Conversely, when opting for undersampling, the training time was optimal, but this approach resulted in poor model performance. These challenges highlight the complex trade-offs involved in selecting sampling methods to address class imbalances in sentiment analysis tasks. We tackle these challenges through cost-sensitive approaches aimed at enhancing model performance. In our initial submission during the evaluation phase, we ranked 9th out of 30 participants with an F1-micro score of 0.7088 . Subsequently, through additional experimentation, we managed to elevate our F1-micro score to 0.7186 by leveraging the BanglaBERT-Large model in combination with the Self-adjusting Dice loss function. Our experiments highlight the effect in performance of the models achieved by modifying the loss function. Our experimental data and source code can be found here.[1]

1 Introduction

Sentiment analysis is an important task in natural language processing that involves automatic detection of expressed opinions within text. The proliferation of online social media interactions has led to a surge in textual content, necessitating strategies to address associated challenges.

While sentiment analysis for high-resource languages like English has made significant progress, low-resource languages like Bangla are still lagging behind. Low resource languages have intricate sentence structures and grammatical rules, making the development of systems resource-intensive. Achieving optimal model performance requires substantial annotated data, leading to longer processing times as data volume increases. Additionally, when performing multiclass sentiment analysis, there is a common challenge related to class imbalance, which can lead to models exhibiting bias towards particular classes. Previous studies have concentrated on improving the quantity of training data instances, although this approach can extend the duration of model training.

Numerous studies have been undertaken to advance the development of linguistic resources for the Bangla language. (Islam et al., 2021) introduced SentNoB dataset for multiclass sentiment analysis task. (Patra et al., 2015) summarized the sentiment analysis task for three Indian language , namely Bangla, Hindi and Tamil. They showed the results for shared task on binary sentiment analysis and introduced the SAIL dataset. (Rezaul Karim et al., 2020) introduced the BengFastText dataset which was able to capture semantics of Bangla words. They experimented their corpus with traditional ML algorithms and also utilized MConv-LSTM network to tackle the binary sentiment analysis task. (Tripto and Ali, 2018) introduced Bangla language corpus from Bangla youtube comments. (Rahman et al., 2018) focused on aspect based sentiment analysis and introduced the research community with ABSA cricket and restaurant datasets. But all of this datasets had class imbalances in their classes. (Hasan et al., 2020) and (Alam et al., 2021) compiled all the previously mentioned datasets and benchmarked their results with different traditional and transformer based models. The ongoing challenge lies in the escalating fine-tuning time due to the increasing data volume. This study seeks to enhance fine-tuned transformer model efficiency

[1] https://github.com/towhidultonmoy/Bangla-Multiclass-Sentiment-Analysis-Shared-Task-.git

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 340–346

by employing cost-sensitive learning to tackle class imbalance problem. Our contributions can be summarized as follows:

- Cost sensitive learning improves the performance of most of the transformer based models. We perform an extensive series of experiments involving SOTA transformer models, exploring various loss functions.
- The best F1-micro score was achieved with **BanglaBERT-Large** variant combining it with self adjusting dice loss.
- Additionally, we examine the impact of diverse preprocessing techniques on model performance.

2 Related Work

2.1 Sentiment Classification with Deep Learning

In the context of text classification for sentiment analysis in Bangla, researchers have utilized a range of models, from traditional ones to the latest prompt-based large language models (LLMs).

(Rahman et al., 2018) employed SVM, RF, and KNN models to perform ABSA in Bangla. They achieved F1 scores of 0.37 and 0.42, respectively, using TF-IDF features on their ABSA cricket and restaurant datasets. (Rezaul Karim et al., 2020) explored a comprehensive set of models, including LR, NB, SVM, KNN, GBT, RF, MConv-LSTM, and MAE. They achieved impressive results with MConv-LSTM, attaining an MCC of 0.746 and an AUC of 0.87 for sentiment analysis in Bangla using BengFastText embeddings. (Hasan et al., 2023b) delved into zero- and few-shot in-context learning for sentiment analysis in Bangla. They compared Open LLMs like Flan-T5 and GPT-4 against fine-tuned models, where BanglaBERT outperformed others with a weighted F1 of 69.39. They utilized SentNoB and introduced the MUBASE dataset, which included Facebook posts and tweets. (Alam et al., 2021) conducted a comparative analysis of Bangla NLP tasks using transformer models, achieving an 82.0 weighted F1 using XLM-RoBERTa on various publicly available datasets. In their study, (Hasan et al., 2020) conducted comparative sentiment analysis on Bangla text using classical algorithms and deep learning models. BERT and XLM-RoBERTa demonstrated strong performance on different datasets, with an average weighted F1 of 0.671 and 0.653, respectively.

2.2 Handling Class Imbalance

(Hasib et al., 2023b) present a system that employs RUS and SMOTE to balance the dataset. Their approach utilizes a range of machine learning and deep learning models, with BERT reaching a maximum accuracy of 99.04% in balanced datasets and 72.23% in imbalanced datasets. Another noteworthy contribution by (Hasib et al., 2023a) introduces MCNN-LSTM, a novel fusion of CNN and LSTM for news text classification. After balancing the dataset using the Tomek-Link algorithm, their model attains remarkable performance, achieving a 98% F1-score and 99.71% accuracy compared to prior research. (Rafi-Ur-Rashid et al., 2022) address class imbalance using various models for binary sentiment analysis, achieving 0.94 accuracy with their CNN model on the original corpus, employing a comprehensive approach that includes data augmentation, focal loss functions, outlier detection, data resampling, and hidden feature extraction across diverse datasets. Lastly, (Ashrafi et al., 2020) introduce BERT-based deep learning models for Bangla NER while addressing class imbalance with a modified cost-sensitive loss function. Their proposed models yield 8% enhancement in F1 MUC score compared to previous Bangla NER research.

3 Dataset

3.1 Data Description

The dataset for this shared task is a combination of two sources: SentNoB (Hasan et al., 2020) and MUBASE (Hasan et al., 2023b). Table 1 reports the number of samples in the train, validation and test sets for each class. The dataset distribution reveals a noticeable class imbalance across the training, validation, and test sets.

Class	Train	Validation	Test
Negative	15767	1753	3338
Positive	12364	1388	2092
Neutral	7135	793	1277
Total	35266	3934	6707

Table 1: Class-wise Dataset Distribution in Train, Validation, and Test Sets.

4 System Overview

Recent developments in NLP have seen the emergence of pre-trained transformer models, based on

the transformer architecture proposed by (Vaswani et al., 2017). These models consistently achieve state-of-the-art performance across a wide range of NLP tasks.

In our study, we initially fine-tuned multiple pre-trained transformer models using the default cross-entropy loss as our baseline approach. Subsequently, we aimed to enhance model performance through cost-sensitive learning, which effectively addresses class imbalances and mitigates biases towards the majority classes.

4.1 Finetuning Pre-trained Language Models (PLMs)

We selected various pre-trained models and fine-tuned them for our baseline. These models include Bangla-Bert (Bhattacharjee et al., 2022), Bangla-GPT2(Flax Community, 2023), Indic-BERT (Kakwani et al., 2020) and mBERT (Devlin et al., 2018). We employed cross-entropy loss and the AdamW optimizer for fine-tuning. Details regarding the hyperparameter values used for training the baseline and subsequent models can be found in the Appendix.

4.2 Cost Sensitive Learning

A prominent challenge we encountered with our dataset was class imbalance, a common issue in machine learning tasks. However, conventional methods like oversampling and undersampling were not feasible in our case due to their drawbacks, which involve increased training times and reduced performance, respectively. Thus, we explored the hypothesis that modifying the loss function could potentially enhance model performance without the need for additional data.

To elevate our model's performance beyond the baseline, we introduced various loss functions, namely, the self-adjusting dice loss (Li et al., 2019), focal loss (Lin et al., 2017), and F1-micro loss. These alternative loss functions were employed as part of our strategy to address class imbalance and improve overall model performance. Details about this loss functions are mentioned in the appendix C

5 Experiments and Results

We explored various model and custom loss function combinations as described in Section 4. In this section, we outline the evaluation for the shared task competition, with the F1-micro score as the key performance metric. Our model assessments were conducted on the test set, and, as outlined in Section 6, we noted improved model performance without text preprocessing as mentioned in appendix 6.2. Table 2 presents the test set results, trained upon dataset B . Details about the dataset are mentioned in A.

In our initial experimentation with transformer models, we fine-tuned each model using the default cross-entropy loss function. Among the models in our baseline study, BanglaBERT-Large stood out, achieving the highest F1-micro score of 0.7101. Subsequently, we investigated the impact of cost-sensitive loss functions on model performance. We implemented focal loss, self-adjusting dice loss, and F1-micro loss. Notably, for two models, BanglaBERT-Large and mBERT, these alternative loss functions led to significant improvements compared to the baseline approach.

For BanglaBERT-Large, self-adjusting dice loss produced the best result, with an F1-micro score of 0.7186, surpassing all other transformer models used in our research. For mBERT, focal loss resulted in improved performance, achieving an F1-micro score of 0.6606. Other loss functions for these two models also outperformed the baseline, as shown in the table 2.

However, for BanglaGPT2, incorporating cost-sensitive loss functions did not enhance model performance; the baseline approach yielded the highest F1-micro score at 0.6788. Regarding the IndicBERT model, self-adjusting dice loss improved performance compared to the baseline cross-entropy loss, achieving an F1 score of 0.6263. However, focal loss and F1-micro loss did not yield performance improvements for this model.

6 Ablation Study

In the scope of our study, we conducted a sequence of experiments to understand key factors affecting our model's performance.

6.1 Impact of Combining Training and Validation Set

To evaluate the merging of training and development sets, we analyzed two datasets: Dataset A and Dataset B (the consolidated dataset). We then assessed their impact on the designated test dataset. Appendix A offers a detailed data distribution analysis for both datasets, and Table 3 summarizes the effect of these datasets on the performance of the most promising combinations from Table 2.

Label	Word Unigram Overlap
Negative	খারাপ (Bad), দোষ (Fault), ধর্ষণ (Rape), নিষিদ্ধ (Prohibited), যুদ্ধ (War), হামলা (Attack), গুম (Disappearance), ভুয়া (Fake), ধ্বংস (Destroy), প্রত্যাহার (Withdrawal), কষ্ট (Suffering), হত্যা (Murder), শান্তি (Peace), উন্নয়ন (Development), অবৈধ (Illegal), ভয় (Fear), ধন্যবাদ (Gratefulness), পরিবর্তন (Change), প্রাণহানি (Homicide), অভিযোগ (Complaint)
Neutral	ভয় (Fear), গুরুত্বপূর্ণ (Important), ভুল (Mistake), জয় (Victory), ঘুম (Sleep), হামলা (Attack), খারাপ (Bad), ধ্বংস (Destruction), উন্নয়ন (Development), গুম (Loss), ধর্ষণ (Assault), অবৈধ (Illegal), ভুয়া (Destruction), ধন্যবাদ (Gratefulness), দোষ (Fault), যুদ্ধ (War), কষ্ট (Suffering), প্রিয় (Favorite), আলহামদুলিল্লাহ (Gratitude), সুন্দর (Beautiful)
Positive	পরিবর্তন (Change), ঘুম (Sleep), প্রিয় (Favorite), আলহামদুলিল্লাহ (Gratitude), শান্তি (Peace), হত্যা (Murder), সুন্দর (Beautiful), খারাপ (Bad), উন্নয়ন (Development), গুরুত্বপূর্ণ (Important), ধ্বংস (Destruction) , খারাপ (Bad), ধন্যবাদ (Gratefulness) , নিষিদ্ধ (Prohibited), প্রাণহানি (Homicide), অভিযোগ (Complaint), নিষিদ্ধ (Prohibited), প্রত্যাহার (Withdrawal), যুদ্ধ (War), জয় (Victory)

Figure 1: Example of word unigram overlaps among label categories with English translations. Here distinct colors are used to emphasize concurrent words: color denotes common words across **all** labels, denotes common words between **Negative** and **Neutral** labels, color denotes common words between **Negative** and **Positive** labels, and denotes common words between **Neutral** and **Positive** labels.

Model	Loss Function	F1
BanglaBERT	Cross Entropy Loss	0.7101
	Focal Loss	0.7177
	SA Dice Loss	**0.7186**
	F1 Micro Loss	0.7126
Bangla GPT2	**Cross Entropy Loss**	**0.6788**
	Focal Loss	0.6757
	SA Dice Loss	0.6569
	F1 Micro Loss	0.6707
mBERT	Cross Entropy Loss	0.6497
	Focal Loss	**0.6606**
	SA Dice Loss	0.6528
	F1 Micro Loss	0.6581
IndicBERT	Cross Entropy Loss	0.6166
	Focal Loss	0.6062
	SA Dice Loss	**0.6263**
	F1 Micro Loss	0.6145

Table 2: F1-micro score on the Competition Test Set for Various Transformer Models Trained with Dataset B

6.2 Impact of Different Text Processing Techniques

In our study, we performed two crucial text preprocessing steps: **1)** removing emojis and **2)** eliminating punctuation marks. We assessed the effects of each step independently and when applied together. We've summarized the results in Table 4, using the acronyms: **P1** (for Step 1), **P2** (for Step 2), **All** (for Both Steps), and **None** (for No Preprocessing). This analysis sheds light on how these preprocessing methods impact our research outcomes.

Model	Loss Function	Dataset	F1
BanglaBERT	SA Dice Loss	A	0.7067
		B	**0.7186**
Bangla GPT2	Cross Entropy Loss	**A**	**0.6833**
		B	0.6788
mBERT	Focal Loss	A	0.6446
		B	**0.6606**
IndicBERT	SA Dice Loss	A	0.6230
		B	**0.6263**

Table 3: Impact of Diverse Datasets on Optimal Transformer Model Combinations. **Dataset A**: Original Training Set, **Dataset B**: Combined Train and Validation Sets.

7 Error Analysis

Table 2 present the performance results of BangaBERT-Large, which, notably, outperformed all other methods in our experiments. This section delves into a quantitative error analysis employing a confusion matrix, as displayed in Figure 2, focusing on the top-performing model. Our analysis reveals a distinct pattern of misclassification occurring primarily between the 'neutral' and 'negative' classes.

In Appendix D, Table 7 demonstrates the subpar performance observed in the 'neutral' class. Despite our diligent efforts to mitigate class imbalance

Dataset	BanglaBERT with SA Dice Loss
P1	0.7182
P2	0.7088
All	0.7106
None	**0.7186**

Table 4: F1-micro score for Different Preprocessing Techniques on Dataset B: Combined Train and Validation Sets

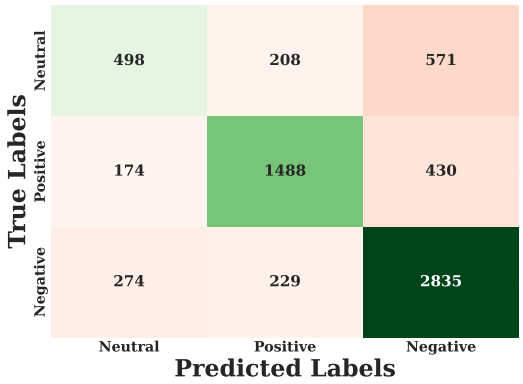


Figure 2: Confusion Matrix of Best Performing Model

through a cost-sensitive loss function, the model continues to encounter difficulties in distinguishing between 'neutral' and 'negative' labels.

Furthermore, this misclassification is influenced by semantic similarities between words across different classes. Figure 1 visually represents the common unigrams across various labels, highlighting the areas where the model exhibits errors, especially when there are concurrent words between the 'negative' and 'neutral' labels.

8 Conclusion

This research paper primarily emphasizes the enhancement of transformer-based models' performance through the application of cost-sensitive learning techniques, aimed at alleviating issues related to class imbalance and overfitting. Among various combinations of transformers and loss functions explored, the BanglaBERT model utilizing the self-adjusting dice loss exhibited the highest F1 score of 0.7186 on the test dataset. Although the combination of cost-sensitive techniques with transformer models led to notable enhancements in performance, it's important to highlight that the model's effectiveness still falls short, especially when it comes to the 'neutral' class.

Limitations

In this research, we chose a cost-sensitive approach as an alternative to augmentation of the training dataset, recognizing its resource-intensive demands in GPU resources and training time. Our objective was to investigate how modifying loss functions could improve the performance of fine-tuned transformer models, presenting a more resource-efficient route to better outcomes.

Despite our experiments demonstrating several strategies for enhancing fine-tuned transformer model performance, we acknowledge the model's ongoing challenge in accurately classifying less frequent classes. This limitation directs our future research towards optimizing loss function hyperparameters and assessing their effectiveness across various model architectures and datasets as a promising avenue for improvement.

References

Firoj Alam, Md Arid Hasan, Tanvir Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Imranul Ashrafi, Muntasir Mohammad, Arani Shawkat Mauree, Galib Md Azraf Nijhum, Redwanul Karim, Nabeel Mohammed, and Sifat Momen. 2020. Banner: a cost-sensitive contextualized model for bangla named entity recognition. *IEEE Access*, 8:58206–58226.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Flax Community. 2023. gpt2-bengali (revision cb8fff6).

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero-and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv preprint arXiv:2308.10783*.

Md. Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020. Sentiment classification in bangla textual content: A comparative study. In *23rd International Conference on Computer and Information Technology (ICCIT)*.

Khan Md Hasib, Sami Azam, Asif Karim, Ahmed Al Marouf, FM Javed Mehedi Shamrat, Sidratul Montaha, Kheng Cher Yeo, Mirjam Jonkman, Reda Alhajj, and Jon G Rokne. 2023a. Mcnn-lstm: Combining cnn and lstm to classify multi-class text in imbalanced news data. *IEEE Access*.

Khan Md Hasib, Nurul Akter Towhid, Kazi Omar Faruk, Jubayer Al Mahmud, and MF Mridha. 2023b. Strategies for enhancing the performance of news article classification in bangla: Handling imbalance and interpretation. *Engineering Applications of Artificial Intelligence*, 125:106688.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *Proc. of MIKE*, pages 650–655. Springer.

Md Rafi-Ur-Rashid, Mahim Mahbub, and Muhammad Abdullah Adnan. 2022. Breaking the curse of class imbalance: Bangla text classification. *Transactions on Asian and Low-Resource Language Information Processing*, 21(5):1–21.

Md Rahman, Emon Kumar Dey, et al. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2):15.

Md Rezaul Karim, Bharathi Raja Chakravarthi, Mihael Arcan, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced Bengali language based on multichannel convolutional-lstm network. *arXiv*, pages arXiv–2004.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *Proc. of ICBSLP*, pages 1–6. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A Dataset

We conducted experiments using two datasets, A and B, as described in our ablation study. The table 5 shows the number of examples in each split.

For dataset A, we utilized the original training and test sets. In dataset B, we combined the training and validation sets into a single unified training set, while keeping the test set unchanged.

Split	Class	Dataset A	Dataset B
Train	Negative	15767	17520
	Positive	12364	13752
	Neutral	7135	7928
Test	Negtaive	3338	3338
	Positive	2092	2092
	Neutral	1277	1277

Table 5: Class wise Dataset Distribution in Dataset A and Dataset B

B Model Training

In this section, we provide the hyperparameter values we used during fine tuning our models to facilate the reproducibility of our results at a later time. The acronyms correspond to:

- **LR** : Learning Rate
- **BS** : Batch Size
- **EP** : Epoch
- **WD** : Weight Decay
- **MP** : Mixed Precision

- **TML** : Tokenizer Max Length
- **ES** : Early Stopping
- **ESP** : Early Stopping Patience
- **FL** : Focal Loss (Gamma , Alpha)

Hyperparameter	BanglaBERT	BanglaGPT2	mBERT	IndicBERT
LR	2E-5	2E-5	2E-5	2E-5
BS	20	1	20	20
EP	20	20	20	20
WD	0.02	0.02	0.02	0.02
MP	True	True	True	True
TML	200	200	200	200
ES	True	True	True	True
ESP	3	3	3	3
FL	2,4	2,4	2,4	2,4

Table 6: Hyperparameter and Fine-Tuning Settings for Various Transformer Models in Our Experiment

C Loss functions

C.1 Self-adjusting Dice Loss

The Self-adjusting Dice Loss(Li et al., 2019) was introduced as an objective function for handling imbalanced datasets in NLP. It derives from the original dice coefficient, an F1-oriented metric for measuring set similarity. This loss function, based on a modified dice coefficient, was reported to yield superior F1 scores compared to models trained with cross-entropy loss.

$$DiceLoss = 1 - \frac{2(1-p_{n1})^{\alpha} \cdot (p_{n1}) \cdot y_{n1} + \gamma}{(1-p_{n1})^{\alpha}(p_{n1}) + y_{n1} + \gamma} \quad (1)$$

Here, for the n_{th} training instance, p_{n1} is the predicted probability of positive class and y_{n1} is the ground truth label. The loss function also has two hyperparameters, alpha and gamma, which we tuned for our models.

C.2 Focal Loss

In order to focus on hard, wrongly classified samples, Focal Loss applies a modulating term to the cross-entropy loss. Given the crossentropy loss formula:

$$CrossEntropyLoss(p_t) = -\alpha_t \cdot \log(p_t) \quad (2)$$

the focal loss formula is as follows:

$$FocalLoss(p_t) = -\alpha_t \cdot (1-p_t)^{\gamma} \cdot \log(p_t) \quad (3)$$

where α and γ are the focusing hyperparameter. The higher the hyperparameter, the more the focal loss function will focus on wrongly classified samples.

C.3 F1 micro loss

We transformed the F1-micro score metric into an F1-micro loss specific to our task. This loss function optimizes the F1-micro score and prioritizes overall performance across all classes, offering a more balanced evaluation of a model's capabilities in scenarios involving class imbalance.

D Error Analysis

Class	Precision	Recall	F1
Neutral	0.53	0.39	0.45
Negative	0.77	0.71	0.74
Positive	0.74	0.85	0.79

Table 7: Classification Report of Best Performing Model

LowResource at BLP-2023 Task 2: Leveraging BanglaBert for Low Resource Sentiment Analysis of Bangla Language

Aunabil Chakma
Bangladesh University of Engineering and Technology
0419052075@grad.cse.buet.ac.bd

Masum Hasan
University of Rochester
m.hasan@rochester.edu

Abstract

This paper describes the system of the LowResource Team for Task 2 of BLP-2023, which involves conducting sentiment analysis on a dataset composed of public posts and comments from diverse social media platforms. Our primary aim is to utilize BanglaBert, a BERT model pre-trained on a large Bangla corpus, using various strategies including fine-tuning, dropping random tokens, and using several external datasets. Our final model is an ensemble of the three best BanglaBert variations. Our system has achieved overall 3rd in the Test Set among 30 participating teams with a score of 0.718. Additionally, we discuss the promising systems that didn't perform well namely task-adaptive pertaining and paraphrasing using BanglaT5. Training codes and external datasets which are used for our system are publicly available at https://github.com/Aunabil4602/bnlp-workshop-task2-2023

1 Introduction

In the field of Natural Language Processing, Sentiment Analysis has earned significant attention as a research area dedicated to the analysis of textual content. A considerable body of research on Sentiment Analysis in Bangla has been conducted. Some of these works (e.g. Islam et al. (2021), Kabir et al. (2023)) are based on introducing new datasets. In parallel, other works(e.g. Amin et al. (2019), Al-Amin et al. (2017)) are done on novel approaches. In spite of these numerous works, different opportunities still exist to improve the Analysis of Sentiments.

In this paper, we describe our system for task 2 of the Bangla Language Processing Workshop @EMNLP-2023 (Hasan et al., 2023a). We employ various systems based on BanglaBert and BanglaBert-Large (Bhattacharjee et al., 2022). Our experimental systems include fine-tuning, increasing the generalization based on dropping random tokens, using open-source external data during pre-training, and other methods described in section 4. Utilization of random token drop and external datasets has benefited our systems by improving micro-f1 scores around 0.006 to 0.01. Our best model, an ensembled model from three top models based on the Development Test-Set score, has scored a micro-f1 score of 0.7179, standing overall 3rd among 30 participants. Table 1 shows the final standings of the task.

Rank	Team	Micro-f1
1	MoFa_Aambela	0.731
2	yangst	0.727
3	LowResource(ours)	0.718
4	Hari_vm	0.717
5	PreronaTarannum	0.716

Table 1: Showing top 5 of the final standings of the BLP-2023 Task 2. Our team stands 3rd among 30 participants.

Additionally, we describe alternate potential methods that have not scored well in the result section 6. To illustrate, we explore Task Adaptive Pre-Training (Gururangan et al., 2020), in fact, has been used by this year's winner of SemEval Task 12 (Muhammad et al., 2023) on sentiment analysis of African Language, and generating paraphrases using BanglaT5 (Bhattacharjee et al., 2023). Moreover, we notice a significant drop in our score in the final test set of our best model. We describe this as our limitations in the section 7.

2 Related Works

Many of the related works are primarily focused on novel datasets covering diverse domains. Islam et al. (2022) have developed a dataset comprised of various public comments from social media platforms. Rahman and Dey (2018) have created their datasets based on Cricket and Restaurant

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 347–353

reviews. Most recently, (Kabir et al., 2023) have published a dataset entirely comprised of book reviews from online bookshops.

Existing approaches to Sentiment Analysis on Bangla Language primarily rely on machine learning and deep learning techniques. For example, Arafin Mahtab et al. (2018) have used Support Vector Machine(SVM) for Sentiment Analysis on public opinions on Cricket. Recurrent Neural Network based models are also highly used. (e.g. Hassan et al. (2016)). Irtiza Tripto and Eunus Ali (2018) have explored a variety of approaches including LSTM, SVM, and Naive Bayes. Moreover, convolutional neural network based models are also used for Sentiment Analysis on Bangla(e.g. Alam et al. (2017)).

In recent years, Large Language Models(LLM), trained on huge corpus, have become popular for their capability to understand the language and can easily fine-tuned for any task like Sentiment Analysis. LLMs based on the Bangla language(e.g. BanglaBert (Bhattacharjee et al., 2022), shahajBert (Diskin et al., 2021), BanglaT5 (Bhattacharjee et al., 2023)) are also available, which opens opportunities to work on various tasks for Bangla.

3 Task Description

This is a multi-class classification task where the objective is to detect the sentiment of the given text into 3 different classes: Positive, Negative, and Neutral. The score will be calculated using the micro-f1. The task consists of two phases: a development phase followed by a test phase. The final standing is based on the score of the test set provided during the test phase.

3.1 Dataset Description

The dataset is comprised of MUBASE (Hasan et al., 2023b) and SentiNob (Islam et al., 2021) datasets. The SentiNob dataset consists of various public comments collected from social media platforms. It covers 13 different domains, for example, politics, education, agriculture, etc. On the other hand, the MUBASE dataset consists of posts collected from Twitter and Facebook. The sample sizes of different sets given for training, validation, and testing are shown in Table 2.

Set Name	Sample size
Training	32566
Development	3934
Development Test	3426
Test	6707

Table 2: Sample sizes of various sets provided in the Task 2.

4 System Description

Here, we discuss several systems that we have experimented with for the task including the preprocessing of the dataset.

4.1 Fine-tuning Pre-trained LLMs

Fine-tuning Pre-trained Models can achieve high scores with fewer training steps. Top competitors of different shared tasks (e.g. Wang et al. (2022), Wang et al. (2023)) use these pre-trained models. For this task, we use several variations of BanglaBert for fine-tuning. Namely, we use BanglaBert and BanglaBert-Large. Besides, we also use XLM-Roberta-Large (Conneau et al., 2020), a multi-lingual model. We don't explore much on multi-lingual models, since we have found that monolingual models are more used than multi-lingual models on monolingual-specific tasks (Muhammad et al., 2023) due to high scores.

4.2 Task Adaptive Pre-training of LLMs

Gururangan et al. (2020) suggest that Domain Adapting Pre-Training(DAPT) And Task Adaptive Pre-Training(TAPT) improve the scores of the corresponding task. Here, we do TAPT on BanglaBert and BanglaBert-Large using the Electra pre-training method (Clark et al., 2020), which was originally used to pre-train these models. We don't perform DAPT since the models already cover the domains.

4.3 2-Stage Fine-Tuning of LLMs

In the first stage, we fine-tune BanglaBert using the external data only. Here, we don't include any given data from the task. In the next stage, we do regular fine-tuning on the train set. We use the term "2FT" as a short form of this approach. The list of the external datasets and sample sizes are shown in table 10.

4.4 Data augmentation

We experiment with 2 data augmentation techniques to improve the generalization. First, instead of dropping random words (Bayer et al., 2021), we drop random tokens(RTD) since dropping words

might change the meaning. We apply RTD on the fly during the training. Second, we employ paraphrasing as data augmentations using BanglaT5 (Bhattacharjee et al., 2023).

4.5 Preprocessing of Data

We remove the duplicates found in the training set and development set. We replace any url and username with URL and USER tag respectively similar to Nguyen et al. (2020). While using BanglaBert we normalize the sentence by their specific normalizer[1] as required by their model. All of the sentences are tokenized by the individual tokenizer required by each model. We set the max length of tokenization to 128 for each text.

We use several external data. However, most of the labels don't match the labels of this task. For the initial fine-tuning of the LLMs, we first map different labels to the three labels for this task. The label mapping is shown in table 11. For TAPT, we didn't need any of these labels since we do masked language modeling. Finally, we also remove the duplicates found in the external datasets.

5 Experimental Setup

We have used Models and Trainer from Huggingface[2](PyTorch version). We employ mixed precision training (Micikevicius et al., 2017) that enables faster training and consumes low GPU memory. Moreover, we built a code such that the results are reproducible. All of the experiments are done using a single V100 GPU in Google Colaboratory[3]. We do hyper-parameters search on learning rate, batch size, dropout ratios, and total epochs. We start the search with the parameter settings as suggested Gururangan et al. (2020). Our best training parameters of fine-tuning and TAPT are shown in the table 8 and 9 respectively. Note that, we don't use samples from the development set, development-test set, and test set for fine-tuning and pre-training.

6 Results

To begin, we discuss the systems that have scored well on the Development-Test's score. The top individual model is BanglaBert-Large with a random token drop that has scored 0.733, and even without any enhancement, it can score 0.723. The next best single model is BanglaBert with random token drop(RTD) and 2-stage fine-tuning that has scored 0.729. Table 3 shows the scores of our selected models in the Development-Test Set. Here, we see that both usages of external datasets and RTD have benefited the BanglaBert and BanglaBert-Large. We have built an ensemble of 3 best individual models(model ID 3, 5, and 6) that has scored 0.734, where we decide the class based on majority voting, and in case of a tie, we use the class predicted by the best model. We chose only the 3 best models for the ensemble because the other model's score was low and taking an odd number of models helps to decide the output class in case of a tie.

We have submitted the ensembled model as our best model in the test phase and has scored 0.718. Moreover, We have submitted the 3 individual best models. Our scores on the Test Set are shown in table 4. Here, we have found some inconsistency: BanglaBert-Large with random token drop, which we have considered the best model based on the Development-Test set, performed worst among the other 2 models, and BanglaBert with random token drop and pre-fine-tuned with external data, our 2nd best model, has performed the best. More importantly, every variant of BanglaBert-Large has scored low on the Test set. We discuss some analysis more in section 7. Finally, table 6 shows the confusion matrix of our ensembled model on Test set. We see that our model performed worst on detecting the Neutral class, i.e. only 412 out of 1277 samples have been correct having an accuracy of 32%, where the accuracy of Positive and Neutral classes are 78% and 83% respectively.

There are some systems that didn't achieve favorable performance from the beginning of our experiments. Firstly, TAPT didn't improve our results but rather declined the score by 0.039 with respect to simple fine-tuning as shown in table 5. What we can infer is that TAPT is supposed to help adapt the BanglaBert to the task domain, but it overfitted on the training samples, where the original model is already in a good optima that covered the task domain better.

Paraphrasing to create additional data using BanglaT5 also didn't work well. Its score is shown in table 5. The most perceptible reason is that paraphrased sentences, although good, were not diverse enough from the original sentences. Examples of generated paraphrases are shown in figure

[1]https://github.com/csebuetnlp/normalizer
[2]https://huggingface.co/
[3]https://colab.research.google.com/

1.

Other than BanglaBert, we try the XLM-Roberta-Large, a multi-lingual model, which is used by several task winners (e.g. (Wang et al., 2022)). However, it has scored low on the Development-Test set even with all enhancements. Its score is also shown in Table 3.

ID	System	Micro-F1
1	BBert	0.718
2	BBert+RTD	0.722
3	BBert+RTD+2FT	0.729
4	BBertL	0.723
5	BBertL+RTD	0.733
6	BBertL+RTD+2FT	0.725
7	XLM-Roberta-Large+RTD	0.713
8	Ensemble(3+5+6)	0.734

Table 3: Scores of our best models of various systems in the Development-Test Set. Here, BBert means BanglaBert, RTD means random token drop and 2FT means 2-stage fine-tuning.

System	Micro-F1
BBertL+RTD	0.711
BBert+RTD+2FT	0.719
BBertL+RTD+2FT	0.713
Ensemble(above 3 models)	0.718

Table 4: Scores of 3 of our best models(based on Development-Test Set) and ensembled model in Test Set. Here, BBert means BanglaBert, RTD means random token drop and 2FT means 2-stage fine-tuning.

7 Limitations and future work

As mentioned earlier, we find inconsistency in the score of our best model (BanglaBert-Large) between the Development-Test Set and the Test Set. Clark et al. (2020) have stated that variance in performance is observed with different seeds when the size of the dataset is small. We assume this might be the cause, although we didn't rely on

System	Micro-F1
Fine-Tuning	0.727
TAPT	0.688
Paraphrasing	0.674

Table 5: Performance of TAPT and Paraphrasing on BanglaBert-Large in comparison with fine-tuning on Development Set.

		Predicted		
		Neg	Neut	Pos
True	Neg	2770	244	324
	Neut	598	412	267
	Pos	331	128	1633

Table 6: Confusion Matrix of the Ensembled model on Test Set.

Seed	BBert	BBertL
1234	0.7156	0.7115
42	0.7179	0.7110
747	0.7197	0.7210
52467	0.7192	0.7122
2779	0.7135	0.7161
362	0.7185	0.7134
8194	0.7182	0.7127
avg.	0.7177	0.7140

Table 7: Scores from using different seeds for BanglaBert(BBert), BanglaBert-Large(BBertL) on Test Set.

other seeds since the distribution of Development-Test Set and Test Set should be similar as they come from the same datasets. To be more certain, we ran an experiment using different seeds for both BanglaBert and BanglaBert-Large on the Test set. As anticipated, models show varying performance when initialized with different seeds. Table 7 shows the results of this experiment. Moreover, we have found that the average score of the BanglaBert is better than the BanglaBert-Large. In fact, this result is consistent with the result found by the authors of BanglaBert that BanglaBert-Large performs lower than BanglaBert on Sentiment Analysis on SentiNob dataset[4]. BangalThus, before considering a model, the average score from different seeds needs to be evaluated when the training data is small.

TAPT is a popular method for pre-training, but it has been ineffective for our task. However, we have inferred this based on a few experiments. Thus, we suggest that more research needs to be done on the effectiveness of TAPT, as well as DAPT, on BanglaBert.

Our research has been mostly based on fine-tuning. As future work, we would like to explore using common data augmentation techniques (Bayer et al., 2021) for the given data. Besides, there are several multilingual Pre-trained

[4]https://huggingface.co/csebuetnlp/banglabert_large

Models that include the Bangla Language are need to be explored along with sophisticated methods and may even achieve better results.

8 Conclusion

In this paper, we stated our systems based on BanglaBert and BanglaBert-Large for that Sentiment Analysis task. We used simple techniques like, 2-stage fine-tuning, using external datasets, and dropping random tokens. Our system scored 3rd overall in the task. We also discussed some potential systems that didn't demonstrate satisfactory performance. More importantly, we have discussed the score inconsistency of our best model between Development-Test Set and Test Set as our limitation. Finally, we discussed directing some future research like applying TAPT and DAPT on BanglaBert and trying more data augmentations or sophisticated methods.

References

Md. Al-Amin, Md. Saiful Islam, and Shapan Das Uzzal. 2017. Sentiment analysis of bengali comments with word2vec and sentiment information of words. In *International Conference on Electrical, Computer and Communication Engineering (ECCE)*.

Md. Habibul Alam, Md-Mizanur Rahoman, and Md. Abul Kalam Azad. 2017. Sentiment analysis for bangla sentences using convolutional neural network. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–6.

Al Amin, Imran Hossain, Aysha Akther, and Kazi Masudul Alam. 2019. Bengali vader: A sentiment analysis approach using modified vader. In *International Conference on Electrical, Computer and Communication Engineering (ECCE)*.

Shamsul Arafin Mahtab, Nazmul Islam, and Md Mahfuzur Rahaman. 2018. Sentiment analysis on bangladesh cricket with support vector machine. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4.

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification. *CoRR*, abs/2107.03158.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, and Rifat Shahriyar. 2023. BanglaNLG and BanglaT5: Benchmarks and resources for evaluating low-resource natural language generation in Bangla. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 726–735, Dubrovnik, Croatia. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Avishek Das, Omar Sharif, Mohammed Moshiul Hoque, and Iqbal H. Sarker. 2021. Emotion classification in a resource constrained language using transformer-based approach. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop:150–158.

Michael Diskin, Alexey Bukhtiyarov, Max Ryabinin, Lucile Saulnier, Quentin Lhoest, Anton Sinitsin, Dmitry Popov, Dmitriy Pyrkin, Maxim Kashirin, Alexander Borzunov, Albert Villanova del Moral, Denis Mazur, Ilia Kobelev, Yacine Jernite, Thomas Wolf, and Gennady Pekhimenko. 2021. Distributed deep learning in open collaborations. In *Advances in Neural Information Processing Systems*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Md. Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023a. Blp-2023 task 2: Sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023b. Zero- and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis.

Asif Hassan, Mohammad Rashedul Amin, Abul Kalam Al Azad, and Nabeel Mohammed. 2016.

Sentiment analysis on bangla and romanized bangla text using deep recurrent models. In *2016 International Workshop on Computational Intelligence (IWCI)*, pages 51–56.

MD Iqbal, Avishek Das, Omar Sharif, Mohammed Moshiul Hoque, and Iqbal H Sarker. 2022. Bemoc: A corpus for identifying emotion in bengali texts. *SN Computer Science*, 3(2):1–17.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from bangla youtube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6.

Khondoker Ittehadul Islam, Md. Saiful Islam, and Md Ruhul Amin. 2020. Emonoba: A dataset for analyzing fine-grained emotions on noisy bangla texts. 23rd International Conference on Computer and Information Technology (ICCIT).

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Khondoker Ittehadul Islam, Tanvir Yuvraz, Md Saiful Islam, and Enamul Hassan. 2022. Emonoba: A dataset for analyzing fine-grained emotions on noisy bangla texts. Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers):128–134.

Mohsinul Kabir, Obayed Bin Mahfuz, Syed Rifat Raiyan, Hasan Mahmud, and Md Kamrul Hasan. 2023. Banglabook: A large-scale bangla dataset for sentiment analysis from book reviews. Findings of the Association for Computational Linguistics: ACL 2023:1237–1247.

Mahfuz Ahmed Masum, Sheikh Junayed Ahmed, Ayesha Tasnim, and Md Saiful Islam. 2020. An aspect-based sentiment analysis dataset for bengali and its baseline evaluation. In Proceedings of International Joint Conference on Advances in Computational Intelligence (IJCACI).

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2017. Mixed precision training. *CoRR*, abs/1710.03740.

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Seid Muhie Yimam, David Ifeoluwa Adelani, Ibrahim Said Ahmad, Nedjma Ousidhoum, Abinew Ali Ayele, Saif Mohammad, Meriem Beloucif, and Sebastian Ruder. 2023. SemEval-2023 task 12: Sentiment analysis for African languages (AfriSenti-SemEval). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2319–2337, Toronto, Canada. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Md Atikur Rahman and Emon Kumar Dey. 2018. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3.

Salim Sazzed. 2020. Cross-lingual sentiment classification in low-resource bengali language. Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020):50–60.

Salim Sazzed. 2021. Abusive content detection in transliterated bengali-english social media corpus. Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching:125–130.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schütze. 2023. NLNDE at SemEval-2023 task 12: Adaptive pretraining and source language selection for low-resource multilingual sentiment analysis. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 488–497, Toronto, Canada. Association for Computational Linguistics.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yueting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. 2022. DAMO-NLP at SemEval-2022 task 11: A knowledge-based system for multilingual named entity recognition. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1457–1468, Seattle, United States. Association for Computational Linguistics.

A Appendix

Here, we show a figure and additional tables related to our descriptions.

Parameter	BBert(+L)	XLMR
Learning Rate(LR)	2e-5	-
LR Scheduler	Linear	-
Warmup Ratio	0.0	-
Train Batch Size	16	32
Train Epochs	3	5
Weight Decay	0.01	-
Token Drop Ratio	0.2	-
Classifier Dropout	0.1	-
Max Length	128	-

Table 8: Best hyper-parameter settings for fine-tuning. BBert(+L) means both the BanglaBert and BanglaBert-Large models, XLMR means the XLM-Roberta-Large model, and "-" means equal to the left column values.

Parameter	BBertL
λ	50
MLM probability	0.25
Learning Rate(LR)	1e-4
LR Scheduler	Linear
Warmup Ratio	0.06
Train Batch Size	64
Train Epochs	100
Weight Decay	0.01
Token Drop Ratio	0.2
Max Length	128

Table 9: Best hyper-parameter settings for Electra Pre-Training of BanglaBert and BanglaBert-Large during Task Adaptive Pre-training(TAPT). λ is the weight of the loss for Discriminator

Dataset	Samples	Total class
Islam et al. (2022)	22739	6
Sazzed (2021)	1000	2
Iqbal et al. (2022)	7000	6
Kabir et al. (2023)	158065	3
Sazzed (2020)	11807	2
Rahman and Dey (2018)	2979	3
Das et al. (2021)	4994	6
Masum et al. (2020)	9014	3
Islam et al. (2020)	14852	3

Table 10: The list of External Datasets used for our training.

Original	Converted
Love	Positive
Joy	Positive
Anger	Negative
Sad/Sadness	Negative
Fear	Negative
Disgust	Negative
Surprise	Neutral
Abusive	Negative
Non-Abusive	Positive

Table 11: Label conversions of external datasets for aligning to our task.

Example-1
Original: কফি এক্সপ্রেস , মিরপুর 11 নং বাস স্ট্যান্ড থেকে পশ্চিমে । ইয়ান তাই রেস্টুরেন্ট এর পেছনে
Paraphrased: কফি এক্সপ্রেস মিরপুর ১১ বাস স্ট্যান্ড থেকে পশ্চিমে, ইয়ান রেস্তোরাঁর পিছনে।

Example-2
Original: শুধু বেশি নম্বর পেয়ে কী হবে ? কেন এ প্রশ্ন শিক্ষামন্ত্রীর ?
Paraphased: শুধু বেশি নম্বর পেলে কি হবে? শিক্ষামন্ত্রীর প্রশ্ন কেন?

Figure 1: Showing 2 paraphrasing examples using BanglaT5.

BLP-2023 Task 2: Sentiment Analysis

Md. Arid Hasan[1], **Firoj Alam**[2], **Anika Anjum**[3], **Shudipta Das**[3], **Afiyat Anjum**[3]
[1]SE+AI Research Lab, University of New Brunswick, Fredericton, Canada
[2]Qatar Computing Research Institute, Doha, Qatar
[3]Daffodil International University, Dhaka, Bangladesh
arid.hasan@unb.ca, fialam@hbku.edu.qa

Abstract

We present an overview of the BLP Sentiment Shared Task, organized as part of the inaugural BLP 2023 workshop, co-located with EMNLP 2023. The task is defined as the detection of sentiment in a given piece of social media text. This task attracted interest from 71 participants, among whom 29 and 30 teams submitted systems during the development and evaluation phases, respectively. In total, participants submitted 597 runs. However, a total of 15 teams submitted system description papers. The range of approaches in the submitted systems spans from classical machine learning models, fine-tuning pre-trained models, to leveraging Large Language Model (LLMs) in zero- and few-shot settings. In this paper, we provide a detailed account of the task setup, including dataset development and evaluation setup. Additionally, we provide a brief overview of the systems submitted by the participants. All datasets and evaluation scripts from the shared task have been made publicly available for the research community, to foster further research in this domain.[1]

1 Introduction

Sentiment analysis has emerged as a significant sub-field in Natural Language Processing (NLP), with a wide array of applications encompassing social media monitoring, brand reputation management, market research, customer feedback analysis, among others. The advancement of sentiment analysis systems has been driven by substantial research efforts, addressing its indispensable utility across diverse fields such as business, finance, politics, education, and services (Cui et al., 2023). Traditionally, analysis has been conducted across various types of content and domains including news articles, blog posts, customer reviews, and social media posts, and extended over different modalities like textual and multimodal analyses (Hussein, 2018; Dashtipour et al., 2016).

At its core, the task of sentiment analysis is defined as the extraction and identification of polarities (e.g., positive, neutral, and negative) expressed within texts. However, its scope has broadened to encompass the identification of: *(i)* the target (i.e., an entity) or aspect of the entity on which sentiment is expressed, *(ii)* the opinion holder, and *(iii)* the time at which it is expressed (Liu, 2020). Such advancements have primarily been made for high-resource languages.

Research on fundamental sentiment analysis remains an ongoing exploration, especially for many low-resource languages, primarily due to the scarcity of datasets and consolidated community effort. Although there has been a recent surge in interest (Batanović et al., 2016; Nabil et al., 2015; Muhammad et al., 2023), the field continues to pose significant challenges. Similar to other low-resource languages, the challenges for sentiment analysis in Bangla have been reported in recent studies (Alam et al., 2021a; Islam et al., 2021, 2023). Alam et al. (2021a) emphasized the primary challenges associated with Bangla sentiment analysis, specifically issues of duplicate instances in the data, inadequate reporting of annotation agreement, and generalization. These challenges were also highlighted in (Islam et al., 2021), further emphasizing the need to address them for effective sentiment analysis in Bangla.

To advance research in Bangla sentiment analysis, we emphasized community engagement and organized a shared task at BLP 2023. Similar efforts have primarily been conducted for other languages as part of the SemEval Workshop. The analysis of sentiment in tweets serves as an example of such efforts, particularly focusing on Arabic and English (Rosenthal et al., 2017). An earlier attempt at such an endeavor for Bangla is reported in (Patra et al., 2015), which mainly focused on

[1]https://github.com/blp-workshop/blp_task2

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 354–364

tweets. Our initiative significantly different from theirs in terms of datasets (e.g., data from multiple social media platforms and diverse domains) and evaluation setup.

A total of 71 teams registered for the task, out of which 30 made an official submission on the test set, and 15 of the participating teams submitted a system description paper.

The remainder of the paper is structured as follows: Section 2 provides an overview of the relevant literature. Section 3 discusses the task and dataset. Section 4 describes the organization of the task and the evaluation measures. An overview of the participating systems is provided in Section 5. Lastly, Section 6 concludes the paper.

2 Related Work

The current state-of-the-art research for Bangla sentiment classification mainly dominated focuses on two key aspects: the development or datasets and model development. Notable recent work in this direction include (Chowdhury and Chowdhury, 2014; Alam et al., 2021a; Islam et al., 2021; Kabir et al., 2023; Islam et al., 2023). Kabir et al. (2023) curated the largest dataset from book reviews, with annotations based on the review ratings. Although the dataset encompasses a large number of reviews, the class distribution poses a challenge for the Negative and Neutral classes. A well-balanced dataset has been explored in (Islam et al., 2021), comprising ∼15K manually annotated comments spanning 13 different domains. This dataset is also used as a part of this shared task.

From a modeling perspective, the existing literature addresses the problem using both classical machine learning and deep learning algorithms. These include Naive Bayes, Support Vector Machine, Decision Tree, Maximum Entropy, and Random Forest (Rahman and Hossen, 2019; Banik and Rahman, 2018; Chowdhury et al., 2019; Islam et al., 2016). Moreover, recent studies have extensively employed deep learning models for Bangla sentiment classification (Hassan et al., 2016; Aziz Sharfuddin et al., 2018; Tripto and Ali, 2018; Ashik et al., 2019; Karim et al., 2020; Sazzed, 2021; Sharmin and Chakma, 2021). Common deep learning approaches incorporate LSTMs, CNNs, attention mechanisms, and multichannel convolutional LSTMs. In the studies by Hasan et al. (2020); Alam et al. (2021a), comprehensive comparisons across various datasets were conducted, illustrating that the deep learning-based pretrained language model XLM-RoBERTa excels in performance. Comparisons between classical and deep learning-based approaches have also been explored (Ashik et al., 2019; Hasan et al., 2020; Alam et al., 2021a).

Given the significant capabilities that Large Language Models (LLMs) have demonstrated across diverse applications and scenarios, Hasan et al. (2023) explored various LLMs such as Flan-T5 (large and XL) (Chung et al., 2022), Bloomz (1.7B, 3B, 7.1B, 176B-8bit) (Muennighoff et al., 2022), and GPT-4 (OpenAI, 2023), comparing the results with fine-tuned models. The resulting performance demonstrate that fine-tuned models continue to outperform zero- and few-shot prompting. However, the performance of LLMs showcases a promising direction towards the development of systems with limited datasets for new domains.

Though there is a surge of research interest and progress, utilizing such systems in real applications remains a challenge in terms of performance and generalization capability. This shared task aimed to advance research through community effort and focus on a standard evaluation setup. As a starting point, we aimed to classify sentiment into three sentiment polarities: positive, neutral, and negative. This approach can be further extended in future studies.

3 Task and Dataset

3.1 Task

The task is defined as "detect the sentiment associated within a given text". This is a multi-class classification task that involves determining whether the sentiment expressed in the text is *Positive*, *Negative*, and *Neutral*.

3.2 Dataset

We utilized the MUBASE (Hasan et al., 2023) and SentNoB (Islam et al., 2021) datasets for the task. Both datasets were annotated by multiple annotators, with the inter-annotation agreement being 0.84 for MUBASE and 0.53 for SentNoB, respectively. The SentNoB data is curated from newspapers and YouTube video comments, covering 13 different topics such as Politics, National, International, Food, Sports, Teach, etc. The MUBASE dataset consists of comments from popular news media sources such as BBC Bangla, Prothom Alo, and BD24Live, which were collected from Facebook and Twitter.

We further analyzed the distribution of sentences based on the number of words associated with each class label, as depicted in Table 1. We created various ranges of sentence length buckets to understand and define the sequence length while training the transformer-based models. It appears that more than 80% of the posts comprise twenty words or fewer, a finding consistent with the typical of social media posts, as observed in previous studies (Alam et al., 2021b). Moreover, the average number of words and sentences per data point are 15.87 and 1.03, respectively.

Split	#Words	Pos	Neu	Neg
Train	<10	5,616	3,595	6,575
	11-20	4,587	2,212	5,613
	21-30	1,263	671	1,949
	31-40	493	287	818
	41-50	260	152	377
	51+	145	218	435
Dev	<10	587	398	723
	11-20	539	244	634
	21-30	160	68	232
	31-40	67	43	90
	41-50	22	14	34
	51+	13	26	40
Dev-test	<10	601	292	783
	11-20	420	178	603
	21-30	68	55	178
	31-40	11	21	54
	41-50	6	16	29
	51+	20	38	53
Test	<10	1,111	627	1,482
	11-20	762	382	1,183
	21-30	140	121	371
	31-40	31	56	111
	41-50	16	26	71
	51+	32	65	120

Table 1: Detailed class label distribution of the shared task data splits. Pos: Positive, Neu: Neutral, Neg: Negative.

Dataset	Train	Dev	DT	Test
MUBASE	✓	✗	✓	✓
SentNoB	✓	✓	✗	✗

Table 2: Data sources utilized in various splits for the shared task. DT: Dev-Test

For the shared task, we combined the MUBASE (Hasan et al., 2023) training set with the SentNoB (Islam et al., 2021) training set, resulting in a total of 35,266 entries for the training set. The SentNoB development set was used as the shared task development set. Additionally, the MUBASE development set served as the dev-test set for the shared task, while the test set was utilized for system evaluation and participant ranking. The specifics of the data sources are outlined in Table 2, and the detailed distribution of the data split is presented in Table 3.

Class	Train	Dev	DT	Test	Total
Pos	12,364	1,388	1,126	2,092	16,970
Neu	7,135	793	600	1,277	9,805
Neg	15,767	1,753	1,700	3,338	22,558
Total	**35,266**	**3,934**	**3,426**	**6,707**	**49,333**

Table 3: Class label distribution of the shared task dataset. DT: Dev-Test, Pos: Positive, Neu: Neutral, Neg: Negative

4 Evaluation Framework

4.1 Evaluation Measures

For evaluation, we used the *Micro-F1 score* and the evaluation scripts along with data are available online[2]. As reference points, we provided both the majority and random baselines. The majority baseline always predicts the most common class in the training data and assigns this class to each instance in the test dataset. Conversely, the random baseline assigns one of the classes randomly to each instance in the test dataset.

4.2 Task Organization

For the shared task, we provided four sets of data: the training set, development set, development-test set, and test set, as outlined in Table 3. The purpose of providing the development set is for hyperparameter tuning. We provided the development test set without labels to allow participants to evaluate their systems during the system development phase. The test set was designated for the final system evaluation and ranking. We ran the shared task in two phases and hosted the submission system on the CodaLab platform.[3]

[2] https://github.com/blp-workshop/blp_task2

[3] https://codalab.lisn.upsaclay.fr/competitions/14587

Development Phase In the first phase, only the training set, development set, and development-test set were made available, with no gold labels provided for the latter. Participants competed against each other to achieve the best performance on the development test set. A live leaderboard was made available to keep track of all submissions.

Test Phase In the second phase, the test set was released without labels, and the participants were given just four days to submit their final predictions. The test set was used for evaluation and ranking. The leaderboard was set to private during the evaluation phase, and participants were allowed to submit multiple systems without seeing the scores. The last valid submission was considered for official ranking.

After the competition concluded, we released the test set with gold labels to enable participants to conduct further experiments and error analysis.

5 Results and Overview of the Systems

5.1 Results

A total of 29 and 30 teams submitted their systems during the development and evaluation phases, respectively. In Table 4, we report the results of the submitted system on dev-test and test sets. We also include the results for the majority and random baselines. The ranking on the table was determined by the results from the test set. Note that some teams participated in the development phase but did not participate in the evaluation phase, and vice versa, as indicated by the symbol ✗. Additionally, the team marked with ∗ did not submit a system description paper.

Upon comparing the results from the dev-test and test sets across different teams, it appears that the performance difference between them is very minimal. The models did not exhibit overfitting; in some cases, the performance on the test set even surpassed that on the dev-test set.

As can be seen in Table 4, almost all systems outperformed random baseline except one system, whereas 26 systems outperformed the majority baseline. The best system, Aambela (Fahim, 2023), achieved micro-F1 score of 0.73, which is an absolute improvement of 0.23. The team mainly fine-tuned BanglaBERT and multilingual BERT along with adversarial weight perturbation. The second best system, Knowdee (Liu et al., 2023), used data augmentation with psudolabeling, which are obtained from an ensemble of models. The third best system, LowResource (Chakma and Hasan, 2023), used ensemble of different fine-tuned models.

In Table 5, we report the overview of the approaches of the submitted systems. The most used models are multilingual BERT, BanglaBERT, and XLM-RoBERTa. Specifically, 9, 8, and 14 out of 15 teams utilized multilingual BERT, BanglaBERT, and XLM-RoBERTa, respectively. Ensembles of fine-tuned models provide the best systems for this task. Additionally, two teams applied few-shot learning using the mT5, BanglaBERT large, and GPT-3.5 models. However, the teams did not provide the details regarding the prompts.

5.2 Discussion

From the official ranking presented in Table 4, early every team outperformed the performance of the random baseline system. The performance difference between the top 22 teams is very small compared with the 23rd-ranked team. In Table 6, we presented the per-class performances for the top 5 teams. Although most of the teams performed better than the random baseline by a large margin, the neutral class is still the most difficult one to identify. The low performance in neutral class might be due to its skewed distribution in the dataset. Data augmentation, up-sampling the minority class, and class re-weighting are common approaches typically used to address such issues. Although some systems employed data augmentation, it seems this issue was not thoroughly considered across all teams.

5.3 Participating Systems

Below, we provide a brief description of the participating systems and their leaderboard rank.

Aambela (Fahim, 2023) (rank 1) emerged as the best-performing team in the shared task, fine-tuning pretrained models BanglaBERT (Bhattacharjee et al., 2022a) and multilingual BERT (Devlin et al., 2019) using two classification heads. Initially, the author removed URLs and HTML tags, then applied a normalizer to the preprocessed text. Adversarial weight perturbation was utilized to enhance the training's robustness, and a 5-fold cross-validation was also conducted.

Knowdee (Liu et al., 2023) (rank 2) partitioned the data set into 10 folds and generated pseudo-labels for unlabeled data using a fine-tuned ensemble of models. They employed standard data

Rank – Team	Micro-F1	
	Dev-Test	Test
1. Aambela (Fahim, 2023)	0.7303	0.7310
2. Knowdee (Liu et al., 2023)	0.7288	0.7267
3. LowResource (Chakma and Hasan, 2023)	0.7224	0.7179
4. LowResourceNLU (Veeramani et al., 2023)	0.7248	0.7172
5. Z-Index (Tarannum et al., 2023)	✗	0.7164
- ShadmanRohan*	0.7207	0.7155
6. RGB*	0.7125	0.7112
7. EmptyMind(Fatema et al., 2023)	0.7215	0.7109
8. KeAb*	0.7125	0.7094
9. Embeddings (Tonmoy, 2023)	✗	0.7088
10. RSM-NLP (Seth et al., 2023)	0.7023	0.7078
11. DeepBlueAI*	✗	0.7076
12. nlpBDpatriots (Goswami et al., 2023)	0.7192	0.7058
13. NLP_CUET*	0.6278	0.7052
14. M1437 (Rahman and Uzuner, 2023)	0.7315	0.7036
15. Semantic_Savants*	0.6961	0.7002
16. meemaw*	✗	0.6996
17. Score_IsAll_You_Need*	0.6909	0.6930
18. VishwasGPai*	0.6970	0.6824
19. UFAL-ULD (Mukherjee et al., 2023)	0.6661	0.6768
20. Semantics Squad (Dey et al., 2023)	0.7201	0.6742
21. BanglaNLP (Saha and Nanda, 2023)	0.6810	0.6702
22. VacLM*	✗	0.6584
23. trina*	✗	0.6194
- Rachana8._K*	✗	0.5962
24. lixn*	✗	0.5889
25. Baseline (Majority)	0.4962	0.4977
26. Xenon*	✗	0.4534
27. Error Point (Das et al., 2023)	✗	0.4129
28. SSCP*	0.5584	0.3390
29. Baseline (Random)	0.3389	0.3356
30. Ushoshi2023 (Khushbu et al., 2023)	✗	0.2626
– Shilpa*	0.7166	✗
– Dhiman*	0.7154	✗
– KarbonDark*	0.7154	✗
– MrinmoyMahato*	0.7107	✗
– shakib034*	0.6734	✗
– Saumajit*	0.6559	✗
– sankalok*	0.6203	✗
– DiscoDancer420*	0.5736	✗
– Devs*	0.5736	✗
– almamunsardar*	0.5642	✗

Table 4: Official ranking of the shared task on the test set. *No working note submitted. - Run submitted after the deadline. ✗ - indicates team has not submitted system in the respective phase.

preprocessing and augmentation techniques to process the data, and fine-tuned BanglaBERT (Bhattacharjee et al., 2022a), MuRIL (Khanuja et al., 2021), XLM-RoBERTa (Conneau et al., 2020), and mT5 (Xue et al., 2021), achieving the second-best performance. The team also implemented Few-shot (3-shot) learning and compared the results with those from fine-tuned models.

Team	Models																	Misc.	
	Classical	multilingual BERT	RoBERTa	XLM-RoBERTa	BanglaBERT	BanglishBERT	MuRIL	mT5	BanglaT5	Indic-BERT	BanglaGPT2	DistilBERT	LSTM	LSTM-CNN	Bangla-BERT	Few-shot	Ensemble	Preprocessing	Data Augmentation
Aambela (Fahim, 2023)		✓			✓												✓	✓	
Knowdee (Liu et al., 2023)				✓	✓		✓	✓								✓	✓	✓	✓
LowResource (Chakma and Hasan, 2023)				✓	✓				✓								✓		
LowResourceNLU (Veeramani et al., 2023)		✓			✓												✓	✓	
Z-Index (Tarannum et al., 2023)	✓	✓			✓													✓	
EmptyMind (Fatema et al., 2023)	✓				✓								✓						
Embeddings (Tonmoy, 2023)		✓			✓					✓	✓								
RSM-NLP (Seth et al., 2023)			✓		✓	✓	✓					✓			✓		✓	✓	✓
nlpBDpatriots (Goswami et al., 2023)	✓	✓		✓	✓		✓									✓			
M1437 (Rahman and Uzuner, 2023)				✓	✓														
UFAL-ULD (Mukherjee et al., 2023)		✓		✓	✓										✓			✓	✓
Semantics Squad (Dey et al., 2023)		✓		✓	✓	✓												✓	
BanglaNLP (Saha and Nanda, 2023)	✓	✓		✓	✓														
Error Point (Das et al., 2023)	✓												✓	✓				✓	✓
Ushoshi2023 (Khushbu et al., 2023)	✓	✓		✓	✓							✓	✓					✓	✓

Table 5: Overview of the approaches used in the submitted systems.

LowResource (Chakma and Hasan, 2023) (rank 3) fine-tuned both the base and large versions of BanglaBERT (Bhattacharjee et al., 2022a), employing randomly dropping tokens, and also fine-tuned XLM-RoBERTa (Conneau et al., 2020). During the development phase, they created an ensemble of three models. However, for the evaluation phase, they ensembled only two variants of BanglaBERT, with one of them being fine-tuned using external data. Additionally, they employed task-adaptive pretraining and paraphrasing techniques utilizing BanglaT5 (Bhattacharjee et al., 2022b).

LowResourceNLU (Veeramani et al., 2023) (rank 4) fine-tuned BanglaBERT base and large (Bhattacharjee et al., 2022a), with MLM and classification heads, and multilingual BERT (Devlin et al., 2019) jointly on the XNLI and shared task dataset. They also created an ensemble of all three transformer-based models and applied multi-step aggregation to capture the most confident class predicted across all models.

Z-Index (Tarannum et al., 2023) (rank 5) utilized standard preprocessing techniques to remove URLs, usernames, emojis, and hashtags from the text. Initially, they employed SVM and Random Forest classical models, and later fine-tuned both the base and large variants of BanglaBERT (Bhattacharjee et al., 2022a), as well as the multilingual BERT (Devlin et al., 2019). The model was trained using the provided training set.

EmptyMind (Fatema et al., 2023) (rank 7) initially applied classical models such as Decision Tree, Random Forest, SVM, and XGBoost, utilizing TF-IDF vectors, as well as Word2Vec vectors. Subsequently, they employed deep learning-based models including Stacked BiLSTM and BiLSTM+CNN. Furthermore, they fine-tuned different variants of BanglaBERT (Bhattacharjee et al., 2022a).

Embeddings (Tonmoy, 2023) (rank 9) fine-tuned pretrained models BanglaBERT (Bhattacharjee et al., 2022a), BanglaGPT2,[4] Indic-BERT (Kakwani et al., 2020), and multilingual BERT(Devlin et al., 2019) using cross entropy loss function. Later to reduce the computational cost, they investigated the performances across the self-adjusting dice loss, focal loss, and F1-micro loss. They also combined training, dev, and dev-test sets as training data to train and test data to evaluate the performances of the models.

RSM-NLP (Seth et al., 2023) (rank 10) submitted their runs by fine-tuning RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al., 2019), BanglaBERT,[5] BanglaBERT (Bhattacharjee et al., 2022a), BanglishBERT (Bhattacharjee et al., 2022a), and MuRIL (Khanuja et al., 2021), with the additional use of training data. They employed standard pre-

[4] https://huggingface.co/flax-community/gpt2-bengali

[5] https://github.com/sagorbrur/bangla-bert

Class	Baseline	Aambela	Knowdee	LowResource	LowResourceNLU	Z-Index
Negative	0.3996	0.7958	0.7943	0.7873	0.7877	0.7877
Neutral	0.2368	0.4998	0.4592	0.3998	0.4021	0.4250
Positive	0.3329	0.7666	0.7599	0.7567	0.7530	0.7559

Table 6: F1 scores of the baseline and top five systems for each class.

processing techniques to process the data. They also submitted ensemble techniques (i.e., weighted and majority-voted) of fine-tuned models.

nlpBDpatriots (Goswami et al., 2023) (rank 12) began with traditional approaches such as logistic regression and SVM. Later, they fine-tuned BanglaBERT (Bhattacharjee et al., 2022a), multilingual BERT (Devlin et al., 2019), MuRIL (Khanuja et al., 2021), and XLM-RoBERTa (Conneau et al., 2020), and ensemble the models using a weighted average of the confidence predicted by each model. They also employed few-shot learning using GPT-3.5 (OpenAI, 2023).

M1437 (Rahman and Uzuner, 2023) (rank 14) fine-tuned large pretrained language models BanglaBERT large (Bhattacharjee et al., 2022a) and XLM-RoBERTa large (Conneau et al., 2020) along with the base version of each model. They also used an existing dataset (Hasan et al., 2020) in addition to the provided training data. To compare among the transformers models, they also fine-tuned the multilingual BERT. During the development phase, they were the best-performing team and they ended the competition in the $14th$ position in the evaluation phase.

UFAL-ULD (Mukherjee et al., 2023) (rank 19) fine-tuned BanglaBERT (Bhattacharjee et al., 2022a), Bangla-BERT[6] multilingual BERT (Devlin et al., 2019), and XLM-RoBERTa (Conneau et al., 2020) to tackle the problem. They followed the standard preprocessing steps to process the data and upsampled the training data to achieve balance among the classes. They also employed a focal loss function to address hard-to-classify examples.

Semantics Squad (Dey et al., 2023) (rank 20) submitted runs for both the development and evaluation phases. Standard preprocessing techniques were applied, with URLs and hashtags being removed from the data, to process and fine-tune BanglaBERT (Bhattacharjee et al., 2022a), BanglishBERT (Bhattacharjee et al., 2022a), XLM-RoBERTa (Conneau et al., 2020), and multilingual BERT(Devlin et al., 2019).

BanglaNLP (Saha and Nanda, 2023) (rank 21) also fine-tuned BanglaBERT (Bhattacharjee et al., 2022a), BERT multilingual (Devlin et al., 2019), and XLM-RoBERTa (Conneau et al., 2020) pre-trained models. Additionally, they performed parameter-efficient tuning (P-tuning) on XLM-RoBERTa. They also employed traditional models such as Logistic Regression, Naive Bayes, SGD Classifier, Majority Voting, and Stacking in their approach to the task.

Error Point (Das et al., 2023) (rank 27) performed preprocessing by removing duplicate text, filtering based on text length, and eliminating punctuation, links, emojis, non-character elements, and stopwords. They also carried out data augmentation. For their analysis, they utilized classical algorithms such as Logistic Regression, Decision Tree, Random Forest, Multinomial Naive Bayes, SVM, and SGD, using n-grams to represent the input. Additionally, they employed deep learning models, namely LSTM and LSTM-CNN.

Ushoshi2023 (Khushbu et al., 2023) (rank 30) applied preprocessing by removing punctuation marks, links, emojis, hashtag signs, usernames, and non-Bangla characters. They also applied an upsampling technique to balance the dataset. Initially, they employed traditional models such as logistic regression, decision tree, random forest, multinomial naive bayes, k-nearest neighbor, SVM, and SGD for classification. Subsequently, they fine-tuned BanglaBERT (Bhattacharjee et al., 2022a), XLM-RoBERTa (Conneau et al., 2020), DistilBERT (Sanh et al., 2019), and multilingual BERT (Devlin et al., 2019). Additionally, they trained a deep learning model, LSTM, to compare the performances across different models.

6 Conclusion and Future Work

We presented an overview of the shared task 2 (sentiment analysis) at the BLP Workshop 2023. Task 2 aimed to classify the sentiment in textual

[6] https://github.com/sagorbrur/bangla-bert

content. Notable systems employed an ensemble of pretrained language models, with the language-specific BanglaBERT being the most popular. Also, some interesting approaches including P-tuning, Few-shot learning, LLMs, and different loss functions have been explored for tackling the problem. In general, numerous models, including different kinds of transformers, have been used in the current submissions for the task.

In future work, we plan to extend the task in various ways, such as aspect-based sentiment analysis and incorporating multiple modalities.

Limitations

The BLP-2023 sentiment analysis shared task primarily focuses on sentiment polarity classification (positive, negative, and neutral) at the post level. This approach limits the identification of specific sentiment aspects and other crucial elements associated with them. Future editions of the task will address this aspect. Moreover, this edition focused solely on unimodality (text-only) models, leaving multimodal models for future study.

References

Firoj Alam, Md. Arid Hasan, Tanvirul Alam, Akib Khan, Janntatul Tajrin, Naira Khan, and Shammur Absar Chowdhury. 2021a. A review of bangla natural language processing tasks and the utility of transformer models. *arXiv preprint arXiv:2107.03844*.

Firoj Alam, Umair Qazi, Muhammad Imran, and Ferda Ofli. 2021b. Humaid: Human-annotated disaster incidents data from twitter with deep learning benchmarks. In *Proceedings of the International AAAI Conference on Web and social media*, volume 15, pages 933–942.

Md Akhter-Uz-Zaman Ashik, Shahriar Shovon, and Summit Haque. 2019. Data set for sentiment analysis on bengali news comments and its baseline evaluation. In *Proc. of ICBSLP*, pages 1–5. IEEE.

Abdullah Aziz Sharfuddin, Md. Nafis Tihami, and Md. Saiful Islam. 2018. A deep recurrent neural network with bilstm model for sentiment classification. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4. IEEE.

Nayan Banik and Md Hasan Hafizur Rahman. 2018. Evaluation of naïve bayes and support vector machines on Bangla textual movie reviews. In *Proc. of ICBSLP*, pages 1–6. IEEE.

Vuk Batanović, Boško Nikolić, and Milan Milosavljević. 2016. Reliable baselines for sentiment analysis in resource-limited languages: The serbian movie review dataset. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2688–2696.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022a. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, and Rifat Shahriyar. 2022b. Banglanlg: Benchmarks and resources for evaluating low-resource natural language generation in bangla. *CoRR*, abs/2205.11081.

Aunabil Chakma and Masum Hasan. 2023. Low-resource at BLP-2023 Task 2: Leveraging banglabert for low resource sentiment analysis of bangla language. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Rumman Rashid Chowdhury, Mohammad Shahadat Hossain, Sazzad Hossain, and Karl Andersson. 2019. Analyzing sentiment of movie reviews in Bangla by applying machine learning techniques. In *Proc. of (ICBSLP)*, pages 1–6. IEEE.

Shaika Chowdhury and Wasifa Chowdhury. 2014. Performing sentiment analysis in Bangla microblog posts. In *2014 International Conference on Informatics, Electronics Vision (ICIEV)*, pages 1–6.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 8440–8451, Online. Association for Computational Linguistics.

Jingfeng Cui, Zhaoxia Wang, Seng-Beng Ho, and Erik Cambria. 2023. Survey on sentiment analysis: evolution of research methods and topics. *Artificial Intelligence Review*, pages 1–42.

Rajesh Kumar Das, Kabid Yeiad, Jannatul Maowa, Moshfiqur Rahman Ajmain, and Mirajul Islam. 2023. Team error point at BLP-2023 Task 2: A comparative exploration of hybrid deep learning and machine learning approach for advanced sentiment analysis

techniques. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Kia Dashtipour, Soujanya Poria, Amir Hussain, Erik Cambria, Ahmad YA Hawalah, Alexander Gelbukh, and Qiang Zhou. 2016. Multilingual sentiment analysis: state of the art and independent comparison of techniques. *Cognitive computation*, 8:757–771.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Krishno Dey, Md. Arid Hasan, Prerona Tarannum, and Francis Palma. 2023. Semantics squad at BLP-2023 Task 2: Sentiment analysis of bengali text with fine tuned transformer based models. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Md Fahim. 2023. Aambela at BLP-2023 Task 2: Enhancing banglabert performance for bangla sentiment analysis task with in task pretraining and adversarial weight perturbation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Karnis Fatema, Udoy Das, Md Ayon Mia, Md Sajidul Mowla, Mahshar Yahan, MD Fayez Ullah, Arpita Sarker, and Hasan Murad. 2023. Emptymind at BLP-2023 Task 2: Sentiment analysis of bangla social media posts using transformer-based models. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Dhiman Goswami, Md Nishat Raihan, and Sadiya Sayara Chowdhury Puspo. 2023. nlpbdpatriots at BLP-2023 Task 2: A transfer learning approach towards bangla sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Md Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2023. Zero-and few-shot prompting with llms: A comparative study with fine-tuned models for bangla sentiment analysis. *arXiv preprint arXiv:2308.10783*.

Md Arid Hasan, Jannatul Tajrin, Shammur Absar Chowdhury, and Firoj Alam. 2020. Sentiment classification in bangla textual content: a comparative study. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Asif Hassan, Mohammad Rashedul Amin, Abul Kalam Al Azad, and Nabeel Mohammed. 2016. Sentiment analysis on Bangla and romanized Bangla text using deep recurrent models. In *2016 International Workshop on Computational Intelligence (IWCI)*, pages 51–56. IEEE.

Doaa Mohey El-Din Mohamed Hussein. 2018. A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*, 30(4):330–338.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271.

Md Ekramul Islam, Labib Chowdhury, Faisal Ahamed Khan, Shazzad Hossain, Sourave Hossain, Mohammad Mamun Or Rashid, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023. SentiGOLD: A large bangla gold standard multi-domain sentiment analysis dataset and its evaluation. *arXiv preprint arXiv:2306.06147*.

Md Saiful Islam, Md Ashiqul Islam, Md Afjal Hossain, and Jagoth Jyoti Dey. 2016. Supervised approach of sentimentality extraction from Bengali facebook status. In *2016 19th International Conference on Computer and Information Technology (ICCIT)*, pages 383–387. IEEE.

Mohsinul Kabir, Obayed Bin Mahfuz, Syed Rifat Raiyan, Hasan Mahmud, and Md Kamrul Hasan. 2023. BanglaBook: A large-scale Bangla dataset for sentiment analysis from book reviews. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1237–1247, Toronto, Canada. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Md. Rezaul Karim, Bharathi Raja Chakravarthi, John P. McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced Bengali language based on multichannel convolutional-lstm network. *CoRR*, abs / 2004.07807.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Sharun Akter Khushbu, Nasheen Nur, Mohiuddin Ahmed, and Nashtarin Nur. 2023. Ushoshi2023 at

BLP-2023 Task 2: A comparison of traditional to advanced linguistic models to analyze sentiment in bangla texts. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Bing Liu. 2020. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge university press.

Xiaoyi Liu, Teng Mao, Shuangtao Yang, and Bo Fu. 2023. Knowdee at BLP-2023 Task 2: Improving bangla sentiment analysis using ensembled models with pseudo-labeling. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, Nedjma Ousidhoum, David Ifeoluwa Adelani, Seid Muhie Yimam, Ibrahim Sa'id Ahmad, Meriem Beloucif, Saif Mohammad, Sebastian Ruder, et al. 2023. Afrisenti: A twitter sentiment analysis benchmark for african languages. *arXiv preprint arXiv:2302.08956*.

Sourabrata Mukherjee, Atul Kr. Ojha, and Ondřej Dušek. 2023. Ufal-uld at blp-2023 task 2 sentiment classification in bangla text. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2515–2519.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In *Mining Intelligence and Knowledge Exploration: Third International Conference, MIKE 2015, Hyderabad, India, December 9-11, 2015, Proceedings 3*, pages 650–655. Springer.

A. Rahman and M. S. Hossen. 2019. Sentiment analysis on movie review data using machine learning approach. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4.

Majidur Rahman and Özlem Uzuner. 2023. M1437 at BLP-2023 Task 2: Harnessing bangla text for sentiment analysis: A transformer-based approach. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Saumajit Saha and Albert Nanda. 2023. Banglanlp at BLP-2023 Task 2: Benchmarking different transformer models for sentiment analysis of bangla social media posts. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Salim Sazzed. 2021. Improving sentiment classification in low-resource bengali language utilizing cross-lingual self-supervised learning. In *International Conference on Applications of Natural Language to Information Systems*, pages 218–230. Springer.

Pratinav Seth, Rashi Goel, Komal Mathur, and Swetha Vemulapalli. 2023. Rsm-nlp at BLP-2023 Task 2: Bangla sentiment analysis using weighted and majority voted fine-tuned transformers. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Sadia Sharmin and Danial Chakma. 2021. Attention-based convolutional neural network for bangla sentiment analysis. *Ai & Society*, 36(1):381–396.

Prerona Tarannum, Md. Arid Hasan, Krishno Dey, and Sheak Rashed Haider Noori. 2023. Z-index at BLP-2023 Task 2: A comparative study on sentiment analysis. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

S.M Towhidul Islam Tonmoy. 2023. Embeddings at BLP-2023 Task 2: Optimizing fine-tuned transformers with cost-sensitive learning for multiclass sentiment analysis. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Nafis Irtiza Tripto and Mohammed Eunus Ali. 2018. Detecting multilabel sentiment and emotions from Bangla youtube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6. IEEE.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023. Lowresourcenlu at blp: Enhancing sentiment classification and violence incitement detection through aggregated language models. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

BLP-2023 Task 1: Violence Inciting Text Detection (VITD)

Sourav Saha † ♣, **Jahedul Alam Junaed** † ♣, **Maryam Saleki** ♠, **Mohamed Rahouti** ♠, **Nabeel Mohammad** ♢, **Ruhul Amin** ♠
♣ Shahjalal University of Science and Technology, Bangladesh,
♢ North South University, Bangladesh, ♠Fordham University, USA
{sourav95,jahedul25}@student.sust.edu, *
{msaleki, mrahouti, mamin17}@fordham.edu,
nabeel.mohammed@northsouth.edu

Abstract

We present the comprehensive technical description of the outcome of the BLP shared task on Violence Inciting Text Detection (VITD). In recent years, social media has become a tool for groups of various religions and backgrounds to spread hatred, leading to physical violence with devastating consequences. To address this challenge, the VITD shared task was initiated, aiming to classify the level of violence incitement in various texts. The competition garnered significant interest with a total of 27 teams consisting of 88 participants successfully submitting their systems to the CodaLab leaderboard. During the post-workshop phase, we received 16 system papers on VITD from those participants. In this paper, we intend to discuss the VITD baseline performance, error analysis of the submitted models, and provide a comprehensive summary of the computational techniques applied by the participating teams.

Warning: The paper examples and the corresponding dataset contain violent inciting, derogatory, abusive, and racist comments. .

1 Introduction

Social media's growth over the past decade has reshaped the distribution of information to the broader public (Ferguson et al., 2014). However, it has also surfaced as a potential breeding ground for provoking violence among different groups, from religious to ethnic to gender-based distinctions. In fact, many of the violent incidents of the recent past era can directly or indirectly be attributed to incitement from social media (Mengü and Mengü, 2015). Such platforms can act as catalysts for the incitement of violence and the radicalization of individuals or groups (Recuero, 2015). Extremist ideologies and hate speech can spread rapidly, leading to real-world acts of violence. Acts of violence, triggered or fueled by content shared on social media, can inflict physical harm to individuals and communities with dire consequences that include physical injuries, destruction of properties, and even loss of human lives.

In the recent past, numerous studies were conducted into areas like hate speech detection (Warner and Hirschberg, 2012; Waseem and Hovy, 2016; Davidson et al., 2017; Karim et al., 2020; Romim et al., 2021), abusive content identification (Nobata et al., 2016), and misinformation detection (Shu et al., 2017; Hossain et al., 2020), aiming to understand and prevent harmful social media activities. There have been several workshops that contributed datasets and organized shared tasks on online harmful content detection in different languages (Bosco et al., 2018; Fersini et al., 2018; Zampieri et al., 2019; Basile et al., 2019). However, to the best of our knowledge, there exists no research work on the violence incitement in the Bengal Region (Bangladesh and West Bengal in India), the residence of more than 272 million[1,2] people of many diverse background. Therefore, this shared task seeks to bridge this gap by contributing a novel dataset on VITD for the development of new systems and methodologies with the objective to advance our collective understanding and capabilities in this crucial domain. In this paper, we discuss the following:

1. **Dataset Overview:** VITD task presents an intriguing challenge centered around the catego-

† Authors have equal contributions

[1] https://en.wikipedia.org/wiki/Bangladesh
[2] https://en.wikipedia.org/wiki/West_Bengal

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 365–375

Category	Definition	Example
Direct Violence	It refers to killing, rape, vandalism, deportation, desocialization, and resocialization.	দোকানে আগুন জ্বালিয়ে দেওয়া উচিৎ (The shop should be set on fire)
Passive Violence	It refers to use of derogatory language, abusive remarks, slang or any form of justification for violence.	সরকারের দোষ, সরকারের দালালি বন্ধ কর (Blame the government, stop the government brokering)
Non-Violence	It refers to discussions about social rights or general conversational topics that do not involve any form of violence.	সত্য প্রকাশে যমুনা টিভিকে ধন্যবাদ (Thanks to Jamuna TV for revealing the truth)

Table 1: The Table depicts examples of 3 different categories: Direct Violence (Red), Passive Violence (Yellow), & Non-Violence (Green). We also show the English translation using Google Translator service.

rization of textual content into three distinct and vital categories: Direct Violence, Passive Violence, and Non-Violence. We discuss how this dataset was prepared for the task.

2. **Baseline Performance:** We present the Macro-F1 score of VITD using both multilingual and Bangla BERT models.

3. **Team Statistics:** We discuss the participant's demographics in terms of gender and background.

4. **Error Analysis:** We present a detailed error analysis of each model submitted by the 27 teams.

5. **Comprehensive System Summary:** We also discuss the computational techniques used by different teams for the shared task.

2 Dataset Overview

The Vio-Lens dataset addresses the challenges of Violence Incitement Text Detection (VITD). It comprises data from YouTube comments related to violent content from Bangladesh and West Bengal. The dataset categorizes violence incitement into three classes: *Direct Violence, Passive Violence, and Non-Violence*. The description of each category along with relevant examples is provided in Table 1. The dataset features 6046 samples: 786 samples for direct violence, 2058 for passive violence, and the remaining 3202 for non-violence. This distribution illuminates a discernible class imbalance within the dataset, underscoring the need for careful consideration when designing and implementing classification algorithms or methodologies. For a detailed description of the Vio-Lens dataset, we refer the reader to the dataset paper Saha et al. (2023)[3].

[3]The dataset is publically available in https://github.com/blp-workshop/blp_task1/tree/main/dataset

3 Task Description and Evaluation

3.1 Task Definition

The shared task provides a classification task on three categories of violence, *Direct Violence, Passive Violence, and Non-Violence*, as discussed below:

- **Direct Violence:** This category encompasses explicit threats directed towards individuals or communities, including actions such as killing, rape, vandalism, deportation, desocialization (threats urging individuals or communities to abandon their religion, culture, or traditions), and resocialization (threats of forceful conversion). The detection of direct violence is crucial due to its potential to have severe consequences in the future.

- **Passive Violence:** This category includes instances characterized by the employment of derogatory language, derogative terms, or abusive remarks aimed at individuals or communities. Moreover, any attempt to rationalize or justify violence is classified within this category. Acknowledging these nuanced forms of hostility is key to understanding the breadth of online aggression.

- **Non-Violence:** Content within this category addresses non-violent matters, ranging from discussions about social rights to general conversations that are free from any violent implications. It's crucial to distinguish these benign exchanges from those that carry a more harmful intent.

3.2 Task Organization

We ran our competition on the CodaLab [4]. platform. There were two primary phases: (i) the Trail

[4]https://codalab.lisn.upsaclay.fr/competitions/14620

phase started on 16 July 2023 and ended on 15 August 2023, and (ii) the Test Phase, which began on 16 August 2023 and ended on 18 August 2023. We provided a training phase with the text and label, while the test phase contained only text data.

Models	F1 Score (Macro)
Majority Voting	23.350
MBERT	63.282
DistillBERT	59.863
XLM-RoBERTa (base)	66.062
BanglaBERT (base)	**71.073**

Table 2: The table shows the outcomes (macro-F1) classification using majority voting, MBERT, DistillBERT, XLM-RoBERTa, and BanglaBERT for the test set. All the experiments used the same dataset and parameters for a fair evaluation. We observe that BanglaBERT achieved the best macro F1 score.

3.3 Evaluation Metrics and Baselines

We evaluated all participating systems with Macro-F1 score. We are providing five baseline models (see Table 2) to benchmark a range of simple to complex systems for VITD. The simplest baseline model is the Majority Baseline, where all the categories are predicted as the majority Non-violence class. We provided four other fine-tuned Large Language models: XLM-RoBERTa (Liu et al., 2019), MBERT (Devlin et al., 2019), DistillBERT (Sanh et al., 2019), and BanglaBERT (Bhattacharjee et al., 2021). The first two are Multilingual models, while the third were monolingual ones. We ran all the models using the following parameters: learning rate 1e-5, train batch size 8, evaluation batch size 8, epochs 50, evaluation steps 250, and early stopping patience 5. Among the four baselines, the monolingual BanglaBERT provided the best Baseline with the highest macro F1 score of 78.791 on the dev set and 71.073 on the test phase.

3.4 Team Statistics

Our contest attracted 27 teams containing members from around the world. Among the contestants, 69 were male and 19 were female (Figure 1). The contest attracted participants including undergraduate students, graduate students, and professionals containing 13 undergraduates majority, 7 graduates majority, and 7 professionals majority teams.

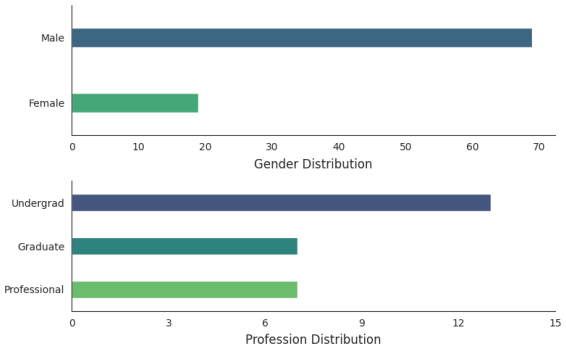


Figure 1: The figure shows gender distribution among the contestants and professions of each category of participants.

4 Participants Results

The baseline model with the best performance, BanglaBERT (Bhattacharjee et al., 2021), was outperformed by 16 teams. We display the ranking and best-performing models performance for each team in Table 3. We also report precision, recall, and F1 score for each category. Team DeepBlueAI achieved the highest overall performance, obtaining the Macro-F1 score of 76.044.

We observe that the highest precision, recall, and F1 score were reported for the **Non-Violence** category and worst on the *Direct Violence* category - indicating potential challenges in identifying explicit content. This may be due to the data imbalances in the dataset. Specifically, *Non-Violence* occupies 51.44%, 53.90%, and 54.37% of data on the train, validation, and test sets, respectively. On the other hand, *Direct Violence* is represented in only 14.41%, 14.74%, and 9.97% of the corresponding sets. In terms of team performance, a total of 20 teams surpassed the benchmark F1 score for the *Direct Violence*, and 17 teams achieved that for *Non-Violence*, while only 11 teams were found to cross the benchmark for *Passive Violence*. In particular, three teams: DeepBlueAI, Aambela, and NLP_CUET, exhibited high F1 scores across all three categories.

4.1 Error Analysis

A total of 27 teams participated in the VITD task. Among the 2,016 test samples, 506 unique samples were accurately predicted by all participating teams. There are a total of 72 samples that were incorrectly predicted by all the 27 teams. Additionally, there are a total of 214 unique samples that were incorrectly predicted by exactly one of the 27

Rank	Team	F1 score (macro)	Direct			Passive			Non-Violence		
			P	R	F1	P	R	F1	P	R	F1
1	DeepBlueAI	76.044	56.811	85.075	68.127	85.634	63.839	73.147	83.800	90.146	86.857
2	Aambela	76.041	59.286	82.587	69.023	84.404	63.978	72.785	82.872	90.055	86.314
3	NLP_CUET	74.587	61.004	78.607	68.696	73.745	71.488	72.599	83.868	81.113	82.468
4	Team Embeddings	74.418	52.761	85.572	65.275	81.122	66.342	72.992	84.755	85.219	84.986
5	Semantics Squad	74.413	57.664	78.607	66.526	81.607	63.561	71.462	82.149	88.595	85.250
6	NLP_BD_PATRIOTS	74.313	54.276	82.090	65.347	78.537	67.177	72.414	85.141	85.219	85.180
7	the_linguists	73.978	54.485	81.592	65.339	80.000	65.090	71.779	83.540	86.131	84.816
8	Panda	73.808	54.430	85.572	66.538	85.655	57.302	68.667	81.870	91.058	86.220
9	EmptyMind	73.797	52.266	86.070	65.038	82.130	63.282	71.485	83.554	86.223	84.868
10	Mavericks	73.699	55.932	82.090	66.532	82.863	61.196	70.400	80.840	87.774	84.164
11	LowResourceNLU	73.468	54.574	86.070	66.795	85.983	57.163	68.672	80.590	89.781	84.937
12	VacLM	72.656	50.286	87.562	63.884	80.536	62.726	70.524	83.183	83.942	83.560
13	LexicalMinds	72.551	51.562	82.090	63.340	83.080	60.779	70.201	81.453	86.953	84.113
14	Score_IsAll_You_Need	72.376	55.805	74.129	63.675	82.163	60.223	69.502	79.624	88.777	83.952
15	winging_it	71.207	45.316	89.055	60.067	83.622	60.362	70.113	83.212	83.668	83.439
16	Semantic_Savants	71.179	51.235	82.587	63.238	82.200	57.163	67.432	79.530	86.496	82.867
–	**Baseline**	**71.073**	**46.690**	**84.081**	**60.033**	**79.680**	**62.732**	**70.194**	**83.271**	**82.663**	**82.970**
17	BpHigh	70.978	53.741	78.607	63.838	80.639	56.189	66.230	78.624	87.591	82.866
18	SUST_Black Box	70.680	47.500	85.075	60.963	83.128	56.189	67.054	81.368	86.861	84.025
19	Team_Syrax	70.450	56.226	74.129	63.948	84.703	51.599	64.131	76.390	91.515	83.271
20	Blue	70.012	45.938	81.592	58.781	82.927	56.745	67.382	81.320	86.588	83.871
21	Team CentreBack	69.390	50.530	71.144	59.091	78.435	57.163	66.130	79.074	87.226	82.950
22	UFAL-ULD	69.009	47.447	78.607	59.176	75.215	60.779	67.231	80.399	80.839	80.619
23	BanglaNLP	68.110	53.650	73.134	61.895	78.602	51.599	62.301	74.646	86.496	80.135
24	KUET_NLP	60.332	36.557	77.114	49.600	75.204	38.387	50.829	76.327	85.310	80.569
25	Shibli_CL	38.427	37.727	41.294	39.430	68.421	01.808	03.523	58.469	94.799	72.329
26	Team Error Point	31.913	08.150	18.408	11.298	31.959	08.623	13.582	63.816	79.653	70.860
27	lixn	31.426	36.000	17.910	23.920	25.000	00.139	00.277	55.126	96.168	70.080

Table 3: The table shows the performance of each team along with the best-performing baseline model (BanglaBERT-base). It contains precision (P), recall (R), and F1 scores of individual categories, and finally a macro F1 score across all categories for final judgment.

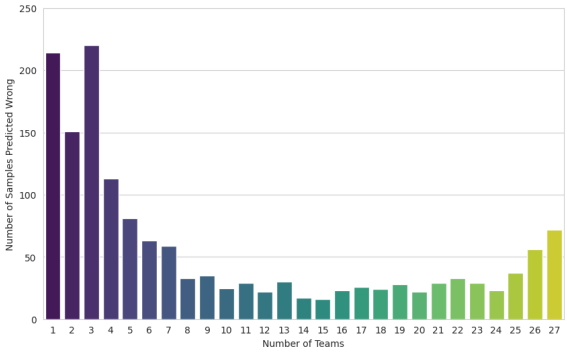


Figure 2: The bar plot shows the number of unique samples (Y-axis) that were predicted wrong by exactly N number of teams (X-axis) out of total 27 teams.

teams. A detailed visualization of these errors can be seen in Figure 2. In summary, a total of 1,510 samples were predicted incorrectly by one or more teams.

For the *Direct Violence* category, out of 201 test instances, 75.05% were predicted accurately by all the teams, while 6.80% were mistakenly identified as *Passive Violence*, and 18.15% were misclassified as *Non-Violence*. The *Passive Violence* test set comprises of 719 samples. Of those, 53.37% were correctly classified by all the teams, while 13.45% were categorized incorrectly as *Direct Violence*, and the rest samples were erroneously categorized as *Non-Violence*. For the *Non-Violence* category, which had 1,096 samples in the test set, an impressive 87.19% were correctly categorized by all the teams. Only 5.54% of those samples were incorrectly identified as *Direct Violence*, and the remaining 7.27% were misclassified as *Passive Violence* (see Figure 3).

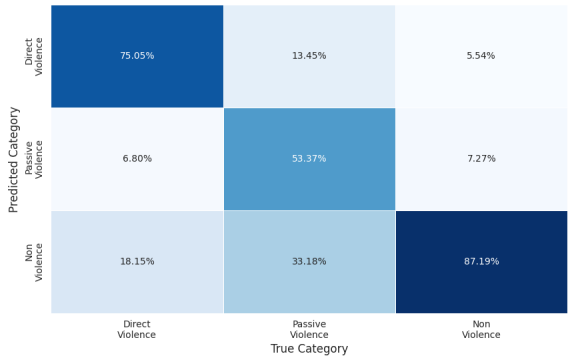


Figure 3: Confusion matrix illustrating category distribution among 27 teams.

We present a few examples from each of the categories, that were predicted wrong by all the teams (see Table 4). For the *Non-Violence* category, no

teams misclassified the same samples, indicating that the category may be easier to predict than the rest.

Example	Category
তোদের মত দাঙ্গাবাজ কুকুরদের বিচার আমি করব (I will judge riotous dogs like you)	Direct Violence
দেখা হবে ভাই মাঠে ময়দানে কোরআন নিয়ে উলটা পালটা কিছু বল্লে (See you in the field if you say something bad about Qur'an)	Direct Violence
ইসলামী আইন অনুযায়ী এই মেয়ের ঘরের মধ্যে আবদ্ধ থাকা উচিত,এ বাইরে কেনো। ইসলামে তো নারীদের যৌন দাসী হিসাবে ব্যবহার করে,এ ঘরের বাইরে গেলে তো ইসলামের অবমাননা করা হয়। (According to Islamic law, this girl should be confined inside the house, why she outside? In Islam, women are used as sex slaves, if she goes out of this house, Islam is insulted.)	Passive Violence
ধর্ম মানেই পাগলামি। সংঘাত, গালাগালি, মারামারি, খুন, ধর্ষন। (Religion means madness, conflict, abuse, fighting, murder, rape.)	Passive Violence

Table 4: This table presents some samples that all the teams predicted wrongfully. It is also to be noted that such wrong predictions were only observed either for *Direct or Passive Violence* categories.

5 Participants System Description

In this section, we present a comprehensive summary of each submitted system for the shared task.

AAmbela (Fahim, 2023) stood second in the competition with an overall Macro-F1 score of 76.040 for the test set. They propose an instruction-finetuned csebuetnlp-BanglaBERT (Bhattacharjee et al., 2022) with three classification heads. As BanglaBERT's vocabulary does not fully cover the tokens in the data, the team added them as special tokens that were learned during the training phase. They also observe the significance of emojis in the dataset, and removing them often leads to a minor result. On the other hand, converting emojis to text and normalizing the text leads to a better result. They experimented with various approaches such as traditional classifiers (SVM, Random Forest, XG-Boost) with Tf-IDF embeddings, Deep learning models (LSTM), and transformer-based architectures (mBERT-case, mDeBerta-v3 base (He et al., 2021a,b), XLM-Roberta base, SagorSarker-BanglaBERT (Sarker, 2020), BanglaBERT (Bhattacharjee et al., 2022). Finally, BanglaBERT trained on three epochs with a batch size of 16 came out on the top.

NLP_CUET (Hossain et al., 2023) achieved 3rd rank in this task with an overall Macro-F1 score of 74.587. They preprocessed data by removing unwanted characters and employed feature extraction methods like TF-IDF and Word2Vec. After investigating several machine learning, deep learning, and transformer-based models, they propose a hybrid method using GAN (Goodfellow et al., 2020) and Bangla-ELECTRA. Here, they considered both labeled data and unlabeled data for model training. The generator and discriminator are both multilayer perceptrons with a single hidden layer of 512 neurons. The generator input is a randomly generated vector of 100 dimensions, and it outputs a fake transformer embedding vector for a single token. The transformer-based model processed the input text, generating a contextualized embedding vector for the CLS token. These embedding vectors from the transformer and generator were then input into the discriminator. The output of the discriminator is extended to K+1 classes where k is the number of classes in this classification task, and the extra class is "REAL." In this approach, they focused on determining whether the embedding produced by the transformer-based architecture is real or fake. During the testing phase, they discarded the generator and used the BERT and discriminator model to classify the input data. They masked the prediction output for the 'REAL' class during testing.

Seamntic Squad (Dey et al., 2023) received the fifth rank with an overall Macro-F1 score of 74.413. They applied a preprocessing step of removing punctuation, lemmatization, and oversampling/undersampling. Afterward, they used different transformer-based models such as XLM-Roberta (base and large), BanglaBERT (Bhattacharjee et al., 2022) (base and large), and mBERT. Among the approaches, BanglaBERT-base achieved the highest result.

nlpBDpatriots (Raihan et al., 2023) received sixth in the competition with a macro f1 score of 74.313. They applied a rigorous data augmentation process, including translation and back-translation to make the dataset 7 times larger. They applied Statistical machine learning models (Linear Regression, Support Vector Machine), GPT-3.5, and various transformer-based approaches. Their two-step approach first classified violence and non-violence with MuRIL(Khanuja et al., 2021), and later XLM-RoBERTa to classify violence and non-violence on the larger dataset performed best.

the_linguists (Tariquzzaman et al., 2023) achieved 7th rank in this task with an overall

Macro-F1 score of 73.978. Firstly they collected 6.8 million data samples from Facebook and YouTube. Then they applied some preprocessing steps which resulted in a refined dataset containing 3.8 million samples. After that, they applied a semi-supervised methodology for training where the training of the informal FastText word embedding model was done by making use of the preprocessed unlabeled data. These embeddings were then integrated into the LR, SVM, LSTM, BiLSTM, and GRU models which were fine-tuned using the labeled data. And they got the best result from BiLSTM.

EmpytyMind (Das et al., 2023b) achieved 9th rank in this task with an overall Macro-F1 score of 73.797. They first preprocessed the dataset and then normalized the text. After that, they applied statistical machine learning-based approaches (Random Forest and Support Vector Machine, XG-Boost), deep learning-based approaches (one three bidirectional LSTM layers and the other four LSTM layers), and transformer-based approaches using a two-step hierarchical approach. In the hierarchical approach, they first classified the text into violence and non-violence categories, then further classified the violence category into direct violence and passive violence to combat the imbalance dataset, and it yielded the best performance.

Mavricks (Page et al., 2023) received 10th place in the competition with an overall Macro-F1 score of 73.699. They applied different transformer-based models (BanglaBERT, BanglishBERT, MuRIL, XLM-Roberta, and BengaliBERT) and ensembled them. They applied different ensembling methods among which hard voting came out on top.

LowResourceNLU (Veeramani et al., 2023) achieved 11th rank in this task with an overall Macro-F1 score of 73.468. Here, they aggregate three BERT-based language models. They configured the first model by incorporating two heads, one for Masked Language Modeling (MLM) and the other for classification, within the BanglaBERT-*large* framework. They used mBERT as their second model. As their third model, they used BanglaBERT-*base* by incorporating two classification heads. The first head focuses on the Bangla version of the XNLI dataset (Conneau et al., 2018). The second head is dedicated to the dataset. Initially, they extracted individual predictions from each model using the argmax function, selecting the class with the highest confidence score for each model. Then they applied another argmax operation, this time on the maximum logit values obtained from each model. Because of the incorporation of MLM in the first model, the F1 score is enhanced by a substantial margin. Similarly, the joint pretraining with XNLI significantly increased the performance of the third model. The combination of three models exhibits superior performance as compared to the use of a single model alone.

VacLM (Chatterjee et al., 2023) ranked 12th on the competition with an overall Macro-F1 score of 72.656. They introduced external information by incorporating data from Karim et al. (2020) and manually annotating them. They observed augmenting data from external sources in this way actually hampers the performance in the 3-way classification task but generally performs better for the violence and non-violence classification task.

Score_Is_All_You_Need (Ahmed et al., 2023) received 14th place in the competition with an overall Macro-F1 score of 72.376. They applied a two-step approach to first classify violence and Non-Violence. Afterward, from the violence category, they classify direct and passive violence using transformer-based approaches. They applied BanglaBERT, M-BERT, and XLM-RoBERTa using an exhaustive hyperparameter search to fit the model.

SUST_Black_Box (Shibu et al., 2023) ranked 18th in the competition with an overall Macro-F1 score of 70.680. They applied to incorporate data from similar sentiment and hate speech-related datasets for data augmentation. They used different transformer-based techniques such as SagorSarker-BanglaBERT(Sarker, 2020), M-BERT, and RoBERTa on the augmented dataset. Finally, they applied different ensembling methods to the augmented dataset.

Team_Syrax (Riyad et al., 2023) received 19th in the competition with an overall Macro-F1 score of 70.450. They applied traditional preprocessing steps such as emoji and punctuation removal. Then, they applied data augmentation from the Bengali hate speech detection dataset (BAD, BD-SHS). They applied different ensemble methods such as bagging and hard majority voting for the classification.

Team CentreBack (Alamgir and Haque, 2023)

ranked 21st in the competition with an overall F1 score of 69.390 in the test set. They applied several approaches using transformer-based architectures (BanglaBERT and XLM-Roberta) and a two-stage approach where they first classified violence and non-violence and then further classified the violence into direct and indirect violence. They also applied a few-shot approach with SBERT but it ultimately resulted in a poor performance. Among those approaches, BanglaBERT (20 epochs received the highest approach with the stage approach closely behind.

UFAL-ULD (Mukherjee et al., 2023) ranked 22nd in the competition with an overall Macro-F1 score of macro 69.009 for the test set. They applied different transformers-based models: XLM-Roberta-base, XLM-Roberta-large, BanglaBERT-Sagor, BanglaBERT-BUET and BanglaBERT-BUET-large. They used focal loss to handle the issue of class imbalance and applied simple data augmentation techniques like synonym replacement, insertion, deletion, swap, and shuffle.

BanglaNLP (Saha and Nanda, 2023) ranked 23rd in the competition with an overall Macro-F1 score of 68.110 for the test set. They used a general paraphrasing technique for data augmentation. In addition using general classification techniques such as logistic regression, SGD classifier, and multinomial naive bayes with ensembling techniques such as majority voting and stacking. They finally used BanglaBERT (Sarker) (Sarker, 2020) and Multilingual-E5-base as transformer-based model, with the later ultimately provided the best performance.

Team Error Point (Das et al., 2023a) ranked 26th with an overall Macro-F1 score of 31.913. They applied different traditional machine learning classifiers along with CNN and LSTM. Their combination of LSTM and CNN achieved the highest performance.

6 Discussion

6.1 Popular Architechture

The large majority of the participants (14 teams) employed transformer-based methods. They used mBERT, mDeBerta-v3 base, XLM-Roberta (*base* and *large*), SagorSarker-BanglaBERT, BanglaBERT (base and large), MuRIL, etc. Notably, variants of BanglaBERT consistently outperformed other models. Several submissions explored statistical machine learning methods leveraging FastText and Word2Vec for word-embeddings and subsequently used SVM, Logistic Regression, and XGBoost for classification. Another popular technique used by some teams is the two-steps approach to first classify the violence and non-violence and then subsequently classify them into *Direct and Passive Violence*. NLP_CUET used a GAN-based architecture. Please see Table 5 for details.

6.2 Popular Methods

Ensembling of different classifiers and transformers is the most prominent method used by the participants. Among the ensembling methods, hard voting gave the best results. Some teams used a two-step approach to classify the violence category and then the direct and passive violence from that category. Some teams tended to add more data to the dataset. They primarily adopted two approaches: One of the approaches included operations on the dataset such as insert, substitution, deletion, translation, and back-translation. The other approaches included datasets from similar datasets such as the Bangla Hate Dataset (Romim et al., 2021), and XNLI Dataset (Conneau et al., 2018), etc.

6.3 Insights

Generally, most of the successful process has been monolingual pre-trained language model modified with various task-specific process. Specially BanglaBERT (Bhattacharjee et al., 2022) has been the most impactful monolingual model. Emojis played a crucial role in the dataset build-up process and played a crucial role in the annotation. So, removing those has a negative impact on the prediction (Fahim, 2023). Also, statistical machine learning methods such as SVM, and XGBoost embedded after Fastext or Word2Vec don't capture the complex context of the dataset and fall short in the prediction. Deep Learning methods such as RNN, LSTM, and Bi-LSTM generally perform better than the statistical machine especially Das et al. (2023b) showed a significant score using a combination of lstm and bi-lstm with a two-step approach. Ultimately BanglaBERT (Bhattacharjee et al., 2022) was the most prominent for all the teams having a vast amount of pretrained knowledge of Bangla at its disposal.

Team Name	Embedding			Statistical Machine Learning							Deep Learning					Transformer Based Approach								GAN
	FastText	Word2Vec	Tf-Idf	LR	SVM	KNN	DT	RF	MNB	SGD	XGBoost	RNN	LSTM	Bi-LSTM	GRU	CNN	BanglaBERT (Sagor)	BanglaBERT (csebuetnlp)	MBERT	XLM-RoBERTa	MuRIL	Indic Bert	mDeBerta -v3 base	
AAmbela			✓		✓			✓			✓		✓				✓	✓	✓				✓	
NLP_CUET		✓	✓	✓	✓		✓	✓	✓	✓				✓		✓		✓						✓
Seamntic Squad																	✓	✓	✓	✓				
nlpBDpatriots				✓	✓														✓	✓	✓			
the_linguists	✓			✓	✓								✓	✓	✓									
EmpytyMind		✓			✓			✓			✓		✓	✓				✓						
Mavricks																		✓		✓	✓			
LowResourceNLU																		✓	✓					
VacLM																		✓			✓	✓		
Score_Is_All_You_Need																		✓	✓	✓				
SUST_Black_Box																	✓	✓	✓	✓				
Team_Syrax																		✓	✓	✓				
Team CentreBack																		✓		✓				
UFAL-ULD																	✓	✓	✓	✓				
BanglaNLP			✓	✓					✓	✓							✓		✓					
Team Error Point			✓	✓	✓		✓	✓	✓	✓			✓			✓								

Table 5: This table shows the most popular techniques used by different teams.

7 Limitations

Quantitative Limitations: The main limitation of the shared task arises from the dataset. First of all, the dataset is small in size, with only 4k data points for the training and validation sets, and around 2k data points for the test set. This often creates problems in terms of over-fitting on large models. Additionally, the dataset is highly imbalanced with only a minor fraction of the data for direct violence creating a challenge for class detection which is also reflective of the participant's results.

Qualititive Limitations: Emojis play a crucial role in sentence classification, so removing any emoji during preprocessing leads to a loss of context. The dataset consists of data from Bangladesh and West Bengal, comprising only in Bengali language. Therefore, the nature of violence-inciting text's nature may differ based on culture and language. Finally, the dataset requires a hectic process to annotate and validate thus expanding the correct data is much more difficult.

Procedural Limitations: The dataset is fully annotated by Bangladeshi residents, all undergraduate students, with an expert resolving the dispute. The annotation is done based on previous literature, personal observations, and a strict framework for annotators to rely on. Then relying on a majority vote and expert adjudication to produce is used to reach a gold standard label. Several previous studies reveal that annotator identity is a critical determinant of data annotation patterns (Sap et al., 2019; Larimore et al., 2021; Waseem, 2016) and so majority voting doesn't always capture the subjective nature of the annotation (Davani et al., 2022). Nonetheless, the definition of violence and its subcategories in taxonomy and how the authors' builders built their dataset and the annotators applied their best judgment are based on societal papers primarily from Galtung (1969, 1990), does not take FRS (Faith, Religion and Societal Impact) into account. Therefore, any dataset and corresponding systems will have the mentioned limitations. Thus, others with different cultural, societal, or religious backgrounds may disagree with some of the annotations.

8 Conclusion

In this paper, we have presented an overview of the shared task on the Violence Inciting Text Detection (VITD) dataset. The contest fostered submissions from 27 teams with 16 teams outperforming the highest baseline system BanglaBERT (Bhattacharjee et al., 2021), and 17 teams submitted the system paper. The system description and subsequent analysis and limitations discussion demonstrate the successful completion of the task.

The task has some vast scope for improvement. As mentioned in Saha et al. (2023), there is significant unlabelled data ready for further improvement of the systems to invoke larger systems without over-fitting the larger models. A potential scope for improvement is adding more data from huge unlabelled data. A future version of the task may be arranged with the challenge of more data from dif-

ferent sources, languages, and regions. Also, real-time violence detection models can be the next step of the task.

Ethical Considerations

We release the dataset and baseline classes and individual systems for specific classes containing violence-inciting texts. We also shared the participants' system descriptions. The malicious actors can use this information to train a generative model and use it for malicious purposes (Kirk et al., 2022). However, we believe that the risk is negligible to the huge potential of such systems in detecting violence-inciting text detection. The annotators were interviewed by the task organizers and they assured that they were given proper mental support and did not face any challenges at the time or after completing the annotation procedure.

References

Kawsar Ahmed, Md Osama, Md. Sirajul Islam, Md Taosiful Islam, Avishek Das, and Mohammed Moshiul Hoque. 2023. Score_isall_you_need at blp-2023 task 1: A hierarchical classification approach to detect violence inciting text using transformers. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Refaat Mohammad Alamgir and Amira Haque. 2023. Team centreback at blp-2023 task 1: Analyzing performance of different machine-learning based methods for detecting violence-inciting texts in bangla. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 54–63.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Samin, M Sohel Rahman, Anindya Iqbal, and Rifat Shahriyar. 2021. Banglabert: Combating embedding barrier for low-resource language understanding. *arXiv preprint arXiv:2101.00204*.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad Uddin, Kazi Mubasshir, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2022. Banglabert: Lagnuage model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL*.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, Tesconi Maurizio, et al. 2018. Overview of the evalita 2018 hate speech detection task. In *Ceur workshop proceedings*, volume 2263, pages 1–9. CEUR.

Shilpa Chatterjee, P J Leo Evenss, and Pramit Bhattacharyya. 2023. Vaclm at blp-2023 task 1: Leveraging bert models for violence detection in bangla. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Rajesh Kumar Das, Jannatul Maowa, Moshfiqur Rahman Ajmain, Kabid Yeiad, Mirajul Islam, and Sharun Akter Khushbu. 2023a. Team error point at blp-2023 task 1: A comprehensive approach for violence inciting text detection using deep learning and traditional machine learning algorithm. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Udoy Das, Karnis Fatema, Md Ayon Mia, Mahshar Yahan, Md Sajidul Mowla, MD Fayez Ullah, Arpita Sarker, and Hasan Murad. 2023b. Emptymind at blp-2023 task 1: A transformer-based hierarchical-bert model for bangla violence-inciting text detection. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Krishno Dey, Prerona Tarannum, Md. Arid Hasan, and Francis Palma. 2023. Semantics squad at blp-2023 task 1: Violence inciting bangla text detection with fine-tuned transformer-based models. In *Proceedings of the 1st Workshop on Bangla Language*

Processing (BLP 2023), Singapore. Association for Computational Linguistics.

Md Fahim. 2023. Aambela at blp-2023 task 1: Focus on [unk] tokens: Analyzing violence inciting bangla text with adding dataset specific new word tokens. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Caleb Ferguson, Sally C Inglis, Phillip J Newton, Peter JS Cripps, Peter S Macdonald, and Patricia M Davidson. 2014. Social media: a tool to spread information: a case study analysis of twitter conversation at the cardiac society of australia & new zealand 61st annual scientific meeting 2013. *Collegian*, 21(2):89–93.

Elisabetta Fersini, Paolo Rosso, Maria Anzovino, et al. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. *Ibereval@ sepln*, 2150:214–228.

Johan Galtung. 1969. Violence, peace, and peace research. *Journal of peace research*, 6(3):167–191.

Johan Galtung. 1990. Cultural violence. *Journal of peace research*, 27(3):291–305.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Jawad Hossain, Hasan Mesbaul Ali Taher, Avishek Das, and Mohammed Moshiul Hoque. 2023. Nlp_cuet at blp-2023 task 1: Fine-grained categorization of violence inciting text using transformer-based approach. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. Banfakenews: A dataset for detecting fake news in bangla. *arXiv preprint arXiv:2004.08789*.

Md Rezaul Karim, Bharathi Raja Chakravarthi, John P McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 390–399. IEEE.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Hannah Kirk, Abeba Birhane, Bertie Vidgen, and Leon Derczynski. 2022. Handling and presenting harmful text in nlp research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 497–510.

Savannah Larimore, Ian Kennedy, Breon Haskett, and Alina Arseniev-Koehler. 2021. Reconsidering annotator disagreement about racist language: Noise or signal? In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 81–90.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Murat Mengü and Seda Mengü. 2015. Violence and social media. *Athens Journal of Mass Media and Communications*, 1(3):211–227.

Sourabrata Mukherjee, Atul Kr Ojha, and Ondrej Dusek. 2023. Ufal-uld at blp-2023 task 1: Violence detection in bangla text. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Saurabh Page, Sudeep Mangalvedhekar, Kshitij Deshpande, Tanmay Chavan, and Sheetal S. Sonawane. 2023. Mavericks at blp-2023 task 1: Ensemble-based approach using language models for violence inciting text detection. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Md Nishat Raihan, Dhiman Goswami, Sadiya Sayara Chowdhury Puspo, and Marcos Zampieri. 2023. nlpbdpatriots at blp-2023 task 1: Two-step classification for violence inciting text detection in bangla - leveraging back-translation and multilinguality. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Raquel Recuero. 2015. Social media and symbolic violence. *Social media+ society*, 1(1):2056305115580332.

Omar Faruqe Riyad, Trina Chakraborty, and Abhishek Dey. 2023. Team_syrax at blp-2023 task 1: Data augmentation and ensemble based approach for violence inciting text detection in bangla. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam. 2021. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*, pages 457–468. Springer.

Saumajit Saha and Albert Aristotle Nanda. 2023. Banglanlp at blp-2023 task 1: Benchmarking different transformer models for violence inciting text detection in bangla. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2019. Social bias frames: Reasoning about social and power implications of language. *arXiv preprint arXiv:1911.03891*.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understading.

Hrithik Majumdar Shibu, Shrestha Datta, Zhalok Rahman, Shahrab Khan Sami, MD. SUMON MIAH, Raisa Fairooz, and Md Adith Mollah. 2023. Sust_black box at blp-2023 task 1: Detecting communal violence in texts: An exploration of mlm and weighted ensemble techniques. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Md. Tariquzzaman, Md Wasif Kader, Audwit Nafi Anam, Naimul Haque, Mohsinul Kabir, Hasan Mahmud, and Md Kamrul Hasan. 2023. the_linguists at blp-2023 task 1: A novel informal bangla fasttext embedding for violence inciting text detection. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, and Usman Naseem. 2023. Lowresourcenlu at blp-2023 task 1 2: Enhancing sentiment classification and violence incitement detection in bangla through aggregated language models. In *Proceedings of the 1st Workshop on Bangla Language Processing (BLP 2023)*, Singapore. Association for Computational Linguistics.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Author Index