A Study on Contrastive Learning for Bengali Social Meaning

A Thesis

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

Submitted by

M Saymon Islam Iftikar	**190204005**
Farhana Hossain Swarnali	**190204044**
Jannatim Maisha	**190204055**
Sarkar Bulbul Ahammed	**190204110**

Supervised by

Mr. Faisal Muhammad Shah

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

April 2024

CANDIDATES' DECLARATION

We, hereby, declare that the Thesis presented in this report is the outcome of the investigation performed by us under the supervision of Mr. Faisal Muhammad Shah, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this Thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

M Saymon Islam Iftikar
190204005

Farhana Hossain Swarnali
190204044

Jannatim Maisha
190204055

Sarkar Bulbul Ahammed
190204110

CERTIFICATION

This Thesis titled, **"A Study on Contrastive Learning for Bengali Social Meaning"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in April 2024.

Group Members:

M Saymon Islam Iftikar	**190204005**
Farhana Hossain Swarnali	**190204044**
Jannatim Maisha	**190204055**
Sarkar Bulbul Ahammed	**190204110**

Mr. Faisal Muhammad Shah
Associate Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dr. Md. Shahriar Mahbub
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

This thesis and the accompanying research owe their existence to the invaluable guidance and support provided by our supervisor, Mr. Faisal Muhammad Shah, Associate Professor in the Department of Computer Science and Engineering at Ahsanullah University of Science and Technology.

Grateful acknowledgment is extended to Professor Dr. Md. Shahriar Mahbub, the Head of the Department of Computer Science and Engineering at Ahsanullah University of Science and Technology. Additionally, appreciation is extended to the university librarians, research assistants, and fellow students whose contributions and inspiration significantly influenced our work.

Lastly, heartfelt thanks go to our parents for their unwavering encouragement and support throughout our educational journey.

Dhaka
April 2024

M Saymon Islam Iftikar

Farhana Hossain Swarnali

Jannatim Maisha

Sarkar Bulbul Ahammed

ABSTRACT

Across various domains, Contrastive Learning (CL) has already proven to be a powerful technique but using the Bengali language in the domain of Natural Language Processing (NLP) its' application is still unexplored. In this research, we introduce the implementation of CL in Bengali NLP by presenting a comprehensive benchmark study on two distinct datasets for binary class classification: Bengali Hate Speech, and Rokomari Book Review, and two distinct datasets for multi-class classification: Bengali-Hate-Speech-Dataset, Daraz Product Review. The efficiency of the supervised contrastive techniques is emphasized by our methodology. We have implemented a contrastive learning technique through Bangla Bert. The superiority of supervised contrastive learning techniques over traditional Cross Entropy (CE) methods has been showcased by our result. Detailed experiments reveal performance variations across datasets, models, and hyperparameters but ensure the superiority of CL techniques over CE. For binary class classification, we got our best F1 score in a CL method named TACT which is 98% which outperformed traditional CE by 4% as we got 90% F1 score for CE and also for multi-class classification, TACT (91%) outperformed CE (88%) by 4%. Our findings show the significance of contrastive learning methods in low-resource settings, contributing to the advancement of Bengali NLP research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Nowadays, various online platforms have emerged as powerful arenas for individuals to share their thoughts and emotions with a global audience. These online platforms made it easier for different ideas to come together and create forums for unrestricted discussion. While this free flow of perspectives has its merits, it is not without drawbacks. A good side of online platforms is that they foster an environment where people can engage in the exchange of diverse ideas, contributing to personal and intellectual growth. Conversely, the negative aspect involves the unfortunate occurrence of individuals using harmful language to disparage others, a behavior that is deemed unacceptable in a civilized society. Consequently, online platforms provide a vast resource for examining thought processes and behavioral patterns. Bengali, the sixth most spoken native language worldwide gives content published on social networking sites a unique perspective that draws researchers to the field. Numerous research studies have already delved into Bengali content shared on online platforms, exploring various sectors and seeking reviews from individuals. Investigating online platform comments or posts that involve the expression of hateful emotions has been a focal point of research. Additionally, other types of Bengali content have served as valuable experimental material in diverse research studies. Our research, specifically, aims to enhance the comprehension of different Bengali online platform contents by applying Contrastive Learning. In recent times, Contrastive Learning has demonstrated remarkable improvements across various fields of tasks, yet its potential within Bengali-related NLP remains largely unexplored. Also, the sage of Contrastive Learning helps to provide better performance for a variety of tasks in deep learning techniques. This study seeks to uncover and leverage the power of Contrastive Learning to enhance the understanding of Bengali online platform content. In our work, we have utilized contrastive learning, specifically supervised contrastive learning for acquiring better performance in

Bengali NLP tasks. We have used several different supervised contrastive learning methods to train models. In this work, we demonstrate that models trained using CL-based methods outperform models trained using CE through an extensive experimental study.

To sum up, our contributions are the following:

- We have used Supervised Contrastive Learning involving the Bengali NLP tasks. According to our knowledge, this is the first work that uses Supervised Contrastive Learning for Bengali NLP tasks.
- We have also shown that CL-based methods outperform CE-based finetuning of models.

1.2 Motivation

In our pursuit to delve deeper into understanding sentiment in Bengali text, we aim to address the limitations of existing methods. Take, for instance, the phrase **"এই গানটি মনে রাখতে ভুলবো না"** (This song will not be forgotten easily) in Bengali, which conveys a positive sentiment about a memorable experience. However, the inclusion of **"না"** (not) in the context of ”remembering” adds complexity to the sentiment. This complexity introduces a positive twist to the song, potentially leaving a lasting favorable impression.

Our approach combines the ”push and pull” strategy, known as contrastive learning, with a pre-trained Bengali BERT model. The goal is to accurately capture the nuanced semantic choices embedded within sentiment-laden expressions.

By integrating these methodologies, we aim to enhance our understanding of sentiment analysis in Bengali text, accounting for its intricacies and capturing the subtle nuances that contribute to the overall sentiment conveyed. Through this approach, we seek to provide a more comprehensive and nuanced interpretation of sentiment, enabling a deeper understanding of the complexities inherent in Bengali language expression.

1.3 Objective

- **Application of Supervised Contrastive Learning:** The Supervised Contrastive Learning is used for the task of increasing the performance of Bengali Natural Language Processing. This is done by Supervised Contrastive learning as it captures the complexity hidden in the sentences. This is the process that helps the models for improving performance.

- **Evaluation of Performance Across Dataset Sizes:** One of the main objectives is to analyze how different models perform for varying sizes of datasets in the Bengali NLP tasks. Using different amounts of data for training the model, the objective is to observe the stability of the models when the datasets are of varying sizes. Having stable models is important for using these in the real world.

- **Examination of Adversarial Training Synergy:** Analyze the results of combining adversarial training with supervised Contrastive Learning. And observe if the model performs even better with adversarial training techniques—particularly in terms of resilience and generalization.

- **Analysis of Cross-Dataset Performance:** Analyzing the performance of the models for different kinds of datasets is an important task. This helps to understand if the models are capability of the models. So one of the objectives is to examine the resilience and generalization capacities of supervised contrastive learning models by analyzing their performance on several Bengali NLP datasets. This research will assess the model's applicability and dependability for real-world applications by revealing how effectively it transfers knowledge and adjusts to new linguistic contexts found in datasets that have not yet been observed.

- **Comparative Study between Contrastive Learning and Cross-Entropy Loss:** The objective includes the showcase of the benefits of contrastive learning above cross-entropy loss. For this, the comparison between cross entropy loss and contrastive losses is required. So, the objective of this work is to find out the ability of these methods to capture the context of the sentences. This in turn helps to find out the methods ability to understand the language. With the help of the results for the cross entropy and contrastive learning-based methods, the one that performs better can be obtained. Also, the results provide important insights about the two approaches.

Chapter 2

Literature Review

2.1 Paper Reviews

2.1.1 Hate Speech detection in Bengali language: A dataset and its baseline evaluation [1]

Methodology:

This paper describes the newly introduced Bengali text-based dataset called as Bengali Hate Speech Dataset (BHSD) for identifying hate speech. The dataset comprises 5,126 comments collected from comment sections on YouTube and Facebook, manually annotated. Three annotators who were independently marked each comment as not hate speech or hate speech for each comment. The resulting dataset is balanced where approximately half of the comments are labeled with hate speech while the remaining half from the other half is labeled as non-hate speech. The authors then proceed to cross-validate some baseline hate speech detection models composed of Support Vector Machines (SVMs), Random Forest, and Naive Bayes classifiers. The models were trained and then tested against the BHSD dataset. From the evaluation results, the SVM model produced the best performance at 87.5%. Random Forest and Naive Bayes classifiers had an accuracy of 70.1% and 52.20%, respectively

Strength:

- This dataset is a resource of much need for research in hate speech detection in the Bengali language. It is unique and the first publicly available dataset consisting of only Bengali text expressly constructed for classifying hate speech.

- The different kinds of hate speech are fairly uniform across the dataset.
- The paper has been an in-depth review of multiple baseline models that offer insightful statistical understanding on how the various techniques to hate speech detection in Bengali fare.

Weakness:

- The dataset is of small size consisting of only 5,126 comments. Therefore there might be a limitation inducing its generalizability over other sources of Bengali text.
- This is something that the authors say nothing about in terms of how well they expect their models to transfer on real data.
- No explicit mention of the model's performance in actually being used to conduct attacks on YouTube and Facebook comments.

2.1.2 Contrastive Learning of Sociopragmatic Meaning in Social Media [2]

Methodology:

The paper proposes InfoDCL, a novel contrastive learning framework to sociopragmatic meaning learning in social media. InfoDCL leverages distant supervision for learning general knowledge of all SM tasks and introduces corpus-level information for capturing inter-class relationships while enhancing the uniformity of PLM and preserving underlying semantic structure.
There are three main components of InfoDCL:
Distantly supervised contrastive loss (LDCL): The LDCL leverages the surrogate labels(e.g., emojis) to build positive and negative samples for contrastive learning. The weight of each negative sample is the normalized point-wise mutual information(npmi) between that negative sample and the anchor one.
The objective of surrogate label prediction (SLP): In this, the feature learned in SLP optimizes the encoder for the emoji prediction task using cross-entropy loss. After this, the classification probabilities of SLP are used to weigh the negatives in LDCL.
Corpus-aware contrastive loss (CCL): CCL leverages the pointwise mutual information (PMI)–based corpus level information for mining the relations between surrogate labels. The weight of a negative sample will be as its PMI with the anchor sample.

Strength:

- InfoDCL is a general-purpose framework for learning sociopragmatic meaning in social media that does not require task-specific labels, and hence it can be applied to a wide range of SM tasks.
- The reason for this difference may be the fact that InfoDCL uses corpus-level knowledge to indirectly encode inter-class relationships and does not lose semantic structure in the learned representations. In turn, it gives a more informative and generalizable representation.
- InfoDCL beats both baselines and is compared to state-of-the-art models across various SM tasks including emotion recognition, hate speech detection as well and irony detection.

Weakness:

- InfoDCL is the computationally expensive framework as the generalization of InfoDCL to larger datasets relies on large-scale training of the language model and optimization over multiple objectives.
- InfoDCL is evaluated on a small dataset of tweets acquired from the English language. It yet remains to see how well InfoDCL would generalize to other languages or larger datasets

2.1.3 ConOffense: Multi-modal multitask Contrastive learning for offensive content identification [3]

Methodology:

The paper introduces a novel multi-modal, multi-task contrastive learning framework, named ConOffense, for offensive content identification. ConOffense leverages both text and audio modalities to capture offensive information, in addition to the mechanism of multi-task learning, which is introduced to enhance the generalization ability of the model.

ConOffense mainly consists of two parts:

Multi-modal contrastive learning (MCL): MCL leverages both the text and the audio modalities to learn offensive representations. Pre-trained language models are used to obtain the text embeddings. Similarly, pre-trained audio encoders are used to obtain the audio embeddings. A multi-modal attention sub-layer is employed to combine the two

modalities together.
Multi-task contrastive learning (MTL): MTL sculpts the training of the model across multiple tasks for identifying offensive content. This generalizes the model representations more than being learned for a single task.

Strength:

- ConOffense is an innovative efficient framework of offensive text and audio content detection. It exploits textual and audio information for the extraction of offensive information, and multi-task learning combines both textual and audio information to leverage on the generalization ability of the model.
- Consequently, ConOffense attains state-of-the-art performance on different benchmarks for identifying offensive content, including both the Hateful Content Detection and OffensEval tasks.

Weakness:

- The application of numerous models in ConOffense needs immense computational resources, such as training associated with multi-modal contrastive learning.
- Consequently, if noisy input data gets, or if input data characterizing errors is obtained, model performance can be degraded as ConOffense is quite sensitive to the quality of the input data.

2.1.4 Improved Text Classification via Contrastive Adversarial Training [4]

This research paper proposes a method to regularize the fine-tuning of Transformer-based encoders for text classification. During fine-tuning adversarial examples are generated which along with the clean texts are used to teach the model to work well on unseen data with the help of contrastive learning.

Methodology:

In this paper method for generating adversarial examples are introduced and the method of Contrastive Adversarial Training(CAT) that uses these examples to perform contrastive learning with clean examples is proposed.

Adversarial Examples: Adversarial examples are imperceptibly perturbed input to a model that are created to mislead the model to give wrong output for unseen data. Fast Gradient Sign Method is used here to generate adversarial examples. Here perturbation of the word embedding matrix of Transformer-based encoders is done to generate adversarial examples.

Contrastive Learning: In this paper contrastive learning is used as an additional regularizer during fine tuning process. Contrastive learning is used for pushing the clean examples and their corresponding perturbed examples close to each other in the representation space while pushing apart examples not from the same pair. This process helps the model to be noise invariant.

Strength:

- The paper first introduced CAT which is a simple and general method for the model to be more generalized for text classification tasks.
- With the help of adversarial training and contrastive learning, the model outperformed the standard fine-tuning method for text classification.
- Adversarial training helps the model defend against adversarial attacks.

Weakness:

- In this paper, the discussion is limited to only white box adversarial attacks i.e., assuming access to model architecture and parameters.
- This paper is only concerned about improving text classification and does not focus on other NLP tasks.

2.1.5 Improving Health Mentioning Classification of Tweets using Contrastive Adversarial Training [5]

This paper deals with Health Mentioning Classification (HMC) and classifies an input text as health mention or not. This classification serves for early detection and tracking of a pandemic. This paper uses adversarial training and contrastive learning for better classification output. The paper also used explainable AI to analyze the results.

Methodology:

The paper describes the basics of the transformer-based encoder for text classification. Transformed methods are good at capturing the textual meaning of the words and it has succeeded in many natural language processing tasks.

Strength:

- The paper shows that contrastive adversarial training as a regularizer improves the performance of the model.
- The improvement in performance has been explained using the power of explainable AI.

Weakness:

- Focus is given on only health mentioning tweets.
- Classification of only 10 diseases is done.

2.1.6 Generalizable Implicit Hate Speech Detection Using Contrastive Learning [6]

Methodology:

Previously various method was introduced like the lexicon-based method, a neural-based method that detects explicit hate speech but fails to detect implicit hate speech. In the previous year, there were a few CLs used in the shot learning setup. Motivated by this, here CL is proposed to improve the generalization of implicit hate speech detection in cross-dataset. The reason why CL is used is that CL is good for working on positive things. In CL, positive data are pulled whereas negative data are pushed and apart from positive data.

Here, two positive sampling strategies followed. One is AugCon and another is ImpCon. In AugCon, lexically different but semantically same as the original post are generated. In ImpCon, ImpCon leverages implications as positive samples of hateful posts.
Here, cross-dataset is used to see the generalization ability of a model. Cross-dataset is better than in-dataset because in-dataset works on similar data in training whereas

cross-dataset uses different data.

Since, the cross-entropy loss has limitations on making large inter-class margin or intra-class compactness, fine-tuning using only cross-entropy loss can result in suboptimal generalization. They proposed combined contrastive loss with cross-entropy loss to train generalizable implicit hate speech detectors.
-Augmented post positive samples
-Implication as positive samples

Strength:

- Implicit hate speech detected.
- CL outperforms any other fine-tuning process.

Weakness:

- Give false result if data is not same group or unseen target group.
- HateBERT performance result was not enough good.

2.2 Research Gap

A notable study gap in this language setting is highlighted by the literature assessment, which emphasizes the scant investigation of social meaning in Bengali. The necessity of this study is further increased by the lack of previous research on Contrastive Learning in Bengali, which highlights the need for innovative methods to improve comprehension and applications in Bengali NLP tasks.

Chapter 3

Background Study

3.1 Contrastive Learning:

Some approaches in machine learning use contrastive learning as a method for creating informative representations of data by distinguishing between similar and dissimilar examples. In the process of distinguishing between similar and dissimilar instances, the method aids models in weakening the resemblance among different classes but at the same time strengthening it within a specific class. To make this happen, contrastive learning utilizes multiple loss functions. One of the applications of contrastive learning is its ability to separate instances from different classes and at the same time allow those that belong to a particular class to come together. Fundamentally, in contrastive learning, we use contrastive loss functions as an attraction.

From the very beginning of its inception, contrastive learning works on the basis of attracting similar samples towards each other and repelling dissimilar ones by using contrastive loss functions. CL draws together neighboring sentences with its use of contrastive loss functions and pushes apart non-neighbors [7]. SimCSE [8] was proposed as an unsupervised method for generating sentence embeddings with CL. SimCSE achieves this task by taking a sentence as input and predicting itself using dropout to augment noise during training.

To discern similarities and differences among samples, the contrastive learning method is very important for extracting meaningful representations. The model understands the underlying structure in data better by using this approach that applies contrastive loss functions leading to superior performance in various tasks.

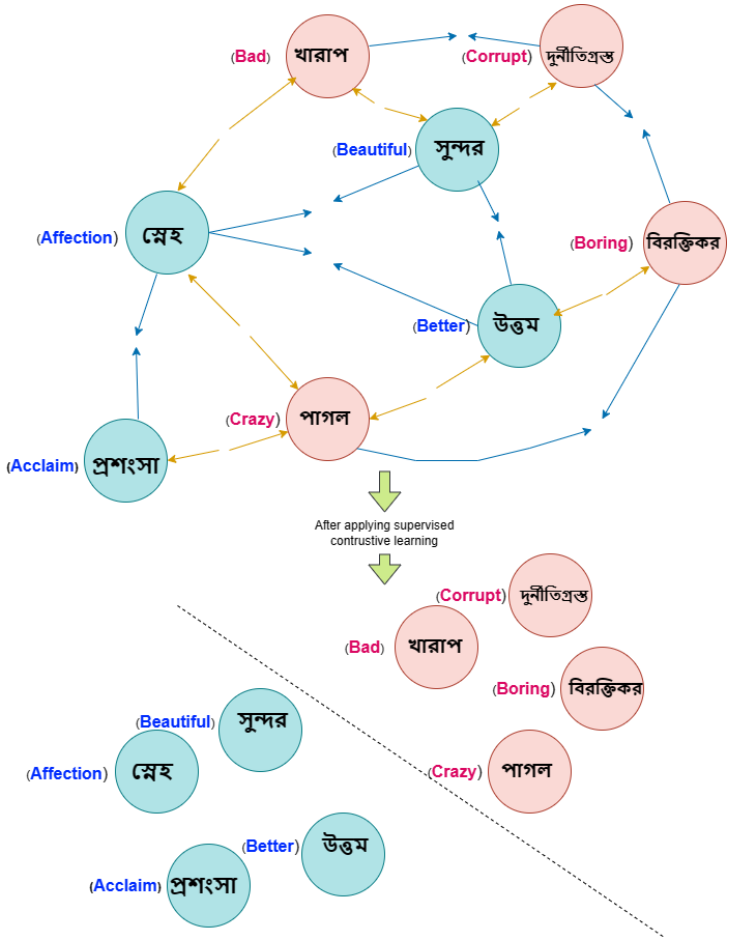

Figure 3.1: The working procedure of Contrastive Learning is demonstrated here. Different class representations are pushed apart to emphasize their dissimilarity, similar class representations are pulled closer to emphasize their similarity.

3.2 Supervised Contrastive Learning:

The work of [9] introduced a fully supervised framework for expanding Contrastive Learning. The reason behind this was to maximize the availability of plenty of label information. In this case, samples with the same class label need to be organized closely together in the embedding space while pushing away clusters of samples from different labels at the same time. Using these labels makes it easier to choose positive examples for anchors so as not to have any false negatives and positives that can be easily avoided. Labels are important for determining which class each sample falls into, which is important for selecting appropriate positives and negatives for each anchor.

The work of [9] has proposed SupCon Loss as a novel loss function specifically designed for supervised contrastive learning. By doing so, this approach allows many positive and negative samples associated with one anchor thus ensuring that the model learns better by properly using the label information available during training sessions. With SupCon Loss incorporated into supervised contrastive learning, there is an opportunity

to enhance learning by maximizing the use of labeled data hence leading to more robust and discriminative representations in embedding space.

3.3 Adversarial training (AT):

A machine learning technique called adversarial training (AT) is a powerful tool in machine learning that tries to expose the models used in machine learning to specially crafted adversarial examples during training, which can make it more robust and have better generalization abilities. First proposed by [10], adversarial training has gained prominence as a defense strategy against various weaknesses and imperfections, particularly for computer vision tasks.

The fundamental idea of adversarial training is to include adversarial samples—inputs that are intentionally changed to mislead the model—in the training data. When these adversarial examples are added to the training process, models get better at handling slight variations and unexpected inputs. As such exposure facilitates understanding of underlying data patterns, therefore enabling accurate identification of unique inputs of real-world scenarios.

Basically, Adversarial Training acts as a preventive action toward strengthening machine learning models' immunity against possible attacks or unforeseen fluctuations in input data. Models that involve such kinds of adversarial examples in their regime can better adapt to new conditions of operation leading to improved performance and resilience against adversaries.

3.4 Contrastive Adversarial Training (CAT):

Adversarial training in Natural Language Processing (NLP) has become an important method to improve the robustness and effectiveness of models. However, applying conventional computer vision adversarial training techniques directly to NLP tasks is challenging due to the discrete nature of textual input.

Researchers have put forward several modifications and extensions for adversarial training aimed at NLP applications. For example, [**?**] proposes one such extension involving perturbations made on word embeddings in NLP models. In this approach, perturbations are introduced into the textual input thereby increasing noise in the model hence enhancing its ability to handle adversarial variations in data.

Contrastive Adversarial Training (CAT) is a recently developed technique for adversarial

training in NLP with a focus on text classification tasks. CAT applies Transformer-based models widely used in NLP and introduces contrastive objectives for better noise-invariant representations. A key modification that CAT makes is altering the word embedding matrix within Transformer encoders so that they can learn robust representations that are immune to noisy or adversarial perturbations.

To recap, breakthroughs like the idea from [**?**] and Contrasting Adversarial Teaching help with the problems of using adversarial training for language tasks. These methods try to make models stronger and better at generalizing, by adding noise and improving how models learn noise-resistant representations. In the end, this pushes the abilities of language systems to handle adversarial inputs further.

3.5 Pretrained Models:

3.5.1 Bert-Based:

In the past few years, natural language processing (NLP) was totally transformed. BERT or Bidirectional Encoder Representations from Transformers played a pivotal role. It was a significant breakthrough introduced by Devlin et al. in their highly influential research paper [11].

Conventional NLP models analyzed text in one direction only. They often failed to fully grasp the context, leading to subpar performance across different language processing tasks. BERT solved this issue by using deep bidirectional transformers - an innovative approach for better context comprehension.

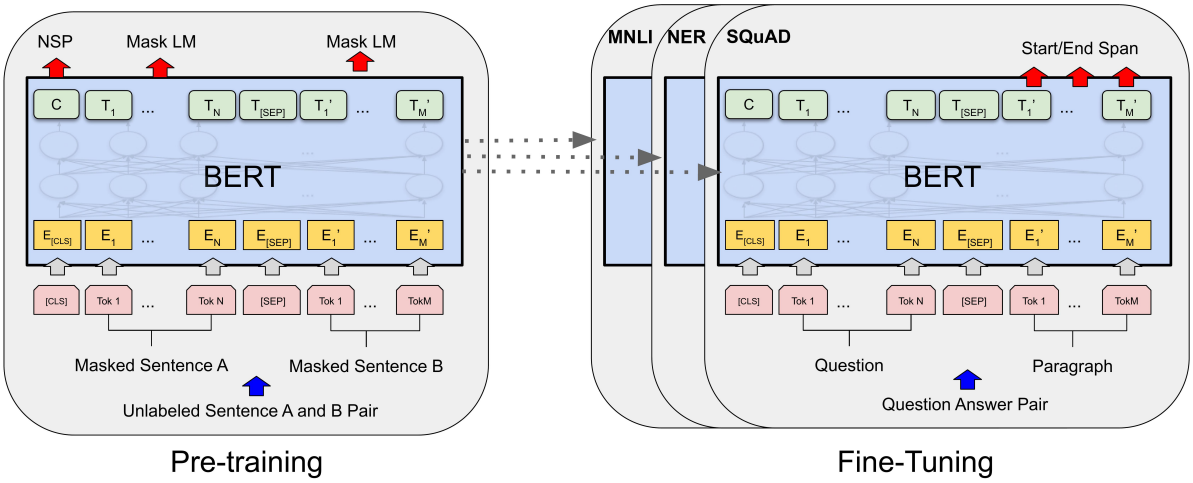

Figure 3.2: BERT pretrained model
[12]

BERT uses a masked language model pre-training method. It predicts missing words

in sentences. This helps BERT understand language structure [11]. BERT creates contextualized representations that show deeper comprehension of language. This innovation influenced later transformer-based models. BERT became standard for many NLP tasks. It is used for named entity recognition, question answering, sentiment analysis, and natural language inference [11]. BERT provides a foundation for advancing linguistic understanding and future research.

BERT tokenizes input to convert it into numbers. It pre-trains on data by masking some words, learning to predict them. This bidirectional approach lets it understand context both ways. Next, BERT fine-tunes using labeled task data. Embedding layers represent words numerically. Encoder blocks process contextual information. A pooler layer aggregates this for output. BERT's architecture captures complex language patterns. It enables diverse NLP applications through these core features.

3.5.2 Electra-Based:

Natural language processing (NLP) has a new pretraining method. It's named Efficiently Learning an Encoder that Classifies Token Replacement Accurately (ELECTRA). This technique is inspired by Replaced Token Detection (RTD) [13]. The ELECTRA paper introduces this approach. It allows for better learning from input tokens compared to traditional methods. ELECTRA is more sample-efficient for pretraining language models.

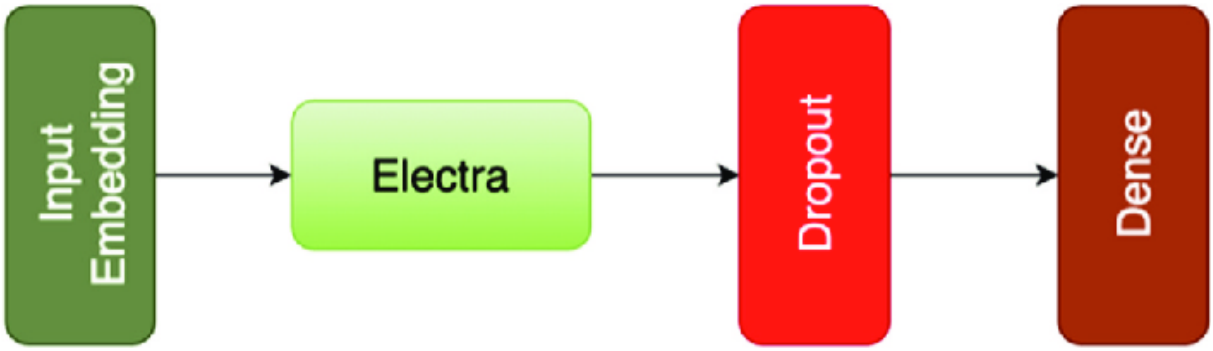

Figure 3.3: Electra-based pretrained model [14]

Specific tokens within the input sequence get replaced by viable options from a small generator network in the ELECTRA framework. A discrimination model then gets trained to predict whether each token comes from the generator network or not. This approach principally involves the discrimination model identifying original tokens versus tokens the small generator network produced.

A big change from BERT's Masked Language Model (MLM) technique is how ELECTRA

uses Replaced Token Detection (RTD) instead. Unlike masking some input tokens, RTD allows the model to utilize every token for learning. The discrimination model spots which tokens got swapped, so ELECTRA analyzes the full input sequence. This adjustment boosts its performance and efficiency compared to masked learning on partial inputs.

ELECTRA is a significant step towards better NLP pretraining, learning more from the data and with less data compared to the traditional unsupervised method like MLM. Especially by using the RTD and training a discriminator to distinguish between original and replaced tokens, we learn more effectively and affordably from the entire input context and significantly improve the learning effect and large-scale language capability.

3.6 Label-Aware Contrastive Learning (LCL):

Label-Aware Contrastive Learning (LCL) is an advanced technique that extends contrastive learning to consider class label relationships. Traditional contrastive learning focuses on distinguishing between individual samples while LCL takes into account inter-label relationships which is best for fine-grained classification tasks.

Fine-grained classification implies distinguishing between classes that are closely related with little variation among them. For example, in sentiment analysis, fine-grained classification would involve making distinctions between nuanced sentiment categories like “Positive” and “Very Positive.” Such classes sometimes have overlapping meanings which makes classification difficult.

According to Suresh and Ong (2021) [12], LCL is a Contrastive Loss function that considers the relationship between labels. For instance, they propose a weighted network for learning these relationships in addition to the principal encoder of their model. Therefore, the weighting network assigns higher weights on the labels that are hard to distinguish from others thereby making it easier for models to differentiate among similar classes.

The estimated probabilities obtained by prediction probability of labels with higher weight guide the process of calculating class probabilities per sample. The loss function then integrates these estimates, usually combined with NTXent loss during training.

In future studies, examining how effective LCL would be on various NLP problems such as fine-grained classification and multi-label classification could help to understand whether LCL can enhance the performance and robustness of models. Moreover, evaluating the scalability of LCL with larger datasets and more complex architectures would make it widely applicable in natural language processing.

3.6.1 Fine-Grained Classification

Fine-grained classification is the task of distinguishing between classes that are similar and differ from the rest by some very subtle features. Let's explain it through the example with sentiment analysis. In sentiment analysis, one can classify what sentiment a test has, such as a review or post from social media.

Commonly, when we refer to sentiment analysis just as it is, we consider three classes "Positive", "Neutral", and "Negative". However, in fine-grained sentiment analysis, we have to find a difference in more subtle sentiments.

In traditional sentiment analysis, often classes like "Positive," "Neutral," and "Negative" are present. These are some larger classes of sentiment. Conversely, the fine-grained sentiment seeks to distinguish between classes that encompass the subtle and deliberate nuances of sentiment that are overlooked by other, larger classes.

The "Positive" sentiment shows satisfaction. On the other hand, "Very Positive" has stronger delight or excitement. While both sentiments are positive, they differ in how strongly positive emotions are felt. "Positive" is still approval, but "Very Positive" portrays much greater enthusiasm.

In Bengali sentiment analysis, imagine having to classify text into "Positive" or "Very Positive." This task captures nuanced differences in how positivity is expressed in Bengali writing.

Positive Sentiment:

"আজকে আমার খুব ভালো লাগলো, আমি ভালো সময় কাটালাম।" (I had a particularly good day today, I had a good time.)

This example conveys a positive sentiment, indicating that the speaker had a good day and enjoyed their time. However, the sentiment expressed is not excessively enthusiastic or intense.

Very Positive Sentiment:

"আজকের দিনটি অত্যন্ত সুন্দর ছিল! আমি অবশ্যই লালন করব।" (Today's day was extremely beautiful! I will definitely cherish it.)

In contrast, this example expresses a sentiment that is highly positive and enthusiastic. The use of words like "অত্যন্ত সুন্দর" (extremely beautiful) and "অবশ্যই লালন করব" (will definitely cherish) indicates a higher level of satisfaction and excitement compared to the

previous example.

Accurately classifying texts into these delicate emotion categories would be the aim of a fine-grained sentiment analysis assignment for Bengali, taking into account the tiny variations in positivity represented in the Bengali language. The degree of granularity facilitates a profound grasp of Bengali texts' sentiments which is helpful for customer comments, social media views, and product assessments in Bengali contexts.

Chapter 4

Methodology

4.1 Overview

In our study, Contrastive Learning (CL) methods are utilized to help the model learn in an efficient way. We have used contrastive learning in supervised setting which was first introduced in the paper called **Supervised Contrastive Learning** [**?**] . In our work, supervised contrastive learning methods are used in relation to Bengali Social Meaning tasks such as hate speech detection, book review, toxic comment detection, and product review. The four supervised contrastive learning methods used in our paper are **Supervised Contrastive Loss (SCL)**, **Contrastive Adversarial Training (CAT)**, **Token-level Adversarial Contrastive Training (TACT)** and **Label-aware Contrastive Loss (LCL)**.

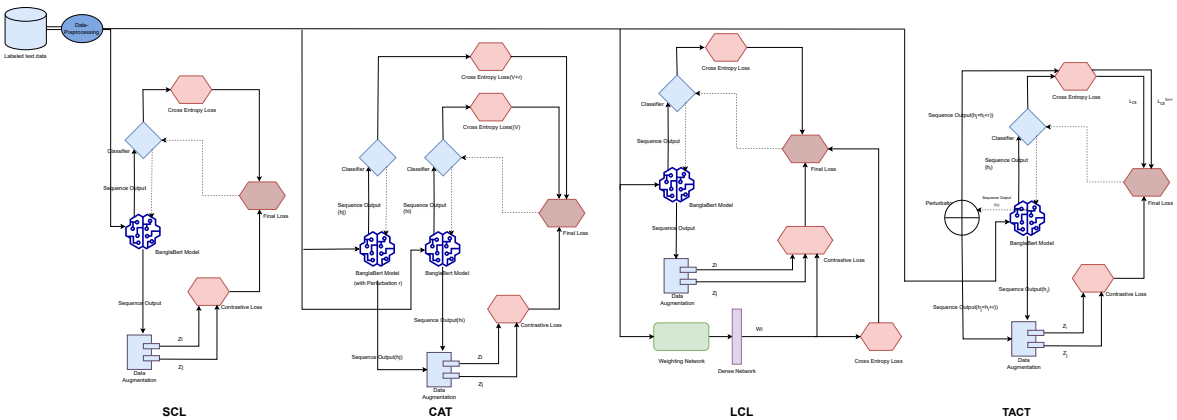

Figure 4.1: Proposed Methodology

In the proposed methodology, the first step involves text that is taken from the datasets. The text data are preprocessed in the next stage. The preprocessed text data are the

input of the model which is sent to Bangla-Bert. After the processing of the data, then using Bangla-Bert , which gives representation of the input text. This representation is used to calculate contrastive loss. The loss is then backpropagated to optimize the model.

4.2 Dataset Preprocessing

4.2.1 Custom Dataset initialization

The model and the raw dataset are connected via the Custom Dataset class. Important settings including the dataset, tokenizer, maximum sequence length, label-to-index mapping, and column names for text and labels are initialized within it. The framework for data processing within the dataset is established by this initialization.

4.2.2 Processing Text and Labels

The extraction of relevant information from the dataset is a major effort in data preparation. Finding the text and label columns in the dataset is required for this. Labels supply the ground truth for training or assessment, whereas textual data is essential for model input. Number indices are mapped to labels, which are frequently in their original textual or category form. During training, the model can efficiently comprehend and handle the labels thanks to this mapping.

4.2.3 Tokenization

Tokenization is a fundamental step in natural language processing (NLP) tasks. It involves breaking down text into smaller units called tokens, which are usually words or subwords. These tokens serve as the basic building blocks for the model input. The tokenizer, typically provided by libraries such as Hugging Face's Transformers, converts raw text into tokenized representations. Each token is associated with a unique identifier (token ID) that the model understands. Tokenization also involves handling aspects like special tokens (e.g., [CLS], [SEP] in BERT), padding sequences to a uniform length, and truncating sequences that exceed a specified maximum length. These measures ensure consistency in the input data format, which is essential for batch processing.

4.2.4 Creating Dictionary of Inputs

The text is arranged in a structured way that may be fed into the model after it has been tokenized and encoded. Usually, this entails storing the various input data components in a dictionary. Depending on the needs and design of the model, text inputs and extra data like attention masks and token type IDs may also be added. To aid in supervised learning, label indices—obtained through label mapping—are additionally incorporated into the dictionary.

4.2.5 Data Loader Setup

During training, assessment, and testing, data loaders are in charge of effectively feeding the preprocessed data into the model. They optimize the training process for performance by managing operations like batching, shuffling, and parallelization. In order to build up data loaders for various dataset splits (such as training, validation, and testing), the regular encode function first initializes instances of CustomDataset. It sets the prerequisites for batch processing, including the size of the batches and the manpower required for processing them simultaneously.

Preprocessed data may be seamlessly integrated into the model training loop thanks to these data loaders, which are an essential component of the training pipeline. Thus, our data preprocessing entails converting unstructured text input into a format that the model can use efficiently. Tokenization, label mapping, and data structuring are just a few of the tasks it includes in order to get the input data ready for testing, assessment, and training.

4.3 Model Architecture

4.3.1 Conversion of input text data into appropriate format:

The supplied text sequences are put through a thorough formatting procedure made possible by the BanglaBert Tokenizer to prepare the input data for use in the pre-trained BanglaBert Model. Incorporating specialized tokens and segmenting the text into discrete tokens are the two purposes of this tokenization process, which makes the sequence compliant with the BanglaBert Model's specifications.
The input text sequence x_i, y_i is broken down into individual tokens as part of the tokenization process, and additional special tokens are introduced. To ensure that the input sequence is in line with the predetermined maximum length limits given by the model,

the tokenizer is also essential for padding and truncating the sequence.
The BanglaBert Tokenizer is used to obtain the encoded input IDs for a given training sample indexed by i (i=1,..., N), where xi is the input text sequence. This means that the token sequence for an input text sequence looks like this: x_i= [CLS], t_1, t_2,..., [SEP]. Each token is then translated into the matching encoded input ID. Then, using the encoded input IDs that it received from the Bert Tokenizer, the pretrained Bangla Bert Model creates vector embeddings for every token in the input sequence. The contextual information associated with each token is captured by these embeddings, which help to create a complete representation of the input sentence in the pre-trained BanglaBert Model.

$$H^L = [h^L_{[\text{CLS}]}, h^L_1, h^L_2, \ldots, h^L_T, h^L_{[\text{SEP}]}] \tag{4.1}$$

$$h^L_{[\text{CLS}]}, h^L_1, ..., h^L_T, h^L_{[\text{SEP}]} = P_{\text{LM}}([\text{CLS}], t_1, ..., t_T, [\text{SEP}]) \tag{4.2}$$

here, H_L represents the contextualized vectors produced by the PLM(pre-trained language model), here L represents the number of layers in the model. A softmax classifier is used for the derivation of the probability distribution of both classes. the final hidden state $h_{[}CLS]$ of the [CLS] token (from bert) is used here.

$$p(y_c \mid h_{[\text{CLS}]}) = \text{softmax}(W h_{[\text{CLS}]}), \quad c \in C \tag{4.3}$$

here W is a trainable parameter and C is the number of classes.
The main goal of the model training is to minimize the cross entropy loss(CE), Formulated as:

$$\mathscr{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log(p(y_{i,c} \mid h^i_{[\text{CLS}]})) \tag{4.4}$$

Again, the contextualized vectors (H_L) (hidden states of the last layer of the Bert model for each token) act as Sequence output for our Bangla Bert classifier and Data Augmentation as shown in the figure [4.2] and figure [4.3].

4.3.2 Supervised Contrastive Loss (SCL):

Our research utilizes Supervised Contrastive Loss (SupCon) to strengthen our model's ability to distinguish between different classes. This loss function brings together representations of the same class in the embedding space while separating representations of different classes. To apply SupCon loss, we employ the PyTorch Metric Learning library, which offers a range of tools for metric learning tasks in PyTorch. This library helps us effectively implement the loss function, enhancing our model's discriminative power.

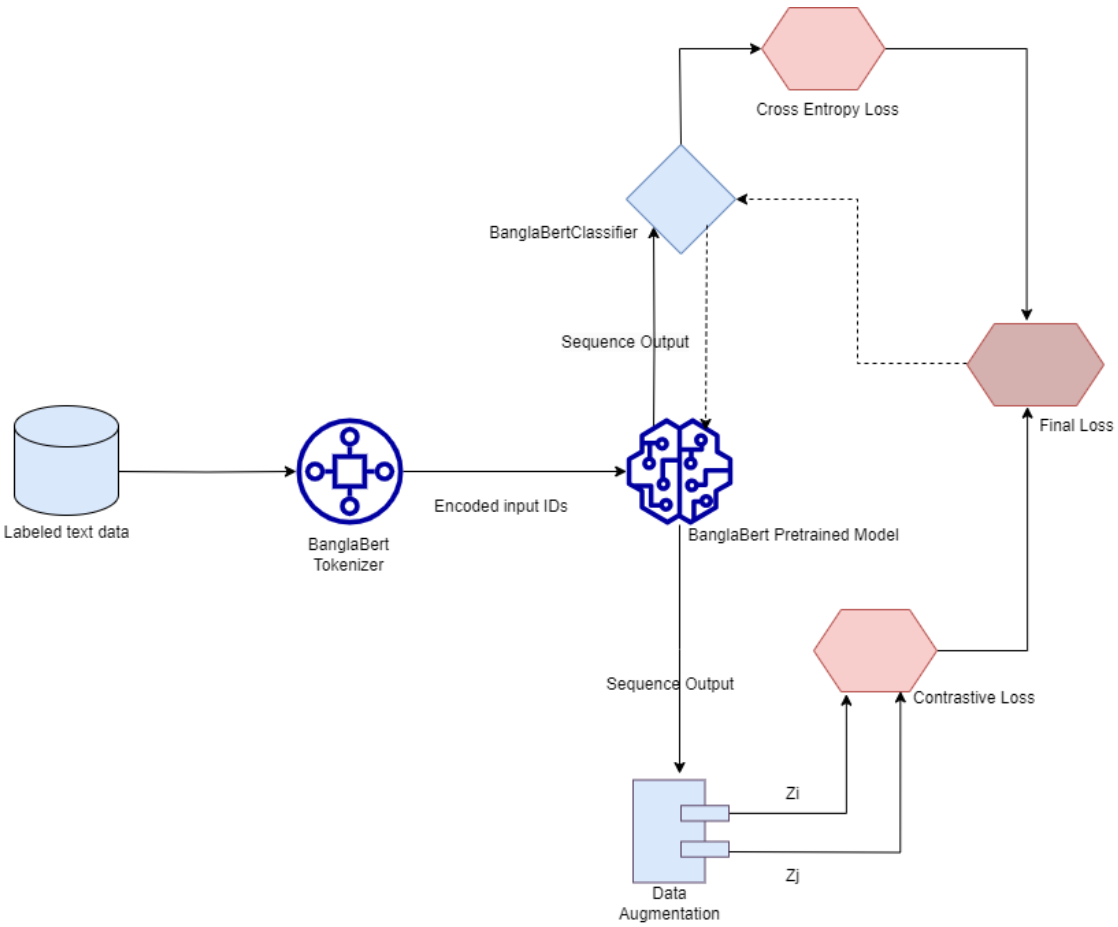

Figure 4.2: Model Architecture for SCL

Central to our implementation is the $[L_{\text{out}}^{\text{sup}}]$ function, which plays a crucial role in computing the supervised contrastive loss. This function ensures that the contrastive loss is calculated in a supervised manner, utilizing class labels to guide the learning process. The specifics of the $[L_{\text{out}}^{\text{sup}}]$ function, including its formulation and implementation details, are elaborated in reference [**?**], where it was initially introduced.

$$\mathscr{L}_{\text{out}}^{sup} = \sum_{i \in I} -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \left(\frac{\exp\left(\frac{z_i \cdot z_p}{\tau}\right)}{\sum_{a \in A(i)} \exp\left(\frac{z_i \cdot z_a}{\tau}\right)} \right) \tag{4.5}$$

In our study, we use supervised contrastive (SupCon) loss to improve the way our model learns representations. This loss function takes into account the advantages and disadvantages of using reference points or "anchors" (called z_i). Instances belonging to the same category as the anchor z_i are considered positive examples, while instances from other categories related to the anchor are considered negative examples.

To enrich our analysis, we introduce dropout-based data augmentation into our framework. The technique involves creating an alternative representation for each instance by extending it to ensure that both representations have the same class label [as shown in

figure 4.2]. This expansion process helps improve the robustness and generalization ability of our model.

The main goal of using the SupCon loss is to minimize the difference between the anchor point z_i and the instance P_i of the same class while optimizing the loss for different class representations. During this process, each instance in the stack is alternately treated as an anchor, allowing comprehensive learning across the entire dataset.

In our approach, we combine cross-entropy loss and supervised contrastive (SupCon) loss by forming a weighted average. This merger aims to take advantage of two loss functions: cross-entropy loss for accurate classification and SupCon loss for learning discriminative representations. This combined approach ensures efficient optimization of the task at hand. Thus, the ultimate loss is represented as follows:

$$\mathscr{L}_{\text{SCL}} = (1-\lambda)\mathscr{L}_{CE} + \lambda\mathscr{L}_{\text{supcon}} \tag{4.6}$$

4.3.3 Contrastive Adversarial Training (CAT):

Adversarial training is a great way to demonstrate the resilience of a machine learning model, especially in the face of hostile, misleading data. In this study, we focus on Contrastive Adversarial Training (CAT), a technique that demonstrates how models such as BanglaBert, a pre-trained language model, handle such adversarial inputs.

The process begins by crafting adversarial examples designed to fool the model. These examples are generated by tweaking the word embedding matrix of BanglaBert. Creating effective adversarial examples requires meticulous adjustments to ensure they effectively challenge the model's predictive capabilities.

Once crafted, these modified input sequences are processed by what we call the perturbed encoder (e^{V+r}). Here, a small perturbation r is applied to the encoded input IDs, resulting in an output denoted as z_j, which serves as the positive pair to another representation z_i. To introduce nonlinearity, we use a ReLU projection layer.

The final representations z_i and z_j are crucial for calculating the NTXent loss. This loss function maximizes the difference between representations from different classes while minimizing the difference between an anchor (the original input representation) and adversarial instances from the same class. This strategy enhances the model's ability to discern between different classes and handle deceptive input data effectively.

The NTXent loss is computed using the following formula:

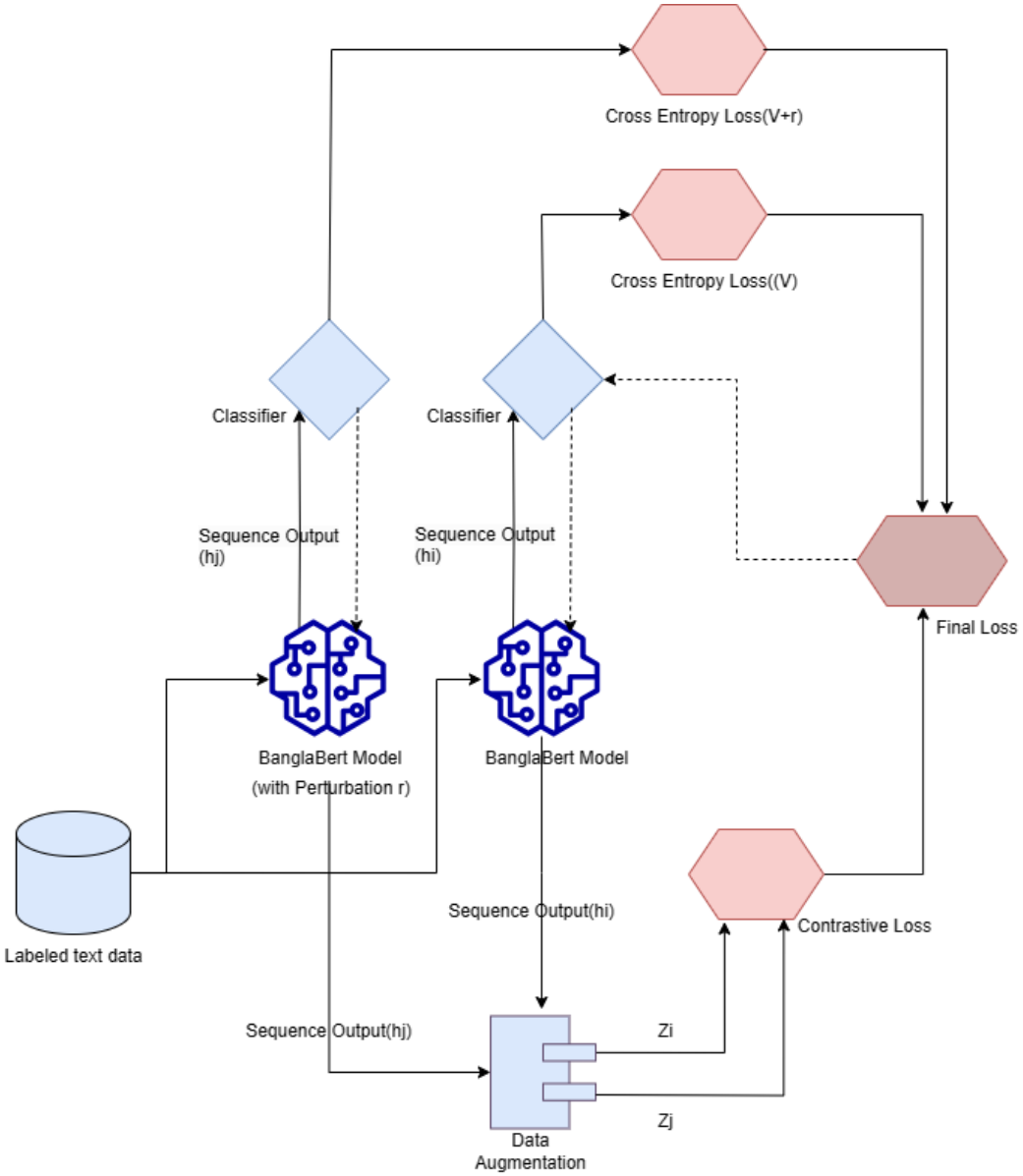

Figure 4.3: Model Architecture for CAT

$$\mathscr{L}_{\text{NTXent}} = \sum_{i=1}^{2N} \frac{-1}{P_i} \sum_{j \in P_i} \log \left(\frac{e^{\text{sim}(z_i, z_j)/\tau}}{\sum_{k=1}^{2N} \mathbb{1}_{i \neq k} e^{\text{sim}}(z_i, z_k)/\tau} \right) \tag{4.7}$$

To incorporate adversarial training into the overall process, we take a weighted average of the Cross-Entropy Loss ($\mathscr{L}_{CE}$) and the NTXent loss ($\mathscr{L}_{NTXent}$). This weighted combination results in the final loss function for Contrastive Adversarial Training ($\mathscr{L}_{CAT}$), given by:

$$\mathscr{L}_{\text{CAT}} = \frac{1-\lambda}{2}(\mathscr{L}_{CE} + \mathscr{L}_{CE}^{\text{V+r}}) + \lambda \mathscr{L}_{NTXent} \tag{4.8}$$

Here, the weight parameter λ controls the balance between the two losses. This formulation ensures that the model maintains accuracy in its predictions while also being robust

against adversarial attacks.

4.3.4 Label-Aware Contrastive Loss (LCL):

In [12], a contrast loss known as LCL is introduced. This loss function considers the link between labels. For applications requiring fine-grained classification, this loss is helpful.

Fine-grained classification is the classification task of distinguishing closely related classes. The differences between these categories are very small. For example, there might be a sentiment analysis task in NLP that involves distinguishing between closely related categories, namely ”positive” and ”very positive”. Both classes have examples with almost the same meaning. In this case, the task of fine-grained classification is to classify highly mixed emotion pairs compared to classifying “positive” and “negative” class samples.

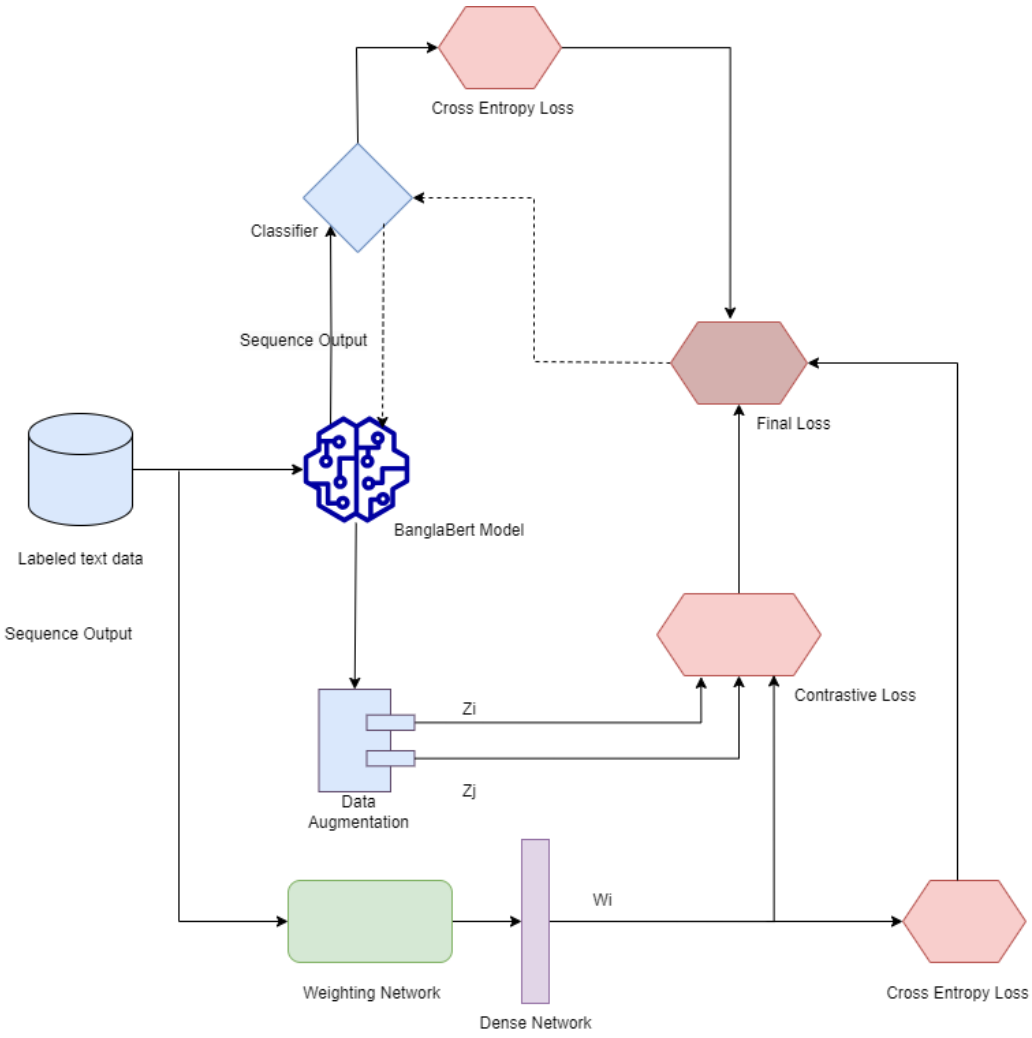

Figure 4.4: Model Architecture for LCL

According to [12] When tackling fine-grained classification issues, employ a dual-model method. This technique adds a weighted network in addition to the primary encoder.

Discovering the connections between labels is the job of weighted networks. A weighted network is responsible for boosting the obfuscated label "X"'s weight in relation to other "X" labels. We use three losses to optimize the encoder and weighted network. Cross-entropy loss Lw is used to optimize the weighted network, and a linear combination of Lf and the cross-entropy loss L_e is used to optimize the encoder network's output.

The prediction probabilities obtained from the softmax layer of the weighting network help to compute the probability of a sample belonging to a class c,

$$w_{i,c} = \frac{e^{z_{i,c}}}{\sum_{k=1}^{C} e^{z_{i,k}}} \tag{4.9}$$

Here, $w_{i,c}$ where C is the number of classes. $w_{i,c}$ denotes the probability of sample x_i belonging to class c.

The weights obtained from weighting are incorporated with NTXent loss to train the model.

$$\mathscr{L}_i = \sum_{j \in P_i} \log \left(\frac{w_{i,y_i} \cdot e^{\mathrm{sim}(z_i, z_j)/T}}{\sum_{k=1}^{2N} l_{i \neq k} w_{i,y_k} \cdot e^{\mathrm{sim}(z_i, z_k)/T}} \right) \tag{4.10}$$

$$\mathscr{L}_f = \sum_{i=1}^{2N} \frac{-L_i}{P_i} \tag{4.11}$$

Here , w_{i,y_k} denotes the relationship between a label y_k and an input sample x_i.

In our work, we use dropout-based data augmentation. The dropout-based data augmentation is used to produce an equivalent representation z_j, for the representation z_i. Both the representations z_i and z_j belong to the same class. For a confusable sample, the weighting network will assign higher weights to classes that are closely related to the sample. This high weighting score is used with the NTXent loss which helps the model to find more distinguishing patterns to differentiate between confusable samples. The final loss is,

$$\mathscr{L}_{LCL} = (1-\lambda)(\mathscr{L}_E + \mathscr{L}_w) + \lambda \mathscr{L}_f \tag{4.12}$$

4.3.5 Token-level Adversarial Contrastive Training (TACT):

Token-Level Adversarial Contrastive Training is a novel technique published in [15], which aims to improve the resilience and discriminative capacities of neural network models in natural language processing (NLP) applications. Tokenization, the process of breaking up input text data into discrete units like words or subwords, is the first step in the

TACT process. In the neural network architecture, each token is first represented as a vector. Traditionally, NLP approaches incorporate perturbations in the model's input by perturbing complete word embedding matrices. Token representations are immediately adjusted by TACT, which adopts an alternative strategy. This makes it possible to precisely alter the representation of every token, which makes it easier to gather complex contextual data.

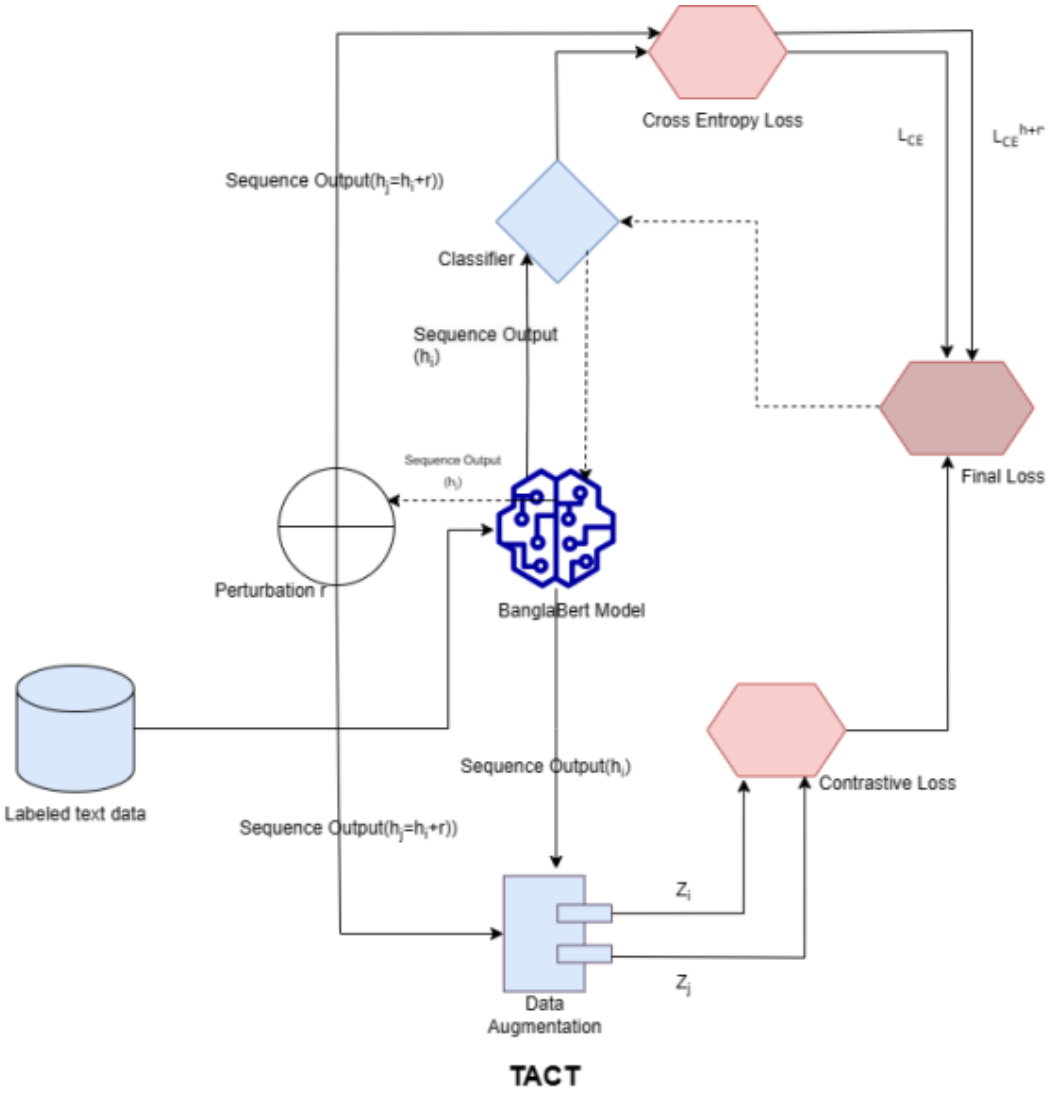

Figure 4.5: Model Architecture for TACT

TACT minimizes a composite loss function in order to optimize the model parameters during training. This loss function combines contrastive features, like InfoNCE loss, to promote the learning of unique token representations, with supervised components, like cross-entropy loss for precise predictions. TACT's primary novelty is the way it perturbs token representations directly. To do this, gradients of the loss function with respect to each token representation must be computed, and the representations must then be modified correspondingly. Mathematically, the perturbation r for a token representation

h_i is calculated as:

$$r = -\epsilon \frac{\nabla h_i L(f(x_i, y_i))}{\|\nabla h_i L(f(x_i, y_i))\|_2} \tag{4.13}$$

Where is a small perturbation factor, $\nabla h_i L(f(x_i, y_i))$ represents the gradient of the loss function with respect to h_i, and $\|.\|_2$ denotes the L2 norm. The perturbed token representation h(i) is obtained by adding this perturbation to the original representation:

$$h_j = h_i + r \tag{4.14}$$

Similar to traditional methods, both the original token representation h_i and the perturbed representation h_j are passed through a non-linear projection layer. This layer helps the model capture complex relationships between tokens. The obtained representations are then used to train the model to minimize the InfoNCE (Info Normalized Cross-Entropy) loss, denoted as $\mathscr{L}_{InfoNCE}$.

The final loss function for this variant of CAT denoted as $\mathscr{L}_{TACT}$, combines the traditional cross-entropy loss $\mathscr{L}_{CE}$ and the $\mathscr{L}_{InfoNCE}$ loss, with a balancing parameter λ:

$$\mathscr{L}_{TACT} = \frac{1-\lambda}{2} \cdot (\mathscr{L}_{CE} + \mathscr{L}_{CE}^{h+r}) + \lambda \cdot \mathscr{L}_{InfoNCE} \tag{4.15}$$

This approach allows the model to learn discriminative token representations directly, offering more precise adjustments that capture contextual nuances effectively. By integrating adversarial training principles at the token representation level, TACT presents a promising avenue for enhancing the robustness and generalization capabilities of NLP models.

4.4 Model Initialization

4.4.1 Attention Mechanism

Introduced by Vaswani et al. in 2017 [16], the Transformer model is a significant advance in deep learning, especially for natural language processing (NLP) problems. The Transformer breaks from sequential processing techniques by using a self-attention mechanism that enables it to continuously record contextual dependencies throughout the whole input sequence. The Transformer is capable of learning complex patterns and long-range dependencies in data with efficiency thanks to its parallelizable architecture and multi-head attention mechanism. This breakthrough has significantly impacted a number of fields outside of natural language processing (NLP), providing the basis for cutting-edge models in computer vision, machine translation, and reinforcement learning. The Trans-

former is a key component of modern deep-learning architectures due to its scalability and versatility.

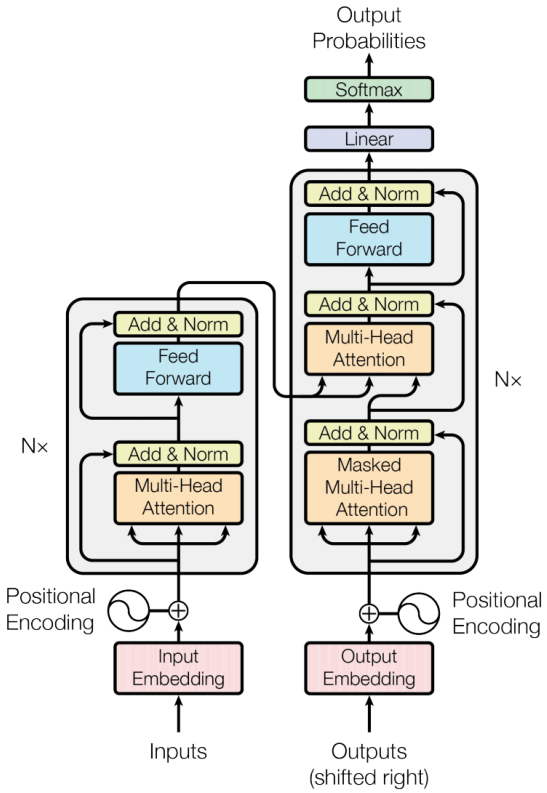

Figure 4.6: Attention is all you need [16]

4.5 BERT(Bidirectional Encoder Representation of Transformers)

The groundbreaking natural language processing (NLP) paradigm BERT (Bidirectional Encoder Representations from Transformers) has greatly improved the state of the art in a variety of language-related tasks. BERT is a transformer architecture with attention mechanisms that was introduced by Devlin et al. in 2018 [11] and enables bidirectional contextual information gathering.

In contrast to earlier models, BERT is pre-trained using masked language modeling on huge corpora, which allows it to acquire highly contextualized word representations. Pre-training gives BERT a deep comprehension of the relationships and subtleties in language, which makes it very useful for tasks that come after, like text classification, named entity recognition, and question answering.

For the required word embeddings for our work, the two Bangla bert models used are

Bangla-bert-base (sagor sarkar), and Bangla-bert(cse buet nlp). The BERT model takes the sentences as input and breaks the sentences into tokens. Each word of the sentence is a token. These tokens are assigned tokenID according to the vocabulary used during pretraining. For the sentences, BERT uses the [CLS] and [SEP] tokens to indicate the beginning and end of a sentence. The output of the [CLS] token for each sentence is used for predicting the class of the sentence as BERT uses the [CLS] token to represent the entire sentence for sentence classification. After processing each sentence, the output representation of the [CLS] token, is passed through a softmax layer that produces probability for the sentence belonging to each class. The class with the highest probability is chosen as the class label of the input sentence. In our work, for maximum sequence 128 has been used, and as we used so here the dimension is 768 and the number of encoders is 12.

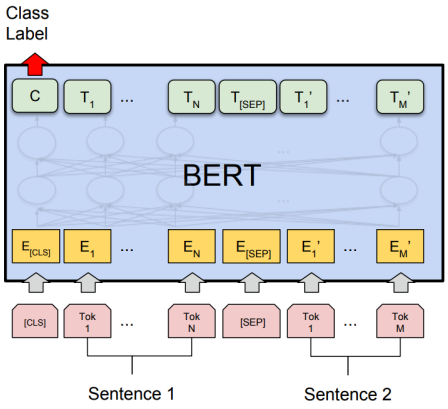

Figure 4.7: BERT pre-trained model
[16]

[CLS]"আমার পড়া সবচেয়ে বাজে বই"[SEP] having label negative is given as input to the BERT models.

The BERT model tokenizes the input sentence and assigns the tokenID based on the vocabulary used during pretraining. Then from the Bert pre-trained model get contextualized representations(hidden states) for each token in the sentence. After the processing of the given sentence, the BERT models give the output of the representation of the [CLS] token that is used to represent the entire sentence. This representation is input to a softmax layer, from where we obtain the class label of the sentence. The class having the maximum probability is assigned as the class label of the input sentence.

Chapter 5

Dataset

In our research, we've examined four distinct datasets to understand their impact on the performance of Contrastive Learning (CL)-based methods. These datasets include Bengali Hate Speech (Binary) (BHSB) [1], Rokomari Book Review (RBR) [17], Bengali Hate Speech (Multiclass) (BHSM) [18], and Daraz Product Review (DPR). Each dataset varies in size and content.

The rationale behind utilizing datasets of different sizes is to gauge how CL-based methods fare across a spectrum of data volumes. By including datasets of varying sizes, we aim to assess the scalability and robustness of CL-based models. Additionally, we've worked with both multiclass and binary class datasets to explore the versatility of the models we're investigating.

By systematically organizing our datasets and employing a consistent evaluation framework, we aim to gain insights into the effectiveness of CL-based methods across different data sizes and classification scenarios. This structured approach enables us to draw meaningful conclusions regarding the applicability and performance of CL-based models in real-world scenarios.

- Bengali Hate Speech(Binary) (BHSB): [1]
  The Bengali Hate Speech dataset is one of the largest datasets in Bengali which comprises 30k posts which are collected from YouTube and Facebook pages. Among the 30k posts, 10k are hateful and 20k are normal. This dataset is heavily biased toward not hate speech. The author employed 50 annotators and instructed them with proper guidelines to build the dataset. To validate the annotation quality, the author randomly sampled 300 posts for each annotator, manually reannotated them, and found that 91.05% annotation was correct. Among all the datasets we used, BHS is the largest one.

[1] Bengali Hate Speech Dataset

Table 5.1: Statistics about the dataset:

Class	Values	Example
Hate	10K	"যত সব পাগল ছাগল"
Not Hate	20K	"ভাল মানুষদের ভাল থাকা উচিৎ!"
Total	30K	

- Rokomari Book Review (RBR): [2]
  The Rokomari Book Review consists of 1,445 instances among which 972 belong to the positive label class and 473 belong to belong to the negative class. This dataset is the smallest among the used datasets. All the collected reviews were manually annotated by two native Bengali speakers. The dataset consists of two columns. The first column is the Review column which is filled with Bengali book reviews and the second column is the Sentiment column which contains the corresponding labels of the reviews (Positive (1) or Negative (0)).

Table 5.2: Statistics about the dataset:

Class	Values	Example
Positive	972	"আসলে কিন্তু অসাধারন একটি বই!!!!!"
Negative	473	"আমার পড়া সবচেয়ে বাজে বই"
Total	1445	

- Bengali Hate Speech(Multiclass) (BHSM): [3]
  From the Bengali Hate Speech Dataset [18], we utilized 5,698 labeled examples. This dataset categorizes observations into various types of hate speech, including political(814), personal(2189), geopolitical(1738), and religious(957) categories. The distribution of samples for each type of hate speech are provided in the Table below.

Table 5.3: Hate Speech Data Distribution and Examples

Class	Values	Example
Political	814	"ডঃ কামাল হোসেন ভারতের দালাল হয়ে বিএনপিকে ধ্বংস করতে এসেছে ।"
Personal	2189	"ফাপর দিয়া নিজেকে ওস্তাদ ভাবা লোক ইনি"
Geopolitical	1738	"তারপরও নরেন্দ্র মোদী মহাভারতে সিংহ পরুষ।"
Religious	957	"প্লিজ সকল মুসলিম ভাইদের বলছি এই সমস্ত মালাউন শয়তান থেকে সাবধান"
Total	5698	

- Daraz Product Review(DPR): [4]
  We employed manual data collection methods as well as web scraping to gather our

[2] Rokomari Book Review Dataset
[3] Bengali Hate Speech (Multiclass)
[4] Daraz Product Review (DPR)

corpus of data. From the web scraped portion, we eliminated any input sequences other than Bengali, such as English sentences. The reviews in our dataset, Daraz Product Review, are classified into three labels: Negative (1613), Positive (3052), and Neutral (649). With two annotators participating in the annotation process, Cohen's kappa statistic was used to assess the inter-annotator agreement. The result was a Cohen's kappa value of 0.65 and a percentage of agreement of 97.26%.

To illustrate how annotators assigned the positive, negative, and neutral classes, here are some examples from our custom dataset:

Positive: **"প্যাকিজিং সুন্দর ছিলো। অফারে কম টাকায় দেওয়ার জন্য ধন্যবাদ ইনফিনিক্স।"** (Translation: "Packaging was beautiful. Thanks to Infinix for offering at a lower price.")
Here, the annotators identified positive sentiment in the sentence based on the presence of words conveying appreciation, satisfaction, or praise. For example, in the provided sentence, phrases like **"সুন্দর ছিলো"** (beautiful) and **"ধন্যবাদ"** (thanks) indicate a positive sentiment towards the packaging and the offer provided by Infinix.

Neutral: **"ভালই চলছে। তবে বেশি ঘন কফি হলে লোড নিতে পারেনা।"** (Translation: "It's going well. However, it may not be able to handle heavy coffee.")
Here, the Annotators categorized sentences as neutral when they did not express strong positive or negative sentiment. Neutral sentences often contain factual statements or expressions without significant emotional content. In the provided sentence, phrases like **"চলছে"** (going well) and **"বেশি ঘন কফি হলে লোড নিতে পারেনা"** (may not be able to handle heavy coffee) convey a neutral tone, indicating a factual description without strong emotional leaning.

Negative: **"ভাবছিলাম যে ভালো প্রডাক পাব কিন্তু পেলাম একটা ছেরা প্রডাক্ট"** (Translation: "I thought I would get a good product, but I received a low-quality product.")
Here, the Annotators recognized negative sentiment by identifying words conveying disappointment, dissatisfaction, or criticism. In the given sentence, the phrase **"ছেরা প্রডাক্ট"** (low-quality product) indicates a negative sentiment towards the received product, suggesting disappointment or dissatisfaction with its quality.

Table 5.4: Daraz product Review Data Distribution and Examples

Class	Values	Example
Negative	1613	"খুব বাজে একটা পন্য"
Positive	3052	"আলহামদুলিল্লাহ ভালো প্রডাক্ট এবং ভালো ভাবে পেয়েছি।"
Neutral	649	"খারাাপ না"
Total	5314	

The figure 5.1 shows the distribution of the amount of data for the four datasets. It is visible from the figure that the datasets vary in size, which is useful for examining

Table 5.5: Inter-Annotator Agreement Results

Percentage of Agreement	97.26%
Cohen's Kappa (κ)	0.65

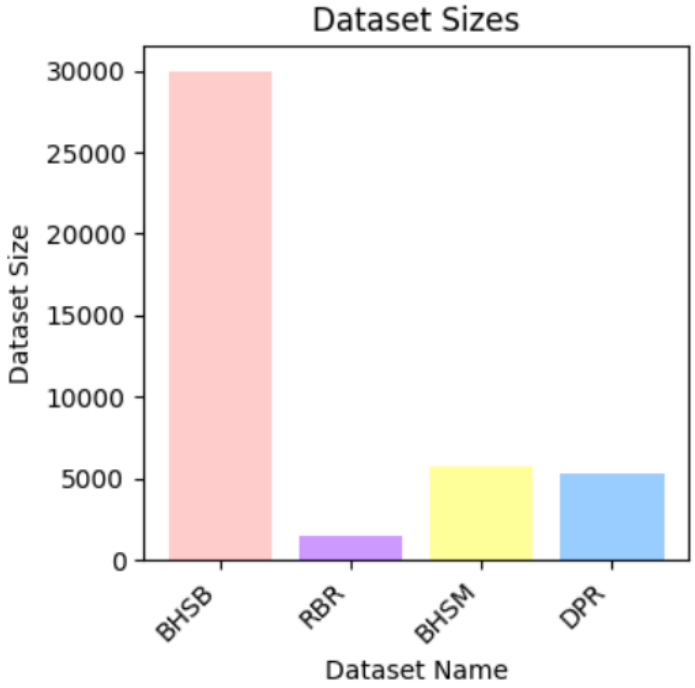

Figure 5.1: Dataset Size

the capability of the method's performance when the amount of data varies.

5.1 Dataset Distribution

To ensure a comprehensive evaluation, we've partitioned each dataset into train, validation, and test sets, maintaining a ratio of 70:15:15, respectively. This partitioning scheme allows us to train our models on a substantial portion of the data, validate their performance on a smaller subset, and finally, assess their generalization capabilities on unseen data. For our custom dataset (DPR) we have collected 5314 datas in out corpus but we have used 4004 datas. The distribution of the datasets across these sets is outlined in TABLE 5.6.

Table 5.6: Distribution of Data for Different Dataset Used in Experiment

Dataset	Total	Train(70%)	Test(15%)	Evaluation(15%)
BHSB	30,000	21,000	4,500	4,500
RBR	1,445	1,011	217	217
BHSM	5698	3988	855	855
DPR	4004	2802	601	601

Chapter 6

Experimental Setup and Result Analysis

6.1 Experimental Setup

6.1.1 Epoch

The epoch is simply the counter denoting how many times through the entire training set has been passed forward and backward through the neural network; as we varied the number of epochs from 10 to 20 to 30, we hoped to better understand how different lengths of training time would impact the trained model's final performance.

6.1.2 Learning Rate

One important hyperparameter that controls the size of the step taken during optimization is the learning rate. To determine how different learning rates might affect the training process and the accuracy of the final model, we conducted experiments. To find an ideal rate that promotes efficient learning without causing the optimization process to diverge or converge too soon, we experimented with learning rates of 0.0005, 0.00005, and 0.00001.

6.1.3 Early Stop

Early stopping is a regularization strategy that keeps an eye on the model's performance on a validation dataset while it is being trained in order to prevent overfitting. Five epochs of patience were used in our tests to implement early stopping. This indicates that in order to avoid overfitting and guarantee ideal model generalization, the training process was stopped if the model's performance on the validation dataset did not improve for five consecutive epochs.

6.1.4 Transfer Learning

Applying the knowledge that pre-trained models have learned on big datasets to a particular task is known as transfer learning. Two pre-trained Bangla Bert models—BanglaBert (csebuetnlp/banglabert) and Bangla-BERT-Base (sagorsarker/bangla-bert-base)—were refined in our study using transfer learning. This strategy enables us to gain from the text representations we've learned in a more broad way, which could increase the efficacy and efficiency of our contrastive learning techniques.

6.1.5 Pretrained Models

Pretrained models offer a foundation for further refinement on particular tasks. Two pre-trained Bangla Bert models were used in our investigation: Bangla-BERT-Base (sagorsarker/bangla-bert-base) and BanglaBert (csebuetnlp/banglabert). These models were selected in order to assess how well they work with contrastive teaching techniques. In order to potentially improve model performance and convergence, we sought to apply the information obtained from large-scale datasets to our work by utilizing pre-trained models.

6.1.6 Batch Size

The number of training samples handled in a single forward and backward pass during each training iteration is referred to as the batch size. To find out how varied batch sizes affected model convergence and training efficiency, we conducted experiments. We tested batch sizes of 16 and 32 in an effort to strike a compromise between the accuracy of parameter updates and computing performance. Larger batch sizes offer a more stable estimate but demand more memory and processing power, whereas smaller batch sizes often provide a noisy gradient estimate but may allow faster convergence.

6.1.7 Optimizer

The efficacy and efficiency of the training process are strongly impacted by the optimizer selection. We used the AdamW optimizer [19], which combines weight decay regularization with Adam's adjustable learning rate capabilities. AdamW incorporates weight decay to reduce the chance of overfitting while assisting in the effective update of model parameters. Our goal in utilizing AdamW was to increase the training process's stability and speed of convergence, which would improve the performance of the model. Additionally, a weight decay of 0.01 was applied to further regularize the model and prevent overfitting.

6.1.8 Dropout

In order to minimize overfitting in neural networks, dropout is a regularization strategy that involves randomly deactivating a portion of the neurons during training. 10% of neurons were arbitrarily dropped out of the training set during each training iteration when we used dropout with a rate of 0.1 in our trials. This method keeps individual neurons from being too dependent on certain properties in the input data, which helps the network acquire more resilient and generalizable representations.

6.1.9 Split Size

We separated the dataset into training, validation, and testing subsets to assess the model's performance consistently and guarantee its capacity for generalization. Seventy-one percent of the data were in the training subset, and fifteen percent were in each of the validation and testing subsets. Using separate datasets for testing and validation, this partitioning technique helps avoid overfitting by training the model on a variety of data sets. We sought to determine the model's generalizability to unseen cases and real-world settings by testing it on untested data.

6.2 Experimentation Results and Results Analysis

6.2.1 Binary Class

We have used 2 distinct Bengali sentiment analysis datasets and implemented different methods involving different losses(CE, SCL, CAT, TACT, LCL). Methods involving SCL ,CAT, TACT, LCL give better accuracy than the method using just CE as depicted in the table of performances given Table 6.1.

Table 6.1: Results of the different models using CE, SCL, CAT, TACT, LCL for binary class datasets BHS, RBR

Pretrained Model	Dataset	Batch Size	Learning Rate	Epoch	F1 Score				
					CE	SCL	CAT	TACT	LCL
BuetNLP	BHSB	16	0.0005	10	0.67	0.44	0.42	0.47	0.40
			0.00005	10	0.67	**0.93**	**0.93**	0.92	0.92
			0.00001	30	0.90	0.92	0.93	**0.93**	0.91
		32	0.0005	30	0.67	0.50	0.50	0.50	0.48
			0.00005	10	0.67	0.93	0.92	0.93	0.93
			0.00001	20	**0.90**	0.93	0.92	0.93	**0.92**
	RBR	16	0.0005	10	0.66	0.94	0.88	0.86	0.80
			0.00005	10	0.66	0.95	0.95	**0.98**	0.96
			0.00001	10	0.90	0.95	0.95	0.92	0.92
		32	0.0005	10	0.40	0.44	0.44	0.89	0.44
			0.00005	30	0.82	**0.96**	**0.97**	0.94	0.89
			0.00001	20	**0.94**	0.95	0.95	0.94	**0.97**
SagorSarkar	BHSB	16	0.0005	10	0.81	0.92	0.92	0.44	0.40
			0.00005	10	0.73	0.91	0.92	0.93	0.93
			0.00001	30	0.75	0.91	0.92	0.93	0.88
		32	0.0005	30	0.40	0.44	0.50	0.50	0.50
			0.00005	10	0.73	0.92	0.92	0.92	0.90
			0.00001	20	0.73	0.92	0.92	0.92	0.90
	RBR	16	0.0005	10	0.82	0.44	0.44	0.44	0.40
			0.00005	10	0.83	0.86	0.92	0.88	0.84
			0.00001	10	0.75	0.89	0.89	0.87	0.87
		32	0.0005	10	0.83	0.40	0.44	0.33	0.40
			0.00005	30	0.83	0.84	0.86	0.88	0.87
			0.00001	20	0.83	0.90	0.90	0.89	0.89

CE, SCL, CAT, TACT and LCL perform better than CE for both the Bert models(csebuetnlp-BanglaBert, sagorsorkar-BanglaBert). These losses performed better as we used contrastive learning for our formulation. For the Rokomari book review dataset, we get 94% accuracy for CE, SCL outperforms CE by 2% and CAT outperforms CE by 3%, and similarly, TACT and LCL outperformed CE by 4% and 3%. And if we consider the Bengali hate speech binary our contrastive learning losses SCL, CAT, TACT, and LCL outperformed CE by 3%, 3%, 3%, and 2%. Between these two datasets we got, binary book review gave us the best results for all the losses.

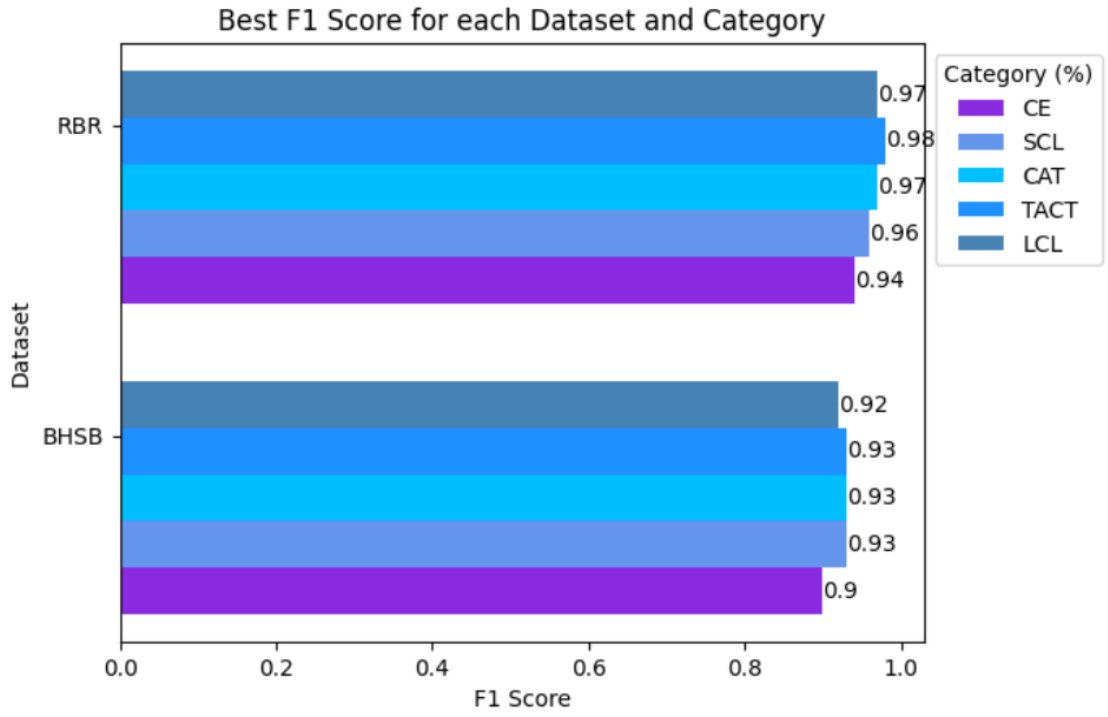

Figure 6.1: The Best obtained result for each method for Binary class Datasets

If we compare CAT, LCL with SCL, in some cases CAT and LCL outperform SCL. CAT works better than SCL for its adversarial training process. Adversarial training makes the model more robust for working with unseen data. LCL has a weighting network and by utilizing it and adding extra weights it can distinguish classes that are closely related and have little variations between them, for example, positive and very positive classes will be classified correctly despite having overlapping meanings. Again if we compare TACT with CAT and LCL, TACT outperformed both CAT and LCL by 1% in some cases as TACT is a variant of CAT that uses token-wise perturbation instead of perturbing the whole embedding matrix so it preserves the context of each token, which may be same words but used in different sentences with different context. Here LCL and CAT gave the same score which is 97%. In some cases, SCL has outperformed CAT in predicting class with the utilization of adversarial training. For the Rokomari book review dataset, SCL has an F1-score of 96%, CAT has an accuracy of 97%, TACT has 98% and LCL has 97%. So, TACT outperforms SCL by 2%, CAT by 1%, and LCL by 1%. For the hate speech dataset CAT, SCL, and TACT give the same performance which is 93% but these losses outperformed LCL by 1%.

For the Rokomari book review task as well as Bengali hate speech, TACT outperformed all other losses. though the performance of CL methods is close to each other outperforming the traditional method (CE) again proves the minimax nature of CL (minimizing the difference between the same classes and maximizing between different classes), the fundamental and most important factor of our formulation is undoubtedly effective.

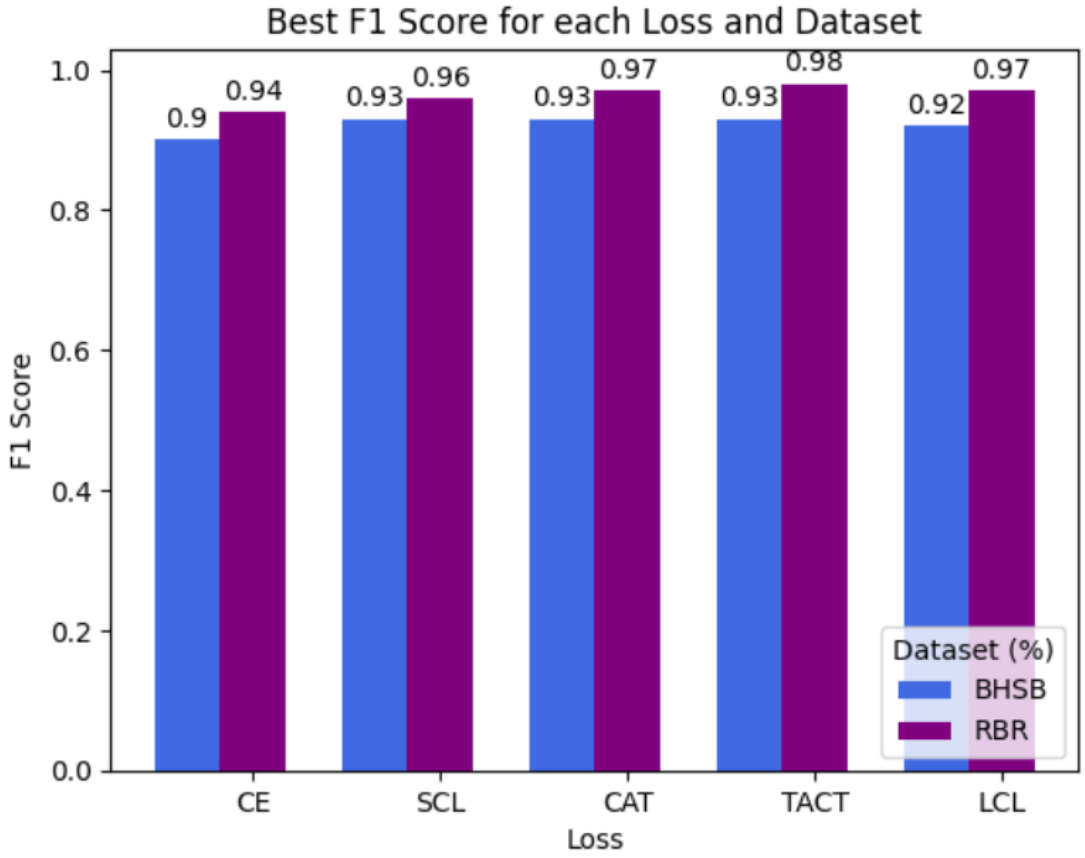

Figure 6.2: The Comparison study of different losses for the Binary class Datasets

From the figure 6.2 it can be seen that for the Binary class datasets, the dataset RBR got the better result for for all the losses. Also for the dataset BHAB, TACT beats all other methods. In summary, the performance of the CL method beats the CE method.

6.2.2 Multiclass

We have used two distinct Bengali sentiment analysis datasets for multiclass classification and implemented different methods involving different losses(CE, SCL, CAT, TACT, LCL). Methods involving SCL, CAT, TACT, and LCL give better accuracy than the method using just CE as depicted in the table of performances given Table 6.2.

CE, SCL, CAT, TACT, and LCL perform better than CE for both the Bert models(csebuetnlp-BanglaBert, sagorsorkar-BanglaBert) for the two multiclass datasets. These losses performed better as we used contrastive learning for our formulation.
For the BHSM dataset, we get 88% accuracy for CE, SCL outperforms CE by 2% and CAT outperforms CE by 2% and similarly, TACT and LCL outperformed CE by 3% and 2%. The figure 6.3 shows that for the dataset BHSM, all the CL methods outperformed the CE method. Among all the CL methods, TACT performed best with an F1 score of

91%.

And if we consider the dataset DPR our contrastive learning losses SCL, CAT, TACT, and LCL outperformed CE by 1%, 2%, 4%, and 1%. Among all the methods used, the highest F1 score was obtained for TACT. From the figure 6.3, for the dataset DPR, the supremacy of CL methods over the CE method is again visible. The best-performing CL method is TACT with an F1 score of 77%. Between these 2 datasets we got, binary book review gave us the best results for all the losses.

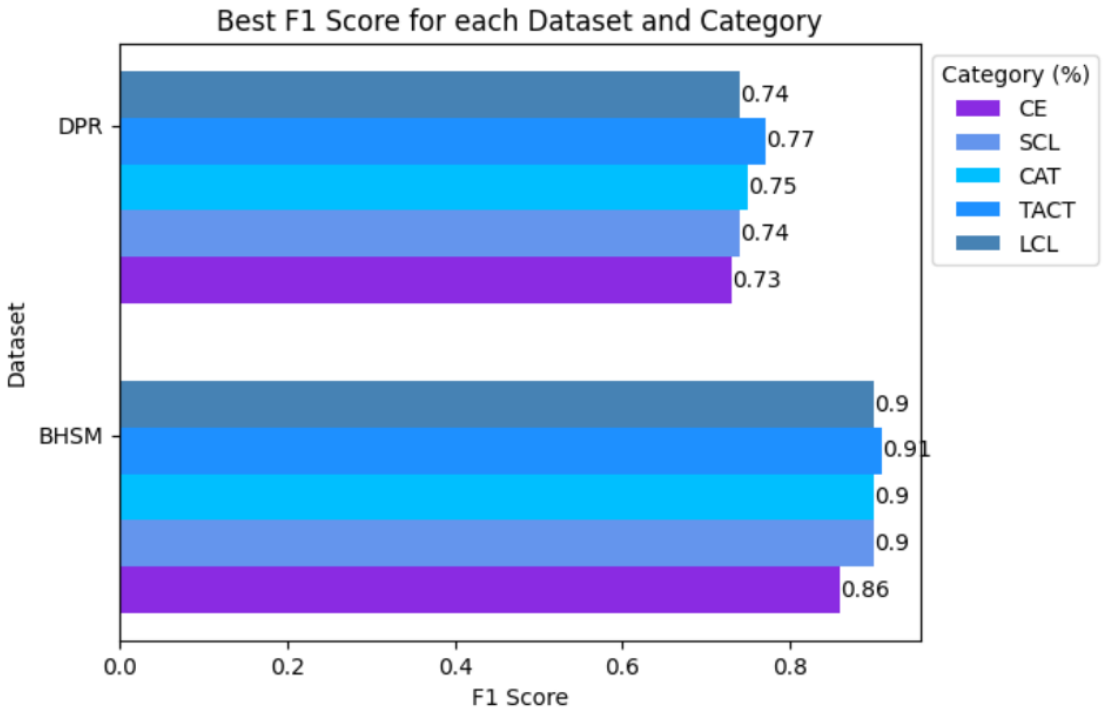

Figure 6.3: The Best obtained result for each method for Multiclass Datasets

Between the two multiclass datasets we used, BHSM gave us the best results for all the losses.

If we compare CAT with SCL, in some cases CAT outperforms SCL. CAT performs better than SCL in its adversarial training process. Adversarial training makes the model more robust for working with unseen data. In CAT the adversarial examples are created by purpose to make the model resistant when it sees unseen data in the real world. For example for the dataset DPR, we get 74% for SCL and CAT beats this by 1%. It shows the advantage of adversarial training.

If we compare LCL with SCL, in some cases LCL outperforms SCL. LCL uses a weighting network. The weighting network gives higher weights to classes that are closely related. Also, the weighting network helps to reduce errors in classes of nominal categories by the help of weighting network. As a result of this, LCL outperforms SCL. It is visible from

Table 6.2: Results of the different models using CE, SCL, CAT, TACT, LCL for Multi-class datasets BHSM, DPR

Pretrained Model	Dataset	Batch Size	Learning Rate	Epoch	F1 Score				
					CE	SCL	CAT	TACT	LCL
BuetNLP	BHSM	16	0.0005	10	0.39	0.44	0.40	0.40	0.40
			0.00005	30	0.84	**0.90**	**0.90**	**0.91**	**0.90**
			0.00001	10	0.84	0.86	0.86	0.85	0.85
		32	0.0005	10	0.40	0.43	0.40	0.44	0.40
			0.00005	10	**0.88**	0.89	0.88	0.89	0.87
			0.00001	20	0.87	0.86	0.87	0.88	0.87
	DPR	16	0.0005	10	0.44	0.50	0.50	0.54	0.50
			0.00005	10	0.72	0.72	**0.75**	**0.77**	**0.74**
			0.00001	10	**0.73**	0.73	0.73	0.73	0.73
		32	0.0005	10	0.40	0.54	0.54	0.54	0.54
			0.00005	10	0.70	0.73	0.73	0.73	0.73
			0.00001	20	0.70	**0.74**	0.73	0.72	0.71
SagorSarkar	BHSM	16	0.0005	10	0.39	0.40	0.40	0.39	0.40
			0.00005	30	0.80	0.83	0.82	0.87	0.82
			0.00001	10	0.80	0.87	0.85	0.85	0.84
		32	0.0005	10	0.40	0.42	0.43	0.32	0.40
			0.00005	10	0.86	0.87	0.86	0.87	0.86
			0.00001	20	0.85	0.86	0.85	0.86	0.85
	DPR	16	0.0005	10	0.39	0.40	0.40	0.40	0.39
			0.00005	10	0.72	0.72	0.74	0.73	0.73
			0.00001	10	0.73	0.72	0.73	0.73	0.73
		32	0.0005	10	0.40	0.44	0.44	0.40	0.40
			0.00005	10	0.73	0.72	0.74	0.74	0.74
			0.00001	20	0.70	0.72	0.73	0.71	0.71

the table that in some cases LCL outperforms SCL.

For the comparison between CAT, and TACT, it can be seen that TACT outperformed both CAT and LCL. TACT is a variant of CAT. The difference between TACT from CAT is that instead of perturbing the embedding matrix like CAT, TACT perturbs the tokens by keeping the context in consideration. The same word may have different meanings based on the use of the word in sentences and this context is preserved by TACT. As TACT preserves more contextual information than CAT, the performance of the TACT improves beating the performance of CAT. For example, for the dataset BHSM the F1 score of TACT is 91% which is 1% higher than the CAT. Also for the dataset DPR the TACT leaves behind CAT by 2%.

For the DPR and BHSM, TACT outperformed all other losses. Though the performance of CL methods is close to each other outperforming the traditional method (CE) again proves the minimax nature of CL (minimizing the difference between the same classes and maximizing between different classes), the fundamental and most important factor of our

formulation is undoubtedly effective.

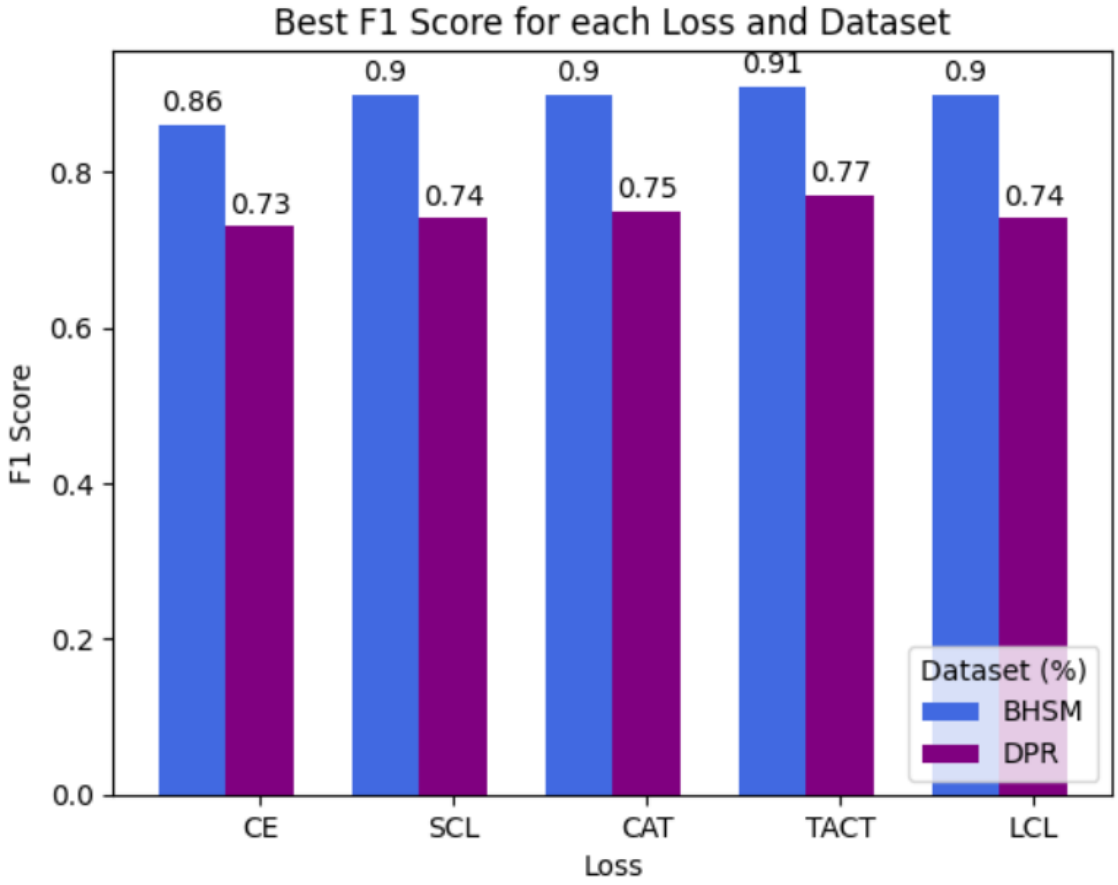

Figure 6.4: The Comparison Study of Different Models for the Multiclass Datasets

From the figure 6.4 it can be seen that for the Multiclass datasets, the dataset BHSM got the better result for all the losses. Also for the dataset DPR, TACT beats all other methods. In summary, The performance of the CL method is the beats CE method.

Again for all the datasets, for the four CL methods models, we get better performance for csebuetnlp-BanglaBert model as it is an ELECTRA discriminator model, and sagorsorkar-BanglaBert and ELECTRA replace the MLM of BERT with Replaced Token Detection (RTD), which looks to be more efficient and produces better results.
Another hyperparameter like learning rate, 0.00005 and 0.00001 performs the best and almost the same, and 0.0005 performs the worst as a smaller learning rate has a smaller step size so the minimum value of a parameter can be determined precisely and a larger learning rate, having larger step size does not perform better. In the case of epoch, we get higher performance for the higher number of the epochs, among 10,20 and 30, 30 and 10 perform the best.

Chapter 7

Project management

7.1 Project Schedule

A Gantt chart is a crucial research management tool, providing a visual representation of the thesis schedule, aiding in efficient planning, resource allocation, realistic deadlines, and identifying dependencies. It enhances team communication, identifies bottlenecks, and maintains project momentum. Here, for the given Gantt chart 2 horizontal cells combined represent one month and the months start from May 2023.

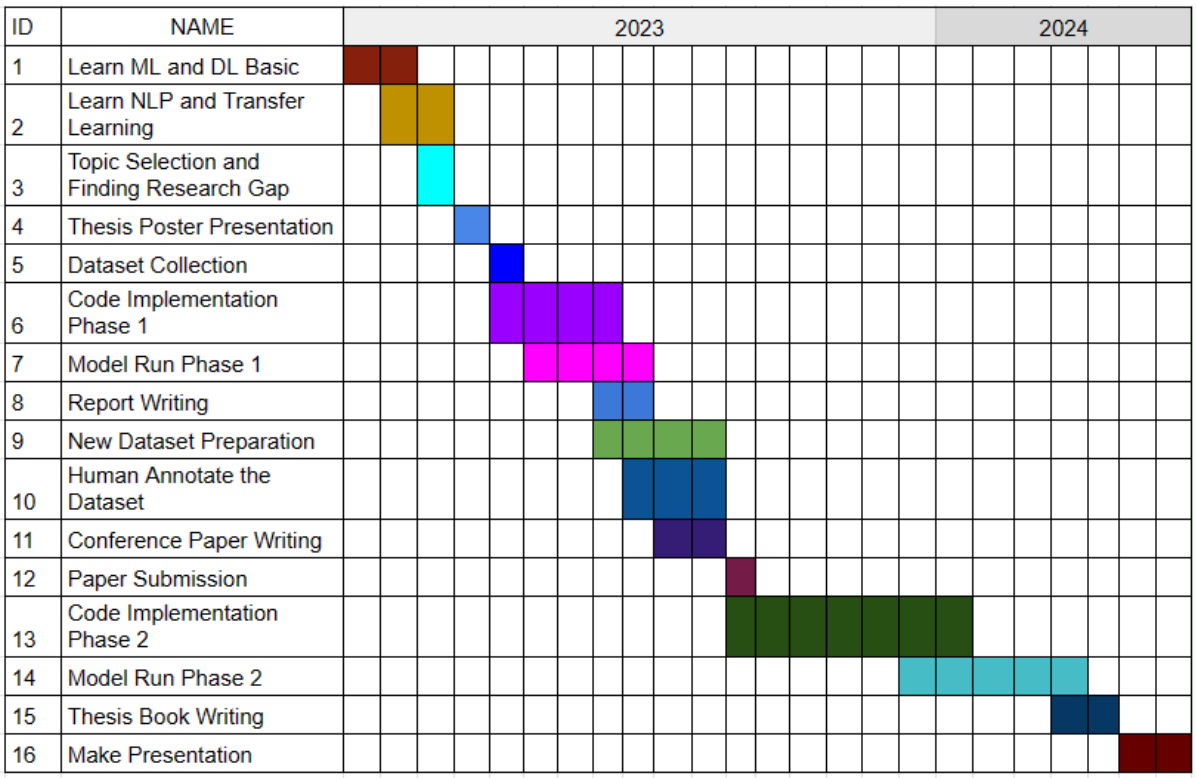

Figure 7.1: Gantt Chart

7.2 Cost Analysis

The cost analysis for our research project encompasses several key components. The dataset creation and annotation incur expenses amounting to 5000 tk. Additionally, the utilization of GPU, an essential computational resource, accounts for 4000 tk. Printing costs contribute 1000 tk to the overall expenditure, while miscellaneous expenses amount to 1000 tk, and also we used Kaggle premium for one month which cost us 4800 tk. Consequently, the total budget for the research project is 15800 tk.

Table 7.1: Cost Analysis

No.	Expense Type	Cost (Tk)
1	Dataset Creation	3000
2	Dataset Human Annotation	2000
3	GPU Fan	4000
4	Printing	1000
5	Miscellaneous	1000
6	Kaggle Premium	4800
Total		**15800**

Chapter 8

Conclusion

8.1 Overview

This research makes use of supervised Contrastive Learning (CL) methods, specifically utilizing sagorsarkar-bert and buetnlp-banglabert implementations. A comparison of the predictions shows that CL methods outperform cross-entropy, underscoring their superior performance in the context of the study's NLP tasks.

8.2 Limitations

The limitations of our research are given below:

- One of the limitations of our proposed methods is that it depends on the learning rate. For larger learning rates, the performance of the CL-based method degrades.
- The CL-based methods tend to depend on the no of epochs used to train the model. This is also a limitation of the proposed method.
- Again, when the batch size used to train the method varies, the result of the proposed method also varies which is an addition to the limitation of the proposed method.
- The proposed method works better for ELECTRA-based pretrained model but not for BERT-based pretrained models which is a limitation in terms of generalization.

8.3 Future work

We're aiming to make our Bangla language models smarter by teaching them to understand both text and images together. This means we'll explore how words and pictures can work together to solve problems like describing images with text, answering questions about what's in a picture, and even understanding emotions expressed in both text and images. We also want to make our models smarter with less training data. We'll do this by starting with models that have already been trained on a lot of general knowledge and then fine-tuning them to understand Bangla language tasks better. This should help our models learn faster and perform better, even when there isn't much Bangla-specific data available. Another thing we're planning to tackle is how our models can adapt to different situations. Imagine if a model trained on news articles could also understand medical texts without needing to be trained all over again. We want to develop techniques to help our models adapt to new tasks and domains more easily. Additionally, we're working on creating better ways to test how well our models are performing. This involves building datasets that accurately reflect the kinds of challenges our models will face in the real world. By doing this, we can ensure that our models are not only accurate but also useful for practical applications in fields like education, healthcare, and more. Overall, our goal is to make Bangla language technology more accessible and effective by improving how our models understand language and images together, learn from less data, adapt to different tasks, and are evaluated for their performance.

References

[1] N. Romim, M. Ahmed, H. Talukder, and M. Saiful Islam, "Hate speech detection in the bengali language: A dataset and its baseline evaluation," in *Proceedings of International Joint Conference on Advances in Computational Intelligence* (M. S. Uddin and J. C. Bansal, eds.), (Singapore), pp. 457–468, Springer Singapore, 2021.

[2] C. Zhang, M. Abdul-Mageed, and G. Jawahar, "Contrastive learning of sociopragmatic meaning in social media," 2023.

[3] D. Shome and T. Kar, "Conoffense: Multi-modal multitask contrastive learning for offensive content identification," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 4524–4529, 2021.

[4] L. Pan, C.-W. Hang, A. Sil, and S. Potdar, "Improved text classification via contrastive adversarial training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 11130–11138, 2022.

[5] P. I. Khan, S. A. Siddiqui, I. Razzak, A. Dengel, and S. Ahmed, "Improving health mentioning classification of tweets using contrastive adversarial training," *arXiv preprint arXiv:2203.01895*, 2022.

[6] Y. Kim, S. Park, and Y.-S. Han, "Generalizable implicit hate speech detection using contrastive learning," in *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 6667–6679, 2022.

[7] S. C. Raia Hadsell and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," *In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 1735–1742. IEEE.*, 2006.

[8] T. Shi, L. Li, P. Wang, and C. K. Reddy, "A simple and effective self-supervised contrastive learning framework for aspect detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 13815–13824, 2021.

[9] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18661–18673, 2020.

[10] J. S. . C. S. Ian J. Goodfellow, "Explaining and harnessing adversarial examples," *arXiv:1412.6572v3 [stat.ML] 20 Mar 2015*, 2015.

[11] K. L. K. T. CJacob Devlin, Ming-Wei Chang, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of NAACL-HLT 2019*, 2019.

[12] V. Suresh and D. C. Ong, "Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification," *CoRR*, vol. abs/2109.05427, 2021.

[13] Q. V. L. C. D. M. Kevin Clark, Minh-Thang Luong, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv:2003.10555v1 [cs.CL] 23 Mar 2020*, 2020.

[14] Researchgate, "The image is taken from researchgate. website: https://www.researchgate.net/figure/Electra-based-pretrained-component_fig2_362990426,"

[15] L. Li and X. Qiu, "Tavat: Token-aware virtual adversarial training for language understanding, 2020," *URL https://arxiv. org/abs/2004*, vol. 14543, pp. 118–168, 2004.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[17] A. D. Gobinda Chandra Sarker, Kazi Md Sadat, "Book review sentiment classification in bangla using deep learning and transformer model," *4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2022.

[18] M. R. Karim, B. R. Chakravarti, J. P. McCrae, and M. Cochez, "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, 2020.

[19] J. L. B. Diederik P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on October 3, 2024 at 2:40pm.