BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

BDA: Bangla Text Data Augmentation Framework

Md. Tariquzzaman
190041101
Audwit Nafi Anam
190041120
Naimul Haque
190041214

Department of Computer Science and Engineering
Islamic University of Technology
June, 2024

Declaration of Authorship

This is to certify that the work presented in this thesis titled **"BDA: Bengali Text Data Augmentation Framework"** is the outcome of the analysis and experiments carried out by Md. Tariquzzaman, Audwit Nafi Anam, and Naimul Haque under the supervision of Md. Mohsinul Kabir, Assistant Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

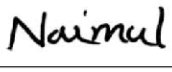
Authors:

Md. Tariquzzaman

Student ID - **190041101**

Audwit Nafi Anam

Student ID - **190041120**



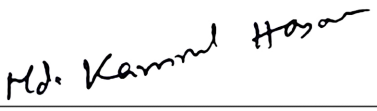
Naimul Haque

Student ID - **190041214**

Supervisors:

Md. Mohsinul Kabir
Assistant Professor
Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Dr. Hasan Mahmud
Associate Professor
Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)



Dr. Md. Kamrul Hasan
Professor
Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Acknowledgements

We extend our deepest gratitude to **Md. Mohsinul Kabir** for his unwavering support and guidance throughout this journey. His inspiration has been the driving force behind this work. His dedication has been a cornerstone of our progress, providing us with invaluable insights, especially when we were directionless. His rigorous review of our work and his expertise were instrumental in developing some of our key ideas. We are truly indebted to him for his significant contributions to this thesis.

We also wish to express our heartfelt thanks to **Dr. Hasan Mahmud** for his valuable suggestions and feedback, which have been essential in shaping the core aspects of our work. His input helped us build a solid foundation for our proposal. Additionally, we are grateful to **Dr. Kamrul Hasan** for his mentorship. His expertise played a pivotal role in guiding us through the proper execution of our research.

We convey our gratefulness to Allah Subhanahu Wa ta'ala and our parents for everything.

This work is dedicated to our parents.

Abstract

Data augmentation can be a valuable technique, particularly in resource-scarce linguistic domains for improving the performance of natural language processing tasks by creating new synthetic data instances. This paper introduces a Bangla text Data Augmentation Framework (BDA) using pre-trained model-based and rule-based approaches, along with a filtering pipeline to ensure semantic similarity and lexical variance between augmented and original text. We provide a comprehensive pipeline for the proposed framework and perform an in-depth analysis of how well it performs in the Bangla text classification tasks. Our framework improved the F1 score of classification tasks by up to 13.92%, 8.58%, and 10.55% among 15%, 50%, and 100% clipping ranges respectively, across five different datasets. Training with BDA while using only 50% of the available training set achieved the comparable F1 score as normal training with all available data. We provide an extensive study of the performance of each augmentation approach at the clipping ranges of datasets using BanglaBERT and variants of SVM. Furthermore, we discuss the indicators for optimal performance of the BDA framework and its shortcomings with in-depth analysis.

Keywords — Augmentation, NLP, Synthetic text generation, Data scarcity.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Bangla is the seventh most spoken language[1] in the world, and there's a growing need for tools that can understand and process it. However, unlike English, which has many resources for constructing these Natural Language Processing (NLP) tools, Bangla has very few. This lack of resources hinders the progression of Bangla NLP. The existing datasets we have are mostly of small sizes as explored by Kabir et al. [1] and are prone to low lexical variance among the samples. While developing large high-quality datasets is the optimal solution, it is not always practical as it requires an extensive amount of annotation. Thus there is a need for an efficient alternative that can help us with this issue. Text data augmentation offers an effective solution to improving the performance of NLP models as shown by Dhole et al. [2] which artificially expands training datasets and generates additional examples coherent to the original dataset, to capture the nuances of the language. This approach is particularly crucial for Bangla, as it enables us to overcome the limitations imposed by the low availability of data and improve the generalization capabilities and robustness of NLP models.

Although there have been recent works to develop NLP tools for Bangla Sen et al. [3], it still lacks a properly evaluated framework for Bangla text augmentation that systematically leverages multiple augmentation approaches, to maximize the diversity and quality of augmented texts. There have been studies such as by Mohiuddin et al. [4]

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

which focus only on back-translation and tools such as bnaug[2], banglanlptoolkit[3] for augmenting Bangla text. However, they lack in-depth analysis and evaluation of the effects of augmentation in actual datasets and filtering augmented text. Since all these existing solutions rely on a limited set of techniques, there is a pressing need to develop an effective augmentation framework for Bangla.

We introduce BDA[4], a framework for Bangla text augmentation, which aims to address these challenges by incorporating a combination of model-based and rule-based approaches Feng et al. [5] for generating a wide array of synthetic samples of semantically equivalent variations. along with a filtering mechanism to remove low-quality augmented texts. By addressing this critical gap, we believe the BDA framework can significantly advance the field of Bangla NLP, enabling the development of more accurate, robust, and versatile models in resource-scarce settings. Our contributions to the BDA framework are as follows:

1. A comprehensive Bangla text augmentation pipeline containing rule-based and transformer-based augmentation methods to cover the widest possible array of augmented texts, along with a filtering process to ensure high-quality augmented texts.

2. An in-depth analysis of how well augmentation works in the Bangla language, what are the individual contributions of each method of augmentation, and why we need an array of augmentation methods instead of a single approach.

3. A detailed explanation of where Bangla text augmentation works best and where it is not suitable.

[2] https://github.com/sagorbrur/bnaug
[3] https://pypi.org/project/banglanlptoolkit
[4] https://github.com/tzf101/BDA-Bangla-Text-Data-Augmentation

1.1 Research Challenges

The progress in Natural Language Processing (NLP) techniques has significantly impacted the creation of machine learning models that can comprehend and produce human language. Nevertheless, the effectiveness of these models largely depends on the availability and quality of training data. In languages such as Bangla, which have fewer digital text resources than English, the lack of comprehensive annotated datasets is a major challenge. To address this, data augmentation methods are used to artificially increase the training dataset by generating altered versions of the existing data.

Adapting these techniques to Bangla text poses unique challenges because of its intricate script and rich linguistic characteristics. Bangla, as an Indo-Aryan language, exhibits intricate morphological structures, a variety of syntactic constructions, and significant contextual nuances. Therefore, developing a framework for Bangla Text Data Augmentation (BDA) requires addressing specific issues that ensure the augmented data is both diverse and representative of the natural language use.

Ensuring Variability while Preserving Semantic Integrity

The overarching challenge in text data augmentation for Bangla lies in balancing linguistic variability with the preservation of semantic integrity. This challenge manifests in several key areas:

1. **Diversity and Complexity:** Bangla's rich dialectical variations and linguistic intricacies necessitate a sophisticated approach to text augmentation. Techniques must not only recognize and adapt to these variations but also enrich the text without distorting its original meaning. These augmented texts also are required to be diverse in nature for models to benefit from it.

2. **Authenticity vs Diversity:** Striking the right balance between authenticity and linguistic diversity is crucial. Augmented texts should align to the original texts in contextual sense while having different structure to introduce diversity.

3. **Semantic Integrity:** Ensuring that augmented texts maintain their original sentiments, themes, and informational content is paramount. This is especially critical in domains like sentiment analysis, where the emotional tone must remain unaltered.

Careful Consideration of Augmented Texts

Effective augmentation requires careful consideration of the techniques used and their impact on the data:

1. **Context Handling:** Augmentation techniques must ensure that the generated text variations adhere to linguistic norms and maintain logical coherence. This involves sophisticated handling of context dependencies and semantic relationships within sentences. Sentence alteration should be applied while avoiding changes in the text's intended meaning.

2. **Scalability and Robustness:** As datasets' size and models' complexity increase, augmentation techniques' scalability becomes crucial. The BDA Framework must be capable of handling large volumes of data efficiently without compromising the quality of the augmentation. This will require us to maintain a balance between speed and quality. Moreover, the robustness of these techniques is essential to ensure consistent performance across different sets of data and varying linguistic scenarios. This includes the ability to handle diverse linguistic phenomena such as idiomatic expressions, compound sentences, and regional variations within Bangla. Ensuring robustness also means that the augmentation processes should be resilient to errors in the original data, such as typographical errors or inconsistencies in formatting, which are common in less-resourced languages.

Optimizing Parameters for Augmentation Methods

Properly tuning the parameters of augmentation methods is critical for maximizing their effectiveness without compromising the linguistic integrity of the data:

1. **Parameter Tuning:** Each augmentation method includes various adjustable parameters, such as the degree of synonym replacement or the extent of sentence rearrangement. Tuning these parameters involves a careful balance to enhance data variety while preserving the original semantics.

2. **Experimentation and Validation:** Optimizing these parameters for Bangla text requires extensive experimentation and subsequent validation. This process ensures that the modifications made by augmentation do not distort Bangla's natural linguistic characteristics, maintaining the augmented data's authenticity and usability.

These challenges frame the scope of this research and underline the complexity and necessity of developing a robust framework for Bangla Text Data Augmentation. By addressing these issues,

1.2 Motivation

Although there is a significant need for Natural Language Processing (NLP) in Bangla, the current environment lacks a comprehensive augmentation framework with effective text augmentation methods. The performance of NLP models heavily relies on the availability and diversity of textual data. Unfortunately, Bangla is still vastly underrepresented in digital resources, highlighting the urgent need to develop such tools.

1. **Limited Resources:**

    - Bangla, like many other languages, suffers from a lack of extensive and diverse datasets Feng et al. [6] compared to languages like English.
    - Data augmentation techniques can address this problem by increasing the effective size of the dataset without requiring manual annotation. This is crucial for training robust and accurate natural language processing (NLP) models.

2. **Variations in Language Usage:**

- Bangla encompasses a variety of dialects, regional differences, and variations in spelling and grammar. These inconsistencies can pose significant challenges for NLP models in accurately understanding and processing text Tareq et al. [7].
- Data augmentation can introduce these linguistic variations into the training data, exposing models to different forms of the language. This helps models become more versatile and capable of handling diverse linguistic patterns.

3. **Cost of Annotating Large Datasets:**

   - Annotating a large volume of training data in Bangla is both costly and time-consuming.
   - Data augmentation offers a cost-effective solution by creating synthetic training examples, which reduces the need for extensive manual annotation and enhances model performance.

4. **Overcoming Bias:**

   - Bias is a frequent issue in machine learning, where a model becomes too specialized in the training data, resulting in poor generalization to new, unseen data Keya et al. [8].
   - Augmentation adds diversity to the training data, preventing the model from memorizing specific instances and promoting the learning of more generalizable patterns, thereby reducing the risk of overfitting Bhowmik et al. [9].

5. **Improving Model Robustness:**

   - Models trained on homogeneous datasets can struggle with real-world text data that exhibits syntactic and lexical variations.
   - Augmentation enriches the training data with diverse linguistic forms, idiomatic expressions, and grammatical structures. This makes models more robust, particularly for tasks like sentiment analysis, machine translation, and text classification, where understanding context and nuances is crucial.

1.3 Problem Formulation

Modifying a Bangla text instance T, which is a sequence of word tokens $W = \{w_1, \ldots, w_n\}$, in order to produce an augmented version T', using a set of augmentation techniques $A = \{a_1, a_2, \ldots, a_k\}$.These techniques should alter the lexical or syntactic structure or both of T to create T', but still preserve the original meaning and context. Each technique must conform to and accommodate the unique grammatical and contextual aspects of the Bangla language.

The main objective is to enhance the robustness, reduce bias, and improve the generalization of machine learning models by artificially increasing the size and diversity of training datasets, especially when the available data is limited, imbalanced, or lacks variety. Developing Bangla NLP models faces substantial challenges due to the scarcity of robust training data, which is further compounded by several critical factors:

1. **High-Quality Data in Large Quantity:** Obtaining extensive, high-quality datasets for Bangla, a language with limited resources, is a complex endeavor.

2. **Diversity and Unpredictability of Real-World Data:** Existing datasets may not adequately represent the diverse and unpredictable nature of real-world language usage.

3. **Class Imbalance:** Certain linguistic patterns are disproportionately represented, leading to skewed model training.

1.4 Objectives

Our approach focuses on the following objectives:

1. **Generation of Synthetic Text:** To generate synthetic text data that closely mimics real-world Bangla language usage in accordance with the given dataset.

2. **Enhancement of Model Robustness for Noisy Real-World Data:** To strengthen the model's ability to process and interpret noisy, real-world data effectively.

3. **Synthetic Sample Creation:** To create synthetic samples that help balance the class distribution in training data.

1.5 Approach

Our strategy includes implementing data augmentation techniques like `Back Translation (BT)`, `Paraphrasing(PP)` and rule-based methods such as `Synonym Replacement (SR)` and `Random Swapping (RS)`. To combat the issues of data diversity, unpredictability, and class imbalance, we introduce the concept of `Augmentation Threshold (AT)`. This threshold ensures that augmented texts maintain a balance between syntactic and semantic similarity to the original texts, enhancing data quality and quantity while respecting privacy concerns and ensuring diverse language representation.

1.6 Thesis Layout

The thesis is thoughtfully structured, adhering to a well-crafted layout that encompasses the following 6 chapters — In Chapter-1, we set the context for the reader by introducing the research problem and delineating the challenges encountered in this domain of scholarly inquiry. We formulate the research problem in precise terms, list down the contributions of this thesis work, and write about the circumstances that inspired us to pursue the topic in this chapter. This thesis layout section is a constituent of the chapter that aids in the readability of the thesis book by recapitulating a cohesive narrative of the work. In Chapter-2, we critically examine the existing scholarly works, theories, and empirical studies related to the research topic, establishing a strong theoretical foundation for the study. We provide a proper taxonomy and chronological development of the models used in the existing literature in this chapter and perform comparative analyses among them. In Chapter-3, we outline our proposed methodology with the aid of intuitive diagrams and explain the underlying mechanisms of the components that are present within the architecture. Chapter-**??** elaborates on the experimental results of our proposed methodology. We provide a detailed statistical analysis of our datasets and their performance, with and without our pipeline. We examine the results from multiple perspectives and uncover insights. Chapter-5 presents an in-depth analysis and interpretation of the obtained results along with a critical discussion of their implications and relevance. We also bring to light some of the weaknesses of the existing models and provide a comprehensive ablation study to convey a deeper understanding of the pipeline's working process to the reader in this chapter. The epilogue of our thesis work, Chapter-7, provides a concise summary of our research findings and insights into potential future directions for further exploration towards the apotheosis of this research domain.

Chapter 2

Literature Review

2.1 Data Augmentation

Although data augmentation (DA) has been a widely applied field in computer vision, such as cropping, flipping, rotation, etc. of images Simard et al. [10], it was introduced in the Natural Language Processing (NLP) field fairly recently. Data augmentation helps in training more robust models and increasing their performance by adding diversity and variance to existing data using modifications to existing data or generating synthetic data. There are many well-received and general purpose augmentation techniques explored in computer vision (CV). But in the field of NLP such methods are yet unexplored or still not solidified, especially for resource-scarce languages like Bangla. This is due to the complications presented by the discrete and complex nature of natural language. This inhibits the efficacy of DA methods like swapping words, inserting words, synonym replacement, etc. For example not every word can be replaced with generic words (articles, prepositions, conjunctions, common nouns, etc.) such as the, this, that, a, an, the, etc. Not every word in the text may have synonym, in case of which the meaning and context of the text will change. The language model will fail to classify such ambiguous text. Another problem is that there is no unified valid opinion that any specific augmentation technique will perform well for all datasets of all types and sizes. Rather effective augmentation techniques must be found out through trial and error by performing experiments Feng et al. [6].

Despite these challenges, there is a surge in interest and popularity in the NLP research community for identifying effective data augmentation (DA) techniques, evaluating their efficacy, and determining suitable evaluation metrics. This surge in interest is fueled by the widespread availability of large pre-trained models, leading to the emergence of more tasks and domains. Numerous researchers have experimented with and proposed a variety of DA techniques. In Natural Language Processing (NLP) field, where the input space consists of distinct words or symbols, creating helpful augmented examples that maintain the desired invariances (things that shouldn't change) is a more complex challenge. Surveys such as Feng et al. [6] and Li et al. [11] provide an overview of the DA field in NLP and highlight the most prominent hurdles and challenges to inspire and drive interest in this area. The typical DA methods used in NLP can be generally divided into three main categories: rule-based, example interpolation-based, and model-based techniques Feng et al. [6].

2.2 Goals and Trade-offs

The primary aim of data augmentation (DA) is to generate additional training samples without the necessity of extensive data collection, which can be both expensive and time-consuming. By artificially enlarging the training dataset, DA enhances the generalization capabilities of machine learning models, especially when working with limited or imbalanced datasets. For DA to be effective, it should meet two main criteria: ease of implementation and enhancement of model performance. However, balancing these criteria often requires making trade-offs.

2.2.1 Quality and Quantity Tradeoff

Simple, rule-based techniques are straightforward to implement but may not always yield high-quality augmentations, as noted by Wei and Zou [12], Wei et al. [13], and Li et al. [14]. Conversely, model-based techniques can generate high-quality data with greater variation, resulting in better performance improvements; however, they are more complex and resource-intensive. While these techniques can be tailored for specific

downstream tasks to significantly enhance performance, they are challenging to develop and apply.

2.2.2 Lexical and Semantic Similarity Trade-off

Moreover, a trade-off exists between selecting lexical and semantic similarity for the original and augmented text. Our goal is to create augmented text that strikes a balance between lexical dissimilarity (diversity) and semantic similarity (fidelity). If the augmented sample is too similar, it may lead to overfitting, causing poor performance on the test set. Conversely, if it is too different from the original example, the model's performance might decline because it would be trained on examples that don't accurately reflect the target domain. Kashefi and Hwa [15] present an unsupervised method to preemptively choose among data augmentation strategies, instead of the typical "try everything" approach, which can be inefficient and expensive.

2.3 Techniques & Methods

Researchers have proposed numerous techniques to generate augmented samples with varying quality and diversity. Here, we discuss some methodologically representative and typical data augmentation (DA) techniques that are relevant to various tasks due to to their adaptable formulation.

2.3.1 Rule-based Techniques

Rule-based DA approaches use simple, predefined manipulations to create new data samples. These techniques operate directly on the input data, making the augmented data easy to interpret and use.

- **Easy Data Augmentation (EDA)**: Introduced by Wei and Zou [12], EDA involves token-level perturbations including random insertion, deletion, and swapping of words. These operations have improved performance across various text classification tasks by enhancing the robustness of models against input variations.

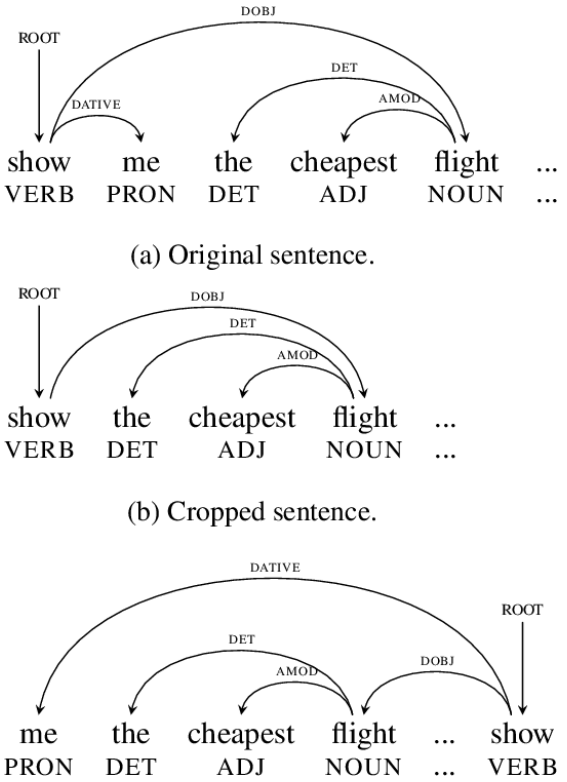
- **Feature Space Analogies**: Leveraging transformations between known class examples to generate data for novel classes, as discussed by Hariharan and Girshick [16] and Schwartz et al. [17], this method enriches training data effectively.
- **Iterative Affine Transformations**: As described by Paschali et al. [18], this technique uses affine transformations and projections to explore the class feature space, uncovering new patterns beneficial for model training.
- **Unsupervised Data Augmentation (UDA)**: Proposed by Xie et al. [19], UDA adapts supervised DA techniques for unsupervised learning through consistency training on pairs of original and augmented data.
- **Paraphrase Identification and Sentence Pair Augmentation**: Using a signed graph approach, Chen et al. [20] infer augmented sentence pairs, enriching the dataset with nuanced paraphrastic variations.
- **Dependency Tree Morphing**: Drawing from image manipulation techniques, this approach employs operations like swapping or removing nodes in dependency trees to enhance linguistic structure, which is especially beneficial for languages with complex case marking systems. For sentences annotated with dependencies, children belonging to the same parent are either exchanged (rotation) or some are removed (cropping), as presented by Şahin and Steedman [21].

2.3.2 Model-based Techniques

This section offers an overview of advanced model-based data augmentation methods, which utilize machine learning models to create new training data.

2.3.2.1 Backtranslation

The widely recognized Back-Translation technique, proposed by Sennrich et al. [22], entails translating a sequence into another language and subsequently translating it back to the original language to generate new data samples. This study builds upon the neural machine translation (NMT) architecture, which employs an encoder-decoder framework using recurrent neural networks (RNNs) with bidirectional gated recurrent units



(a) Original sentence.

(b) Cropped sentence.

Figure 2.1: Demonstration of the working of Dependency Tree Morphing

(GRUs). The encoder processes input sequences $x = (x_1, \ldots, x_m)$ into concatenated hidden states to form annotation vectors h_j, while the decoder predicts target sequences $y = (y_1, \ldots, y_n)$ using a context vector c_i derived from a weighted sum of these annotations. The alignment model, a single-layer feedforward network, estimates the alignment probabilities α_{ij}, essential for contextual translation accuracy.

The authors introduce an innovative approach to enhance NMT models using monolingual data through back-translation. This method involves generating synthetic source sentences from monolingual target texts, effectively creating additional, contextually enriched training data. The technique is mathematically grounded in improving the prior probability $p(T)$ of the target sentence, leveraging the sequence dependency conditioned by the encoder-decoder architecture. By integrating both synthetic and human-translated texts without freezing any network parameters, the approach maintains sensitivity to the source context, crucial for preserving the integrity of translation outputs. This dual strategy of combining real and synthetic data sets positions back-translation as a pivotal technique for optimizing NMT systems' performance by maximizing the

use of monolingual resources.

2.3.2.2 Seq2seq Models

Seq2Seq (Sequence-to-Sequence) models are a type of neural network architecture employed in NLP tasks such as machine translation, text summarization, and speech recognition. These models convert sequences from one domain (like text in one language) to sequences in another domain (like text in another language), making them suitable for tasks that require transforming input sequences into output sequences.

In their Diverse Paraphraser using Submodularity (DiPS) approach, Kumar et al. [23] leverage Seq2Seq models to generate augmented data for various classification tasks. DiPS optimizes a novel submodular objective function aimed specifically at paraphrasing. This method learns context-sensitive transformations and has demonstrated substantial effectiveness across tasks. Extensive experiments show that their method can generate structurally diverse paraphrases while maintaining fidelity.

Previous paraphrasing techniques often relied exclusively on selecting the top-k sequences from a beam search to incorporate diversity. However, this limited selection of best k sequences doesn't fully represent the possibilities within the search space, often leading to sentences that are very similar in structure, with only slight differences in punctuation or morphological forms. To address this, Kumar et al. [23]. introduced a new approach that combines a sentence encoder with a decoder designed to promote diversity.

They use a SEQ2SEQ framework for paraphrase generation, beginning with an encoder that processes tokenized source sentences. The decoder then generates paraphrases, trained using cross-entropy loss $\mathcal{L}(\text{generated}, \text{target})$. Their approach enhances standard decoding techniques by incorporating a submodular objective to improve paraphrase quality.

During the generation phase, following the initial encoding, the decoder iteratively selects $k < N$ from the N most probable subsequences at each time-step t, proceeding until the sequence reaches its target length T or an `<eos>` token.

2.3.2.3 Pretrained Language Models

- Augmentation can be done by replacing words with recurrent language model predictions, considering the current context.Kobayashi [24] introduced a method utilizing Recurrent Neural Networks (RNNs) to create augmented examples. This is achieved by stochastically replacing words in a sentence with other words that are predicted by a bi-directional LSTM-RNN at the same position in the sentence.

  The proposed model makes use of a bi-directional LSTM-RNN for estimating the contextual word probability in a cloze sentence $S\{w_i\}$, calculating $p(\cdot|S\{w_i\})$ for each position i. Words are encoded in both directions, and the concatenated outputs are input into a feed-forward network that outputs a vocabulary-based probability distribution.

  For contextual augmentation, they sample substitute words from $p(\cdot|S\{w_i\})$ using a temperature-controlled annealed distribution $p_\tau(\cdot|S\{w_i\}) \propto p(\cdot|S\{w_i\})^{1/\tau}$. This allows for diverse augmentation, from uniform sampling ($\tau \to \infty$) to deterministic high-probability word selection ($\tau \to 0$), enhancing model training by introducing variability at each update.

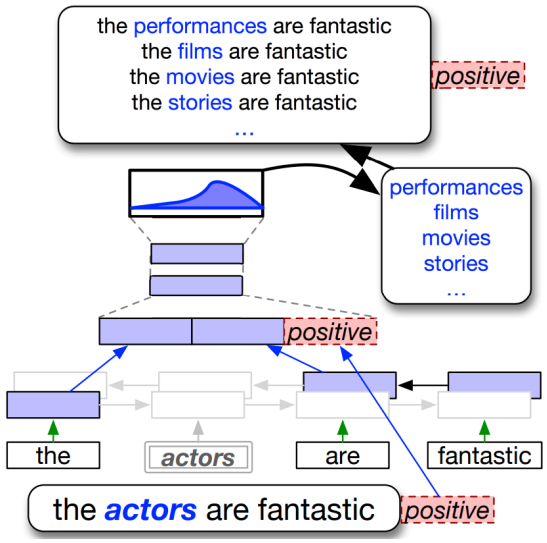


Figure 2.2: Contextual augmentation with a bidirectional RNN language model

- Yang et al. [25] propose Generative Data Augmentation for commonsense reasoning (GDAUGc), which uses pretrained transformer models to generate and then select the most information-rich and lexically diverse text for augmentation. To ensure that the most valuable examples are used for augmentation,they incorporated data selection methods based on influence functions (a way to measure the impact of each data point on model predictions) and a heuristic (a rule of thumb) designed to maximize the variety within the generated data pool. They also proposed an effective training scheme for augmentation with synthetic data consisting of two stages.

2.3.2.4 Soft Augmented Examples

In the paper by Gao et al. [26] suggest maintaining the complete data distribution by using ”soft” augmented examples, which have shown to be advantageous, especially in machine translation tasks. In contrast to traditional augmentation methods, their technique enhances a randomly selected word in a sentence with a contextual blend of several related words. In another work by Kobayashi [24], they recognize that although this approach preserves semantics through contextual relevance, it faces a significant limitation: to achieve new samples with sufficient variation, it requires multiple samplings. The challenge arises because replacing a word can result in a potentially vast number of candidates, given the extensive vocabulary of language models, making it nearly impossible to explore all options for optimal performance.

In the work done by Gao et al. [26], they propose improving Neural Machine Translation (NMT) training data by substituting a randomly selected word in a sentence with a ”soft word”, defined as a probability distribution over the vocabulary $|V|$. For a token t in V, the soft word for it is represented by $P(t) = (p_1(t), p_2(t), \ldots, p_{|V|}(t))$, where each $p_j(t) \geq 0$ and the sum of the probabilities equals one.

An embedding of soft word t is calculated as $e_t = P(t)E = \sum_{j=0}^{|V|} p_j(t)E_j$, where E is the embedding matrix for all words in $|V|$. This process employs a pre-trained language model to determine the probability of a word $P(t)$, based on all preceding words $x_{<t}$. Specifically for the t-th word x_t in a sentence, the probability $p_j(x_t) =$

$LM(w_j|x_{<t})$ indicates how likely the j-th word in the vocabulary is to appear in the sequence $x_1, x_2, \ldots, x_{t-1}$.

This distributional vector replaces the original word, providing a smooth approximation of the one-hot representation. During training, words in the training dataset are randomly selected with a probability γ and replaced with their soft versions.

2.3.3 Context-Sensitive Augmentation Techniques

Context-Sensitive Augmentation Techniques are methods in data augmentation that take into account the surrounding words or overall context when modifying or creating new data. This ensures the augmented data remains meaningful and relevant to the specific task or domain.

Enhancing Named Entity Recognition with Semantic Augmentation

Detecting and labeling proper entities within text, known as Named Entity Recognition (NER), plays an essential role in supporting subsequent tasks in natural language processing (NLP). Challenges in NER often include issues related to the scarcity of data. One effective method to mitigate this issue is through semantic augmentation, which has been particularly successful in enhancing NER for texts on social media, as described by Nie et al. [27]. This approach utilizes an attention-based, context-sensitive method to integrate semantic neighbors from an existing pretrained embedding space, significantly boosting the effectiveness of NER on social media platforms.

Self-Supervised Manifold-Based Data Augmentation (SSMBA)

Building on the concept of denoising autoencoders, Ng et al. [28] introduced the Self-Supervised Manifold Based Data Augmentation (SSMBA), a novel strategy for creating synthetic training samples. This technique employs a dual process of corruption and reconstruction to navigate a data manifold, leveraging BERT to manage domain variations effectively in test sets from diverse datasets. Here, the corruption function $q(x'|x)$ masks several word positions randomly, while the reconstruction function $r(x|x')$ restores them using BERT. Their approach demonstrates success in handling domain shifts across 9

datasets encompassing sentiment analysis, natural language inference, and neural machine translation tasks.

2.3.4 Semantic Text Exchange

In their work, Feng et al. [29] introduced a technique called Semantic Text Exchange (STE). This method modifies the meaning of a text to smoothly incorporate a new word or phrase, referred to as the *replacement entity (RE)*. This modification is executed through a mechanism referred to as **SMERTI**, which integrates **entity replacement (ER)**, **similarity masking (SM)**, and text infilling (TI) alongside a masked language model (LM) strategy. Although this technique was not originally designed for Data Augmentation (DA), it has since been effectively adapted for such applications, demonstrating its adaptability.

2.3.5 Generative Data Augmentation

Generative Data Augmentation involves creating new, synthetic data samples using machine learning models, typically generative models like Generative Adversarial Networks (GANs) or Variational Autoencoder (VAEs). These models learn the underlying patterns and distributions of the original data and generate similar but novel examples, increasing the diversity and size of the training dataset. This technique is particularly useful when data is limited or imbalanced, improving model performance and robustness.

Language-Model-Based Data Augmentation (LAMBADA)

The technique known as **language-model-based data augmentation (LAMBADA)** introduces a specialized approach to data augmentation by fine-tuning an advanced language model for specific tasks, as discussed in Anaby-Tavor et al. [30]. Initially, the model undergoes training with a limited amount of labeled data to tailor its capabilities. Once fine-tuned, the model, conditioned on a given class label, can produce new, relevant sentences for that class. These sentences are subsequently screened using a classifier developed from the original dataset. This method utilizes GPT-2, adapted

through training to generate examples specific to various classes. The most relevant examples are then chosen to enhance the dataset further. A comparative analysis of this method against other data augmentation techniques like Easy Data Augmentation (EDA), Conditional Variational Autoencoder (CVAE), and Conditional Bidirectional Encoder Representations from Transformers (CBERT) across various datasets and classifiers including SVM, LSTM, and BERT, reveals its statistical superiority over traditional data augmentation algorithms.

Label-Conditioned GPT-2 Model for Active Learning and Data Augmentation

In parallel, the research in Quteineh et al. [31] explores the use of a label-conditioned GPT-2 model within an active learning framework to validate the efficacy of this data augmentation strategy. By employing Monte Carlo Tree Search (MCTS) as the optimization technique and focusing on entropy to enhance the utility of the generated data, this method contrasts sharply with the Non-Guided Data Generation (NGDG) approach, which lacks a reward-based optimization.

2.3.6 Other Model-based Methods

Some other augmentation methods that use a pretrained model to generate augmented text are briefly discussed below:

- **Controlled Syntactic Paraphrasing**: Iyyer et al. Iyyer et al. [32] introduce syntactically controlled paraphrase networks (SCPNs). These networks are designed to generate paraphrases of a given sentence that conform to a specified syntactic structure, such as a constituency parse tree.
- **Story-level Paraphrasing**: As described in Gangal et al. [33], the Narrative Reordering (NAREOR) technique is applied to creatively alter the sequence of events in a story without changing its original storyline, thereby rewriting the narrative in a new order.
- **Enriching Misclassified Examples**: The study in Dreossi et al. [34] introduces a method for data augmentation that focuses on incorporating examples that were initially misclassified. This approach selectively enhances the training dataset with these new examples.

- **Labeling New Inputs Using BERT**: The method proposed in Thakur et al. [35] discusses the use of Augmented SBERT (AugSBERT), which employs a BERT cross-encoder for labeling new input pairs. These pairs are then added to the SBERT bi-encoder's training dataset, leading to marked improvements in tasks involving pair classification and regression.

2.4 Existing Work in Bangla

2.4.1 BanglaParaphrase

Akil et al. [36] introduced a high-quality paraphrasing dataset for Bangla, created synthetically and refined through a unique filtering process to guarantee both diversity and semantic similarity. Although not directly a work on data augmentation, the availability of a pre-trained language model fine-tuned on this dataset has proven beneficial for paraphrasing tasks, effectively serving as a data augmentation technique in this context. Here is a brief description on their dataset generation process and filtering pipeline.

Synthetic Data Generation

They generated a synthetic dataset by gathering high-quality Bangla sentences from an online source and translating them to English using a state-of-the-art translation model. For each English sentence, 5 new Bangla translations were generated through beam search. Pairs with a Language Agnostic BERT Sentence Embedding (LaBSE) [37] similarity score above 0.75 were chosen, resulting in a dataset of over 1.364 million sentences, each with multiple reference translations for the original source.

Filtering Pipeline

The different components of the novel filtering mechanism for dataset preparation and evaluation used by the authors are briefly discussed here:

- **PINC Score Filter**: Ensures diversity by measuring lexical dissimilarity using PINC (Paraphrase In N-gram Changes). The optimal threshold maximizes the PINC score with over 63.16% yield.

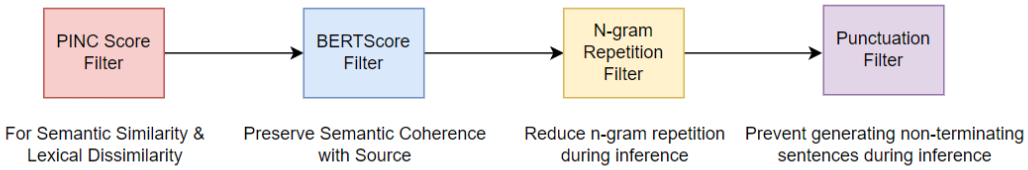


Figure 2.3: BanglaParaphrase Filtering Pipeline

- **BERTScore Filter**: Ensures semantic similarity using BERTScore with BanglaBERT embeddings. The threshold is set at a BERTScore ≥ 0.92 based on human evaluation of 300 samples.
- **N-gram Repetition Filter**: Addresses unnecessary length and repetition by checking for N-gram repeats ($N = 2$) in the target dataset.
- **Punctuation Filter**: Removes sentences without terminating punctuation to ensure a noise-free dataset.

Their work is of special significance in Bangla paraphrase generation, which, as seen earlier, is a decent DA technique. They have analyzed different model performances paraphrasing dataset and IndicParaphrase dataset. They have shown that BanglaParaphrase maintains better evaluation results on all metrics (BLEU, ROUGE-L, BERTScore, etc.).

2.4.2 BanglaNLG

In their work, Bhattacharjee et al. [38] developed a robust benchmark called BanglaNLG for assessing the performance of natural language generation (NLG) models specific to the Bangla language. This benchmark encompasses six diverse tasks tailored for conditional text generation: Machine Translation (MT), Text Summarization (TS), Question Answering (QA), Multi-turn Dialogue (MTD), News Headline Generation (NHG), and Cross-lingual Summarization (XLS).

During this research, they also compiled a novel dataset tailored for dialogue generation. Additionally, they have utilized a substantial corpus of Bangla text, totaling 27.5 GB, to train a new model known as 'BanglaT5'. This model, based on the Trans-

former architecture and designed for sequence-to-sequence tasks, sets new benchmarks in performance. In comparison to various multilingual models, BanglaT5 exhibits enhancements, achieving an absolute gain of up to 9% and a relative gain of 32% across all tasks.

2.4.3 Bangla-English Code Mixed DA

By collecting online reviews of different products and constructing an annotated Bangla-English code mix (BE-CM) dataset, Tareq et al. [39] addresses the issue of scarcity of annotated code-mixed data in the Bangla-English segment.

Proposed Augmentation Process

They extend dictionary-based code-switching data augmentation by introducing hierarchical sampling rates to improve cross-lingual alignment in monolingual word embeddings. Words in Bangla are transformed to their English equivalents and vice versa at different sampling rates. This method ensures better cross-lingual adaptation by converting most words while preserving contextual integrity. Additionally, the original text is included in the augmented corpus to increase training diversity, enhancing the alignment of word embeddings for tasks like sentiment prediction. Fig 2.4 illustrates the augmentation process.

2.4.4 Bnaug

”bnaug” is a sophisticated text augmentation tool tailored for Bangla text, designed to enhance the diversity and robustness of textual datasets. It offers a range of augmentation techniques, allowing researchers to select and apply methods best suited to their specific needs in Bangla Natural Language Processing (NLP). Below, we detail the augmentation methods implemented in this framework:

1. **Sentence Augmentation**

    - **Token Replacement**: This technique involves replacing individual tokens (words) in a sentence with alternatives. The alternatives are selected based on predefined

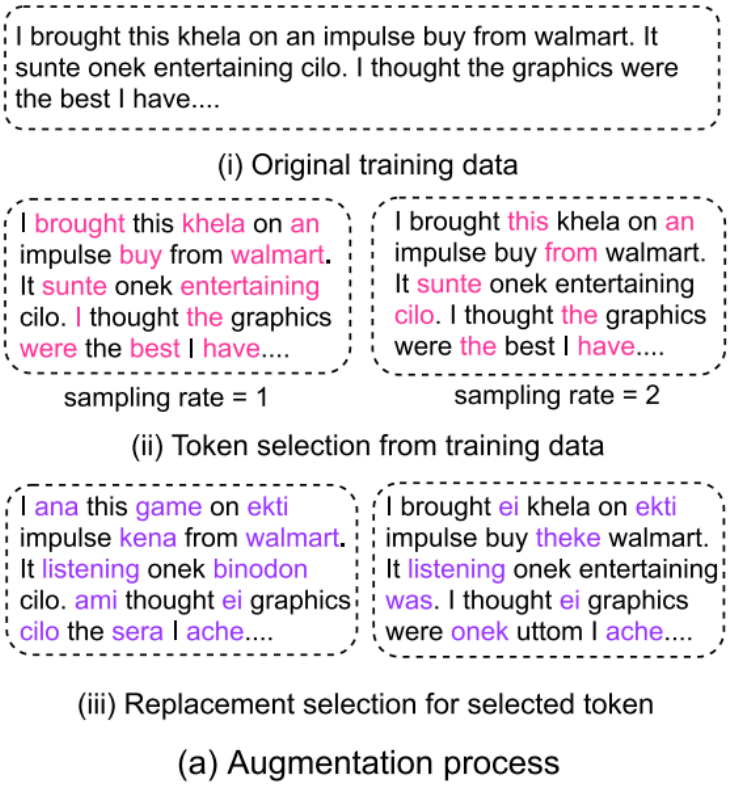


Figure 2.4: Augmentation Process in BE-CM

augmentation strategies, which could include synonyms, related words, or contextually appropriate substitutes. This helps in generating multiple versions of the same sentence, enhancing the lexical variety in the dataset.

2. **Mask Generation Based Augmentation**

   - **Masking Based**: This method generates augmented sentences by masking certain words within the original sentence and then predicting replacements for these masked words. This approach leverages language models trained on Bangla text to ensure the replacements are contextually appropriate.

3. **Word2Vec Based Augmentation**

   - **Word2Vec Based**: Utilizing Word2Vec embeddings, this method replaces words in a sentence with semantically similar alternatives. Word2Vec is a neural network-based model that captures word associations based on their co-occurrences in large text corpora. This method ensures that the augmented sentences remain meaningful while introducing slight variations.

4. **GloVe Based Augmentation**

    - **GloVe Based**: Similar to the Word2Vec method, GloVe (Global Vectors for Word Representation) embeddings are used to find and replace words with semantically similar alternatives. GloVe captures global statistical information about word co-occurrences in a corpus, allowing for effective word substitutions that maintain sentence coherence.

5. **Back Translation**

    - **Back Translation**: This technique involves translating a sentence from Bangla to another language (such as English) and then translating it back to Bangla. This process often results in variations of the original sentence due to differences in language structure and vocabulary. This method introduces natural diversity while preserving the original meaning.

6. **Text Generation**

    - **Paraphrase Generation**: Paraphrase generation involves creating different versions of a given sentence that convey the same meaning. Various models and algorithms, such as transformer-based models, can be used for this purpose. This method is valuable for expanding the dataset with contextually equivalent sentences.

7. **Random Augmentation**

    - **Random Remove**: This technique removes random parts of the sentence, such as words, stopwords, punctuations, or digits, to generate new sentences. The randomness introduces variability while testing the model's robustness.

    - **Remove Digits**: Specifically removes digits from the text. For instance, ”I am 25 years old” would become ”I am years old”.

    - **Remove Punctuations**: Removes punctuation marks from the text, which can affect sentence readability and meaning. For example, ”He said, 'I will come.'” could become ”He said I will come”.

- **Remove Stopwords**: Removes common stopwords, such as ”and”, ”but”, to alter the sentence structure and highlight the impact of stopwords on text analysis. For instance, ”I eat rice and listen to podcast” might become ”I eat rice listen to podcast”.
- **Remove Random Word**: Removes a random word from the sentence to create a new variation. For example, ”I eat rice” might become ”I eat”.
- **Remove Random Character**: Removes a random character from the sentence. This can introduce noise, useful for testing the robustness of text processing models. For example, ”I eat” might become ”I et” or ” eat”.

”bnaug” is a handy toolkit for increasing model robustness. This toolkit demonstrates the potential of these DA methods in enriching Bangla training data but it lacks performance benchmarks to prove its effectiveness. There exists no documentation or research paper associated to this toolkit to the best of our knowledge. Additionally, the token replacement approach is also prone to label-changing issues as discussed by Kesgin and Amasyali [40], and requires an additional filtering process to work effectively and it does not include any filtering mechanisms to ensure the quality of the augmented text. Here, quality refers to introducing sufficient variation while preserving the original text’s label and meaning.

Chapter 3

Proposed Methodology

Classification models generally exhibit suboptimal performance when trained on limited data. To tackle this issue, our approach involved incorporating multiple data augmentation methods to expand the dataset size, which was then evaluated to determine any enhancements in model efficacy. The augmentation operations applied to each sentence within the training dataset were:

1. **Synonym Replacement (SR):** Randomly select *n* words that are not stopwords in a sentence and substitute them with their closest synonyms based on Word2Vec embeddings.
2. **Random Swap (RS):** Randomly generate two indices in the sentence and swapping the words at those indices. This process is repeated *n* times.
3. **Back-Translation (BT):** Convert a sentence from Bengali into a pivot language (English) and then re-translate it back to Bengali.
4. **Paraphrasing (PP):** Rephrase the original Bengali text while maintaining its semantic meaning.

For SR, we use Bengali Word2vec embeddings provided by bnlp-toolkit[1]. For BT, we utilize the pre-trained checkpoint of the BanglaT5 model fine-tuned on the BanglaNMT dataset Hasan et al. [41] to generate translations. The PP method is similar to BT but employs a different aligner, using the paraphrase checkpoint of BanglaT5 fine-tuned on the BanglaParaphrase dataset Akil et al. [42]. For the amount of augmentation, we pass

[1] https://github.com/sagorbrur/bnlp

the whole dataset to BDA, create a 1:1 augmented version of it by randomly selecting a method, and merge it with the normal dataset. Thus the augmented version contains twice the samples of the normal version, half of which came from BDA. For rule-based methods(synonym replacement and random swap), we choose a small value (n=2) as suggested by Wei and Zou [43] for augmentation.

Longer sentences, having more words than shorter ones, can tolerate a higher level of noise while preserving their original class label. This allows for introducing more lexical variations by increasing the augmentation level (*n*, the number of words altered) in the augmented text for rule-based methods SR and RS.

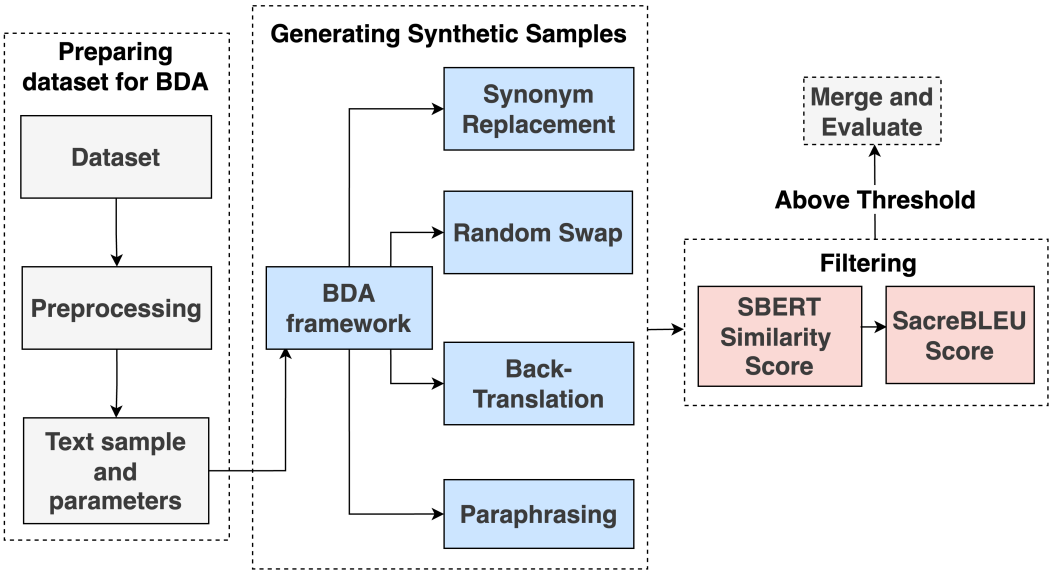


Figure 3.1: BDA Pipeline

Another challenge in text augmentation for Bengali is that many sentences can only be represented in a limited number of ways without altering the label which requires us to filter such low-quality augmented texts. To maintain lexical variation while ensuring high semantic similarity, we set a threshold for filtering the generated augmented texts from BDA.

3.1 Data Preprocessing

Before applying our data augmentation techniques, it is crucial to prepare the dataset through a comprehensive data preprocessing phase. This phase ensures that the data is

Operation	Sentence
Normal	সঠিক তদন্ত করতে হবে বিচারের আওতায় আনতে হবে যে এই কাজ টা করেছ.
SR	সঠিক আপিল করতে হবে বিচারের আওতায় দিতে হবে যে এই কাজ টা করেছে
RS	সঠিক তদন্ত করতে হবে বিচারের আওতায় হবে আনতে যে এই কাজ টা করেছে
BT	সঠিক তদন্ত অবশ্যই করতে হবে এবং যে ব্যক্তি এটা করেছে তাকে বিচারের আওতায় আনতে হবে
PP	সঠিক তদন্ত করতে হবে এবং এই কাজটি বিচারের আওতায় আনতে হবে যে এটি করেছে

Table 3.1: Sentences generated using BDA

in an optimal format for processing, leading to more effective and accurate augmentation results.

- **Text Cleaning:** Remove irrelevant or noisy elements from the text, such as HTML tags, URLs, non-standard symbols, or extraneous white space. Regular expressions and text-processing libraries are employed for systematic cleansing. Also, digits and stopwords are removed as required.
- **Normalization:** We used the normalizer Hasan et al. [44] module to normalize punctuation and characters that have multiple Unicode representations to reduce data sparsity.
- **Tokenization:** We broke down the text into smaller units (tokens) like words or subwords. Language-specific tokenizers handle the structural and linguistic nuances of Bengali text.

This pre-processing phase is fundamental, laying the groundwork for effective and accurate data augmentation. By preparing the text through these steps, we ensure that the subsequent augmentation techniques are applied with optimal efficiency and precision, leading to a robust and diverse dataset for Bengali language processing.

3.2 Data Augmentation Techniques

1. **Back Translation (BT)**: This process involves translating a sentence from Bengali to a pivot language (like English) and then back to Bangla. This often results in a sen-

tence with similar meaning but different phrasing, providing a nuanced variation of the original text.

A key aspect of using BT in data augmentation is to ensure that the meaning of the original text is retained throughout the translation process. The choice of models is crucial here, as they need to be capable of capturing and preserving the semantic essence of the text during translation. This is particularly challenging when dealing with the linguistic and cultural nuances of Bangla.

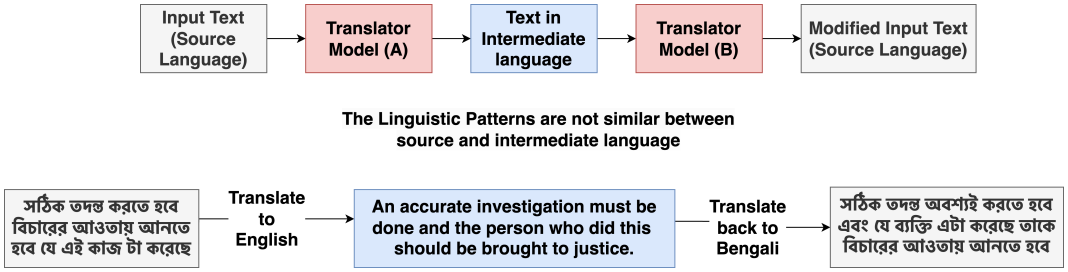


Figure 3.2: Back Translation

In our data augmentation framework, we have utilized BT specifically focusing on the language pair of Bengali and English. This approach is designed to generate linguistically varied yet semantically similar text to the original Bengali sentences. The process involves two main stages: translation from Bengali to English and then back to Bangla. The variance is introduced due to the linguistic differences between these two languages. For the Bangla-to-English translation, we used the pretrained checkpoint of the BanglaT5 model fine-tuned on the BanglaNMT dataset, the largest Machine Translation (MT) dataset for Bengali-English, proposed in Hasan et al. [41]. For the reverse translation from English back to Bangla, we used the other fine-tuned checkpoint of the same model.

2. **Synonym Replacement**: This is a rule-based technique. It randomly chooses an arbitrary number of words from the sentence that are not stopwords and replaces them with one of their synonyms chosen randomly.

   In BDA, Synonym Replacement (SR) is utilized to introduce lexical diversity into the input text. This technique selectively replaces words in a sentence with their synonyms, subtly altering the text while maintaining the original meaning. The following points illustrate our implementation of SR:

- **Word2Vec Model:** The `BengaliWord2Vec` model from the bnlp-toolkit[2] is employed to identify synonyms for Bengali words.

- **Synonym Replacement Function:** The function `synonym_replacement` takes a text and a number `n` (maximum number of words to be replaced). It randomly selects words, excluding stopwords, and replaces them with synonyms using the `get_synonyms` function, which leverages `BengaliWord2Vec`. The rational is that core-words have a better impact on model robustness compared to stopwords which have very little contribution and might even lead to over-fitting.

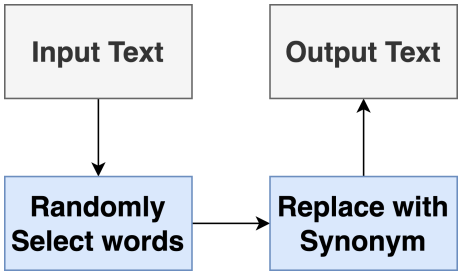


Figure 3.3: Synonym Replacement Pipeline

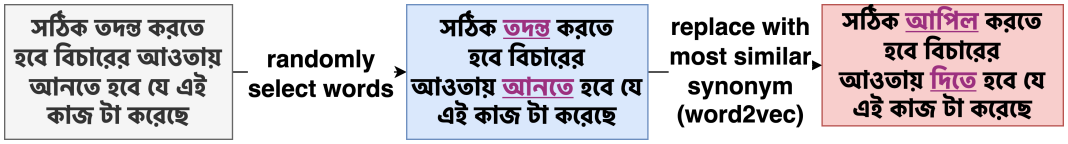


Figure 3.4: Synonym Replacement Example

- **Word Replacement and Sentence Reconstruction:** Up to `n` words in the text are replaced with synonyms and the augmented text is returned.

3. **Random Swap:** This aims to increase syntactic diversity by swapping words within sentences. For an arbitrary number of times, it randomly chooses two words and swaps their positions. The implementation details are as follows:

   (a) *Random Swap Function (`random_swap`):* Takes in text and a number `n`, indicating how many swaps to perform. The text is split into tokens.

[2]https://github.com/sagorbrur/bnlp

(b) *Performing Swaps:* The function `swap_word` is executed `n` times to swap words. Each execution randomly selects two indices in the word list and swaps the corresponding words.

(c) *Swap Word Function (`swap_word`):* Chooses two random indices and ensures they are distinct, with a maximum of three attempts. If distinct indices are found, the words at these positions are swapped.

(d) *Output:* Returns the text with swapped words, thereby enhancing syntactic variation.

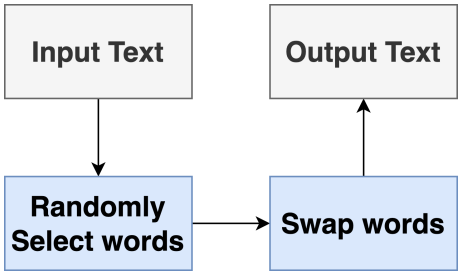


Figure 3.5: Random Swap Pipeline

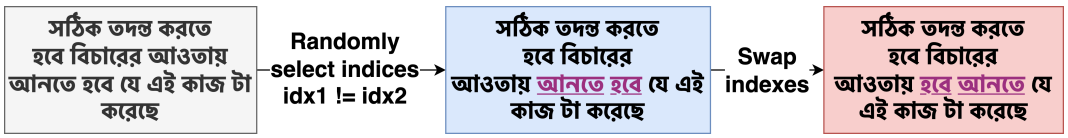


Figure 3.6: Random Swap Example

4. **Paraphrasing**: Paraphrasing is another method we employed for the augmentation of Bengali text data. This is a very viable augmentation approach, allowing us to create varied yet semantically equivalent versions of the original sentences. For our framework, we have used the pre-trained checkpoint of the model BanglaT5 Akil et al. [36] which is finetuned on the BanglaParaphrase dataset. The implementation details are as follows:

   - *Model and Tokenizer Setup:* The `AutoModelForSeq2SeqLM` and `AutoTokenizer` from the `transformers` library are used. Then the pre-trained checkpoint of the BanglaT5 model is used to instantiate the model and tokenizer.

- *Normalization:* The input sentence is normalized using the `normalize` function to ensure standardization and consistency.
- *Tokenization:* The normalized sentence is tokenized with the tokenizer, preparing it for model processing.
- *Paraphrase Generation:* The tokenized input is processed by the model to generate a new set of tokens, which are then decoded back into a paraphrased sentence using `tokenizer.batch_decode`.
- *Output:* The `generateParaphrase` function outputs the paraphrased sentence.

This Paraphrasing method is integral to our data augmentation strategy, creating semantically equivalent yet linguistically varied versions of sentences. It significantly enhances the linguistic diversity of the dataset, crucial for the development of effective and nuanced NLP models.

3.3 Experimental Setup

To assess the effectiveness of the BDA Framework, we designed a comprehensive test bench. To ensure our results are comprehensive and applicable to various scenarios, we selected five datasets with diverse characteristics. These datasets encompass a range of augmentation challenges, including class imbalance, data scarcity, and low sample variation, as well as datasets without these issues. To study the impact of data scarcity, we applied stratified clipping at 15%, 50%, and 100% of the original dataset size. This allows us to observe how augmentation affects performance across varying levels of data availability in BDA.

3.4 Text Classification Models

We chose BanglaBERT Bhattacharjee et al. [45] and linear SVM models to test the classification performance of BDA. SVM was used with traditional hand-crafted linguistic features, word (1-3), and character (2-5) n-grams by vectorizing each instance with the

TF-IDF weighted scores for each n-gram Islam et al. [46]. To ensure consistency across experiments, we kept the following parameters for our BanglaBERT model: The ra-

Parameter	**Value**
Maximum Sequence Length	256
Learning rate	2e-5
Number of epochs	6

Table 3.2: Classification Parameters for BanglaBERT

tionale for choosing these methods to represent classification performance is that SVM shows superior performance while predicting the majority class while BERT-based classifiers predict the minority classes better as shown by studies such as Baruah et al. [47]. So testing on these two classifiers can cover a wide range of real-world scenarios.

Chapter 4

Experimental Design

4.1 Benchmark Datasets

To test how BDA performs, we chose 5 standard datasets with a training set split as follows:

The BemoC dataset Iqbal et al. [48] comprises 48,328 unique words and encompasses

Dataset	Samples	Classes
SentNoB	15,728	3
BemoC	48,328	6
ABSA (Cricket)	1,787	5
ABSA (Restaurant)	1,235	5
Bengali Sentiment	11,851	3
VITD	2,700	3

Table 4.1: Benchmark Datasets

6 labels, offering a substantial vocabulary and multi-class classification challenge. The SentNoB dataset Islam et al. [46] contains 15,728 samples and 3 labels, this dataset exhibits an average of 15.37 words per instance, mostly containing noisy text samples. ABSA dataset Rahman and Dey [49] contains data from two origins (Cricket has 1787 samples and Restaurant has 1235 samples) and features 5 aspect categories. It allows us to examine augmentation in aspect-based sentiment analysis tasks. Bengali Sentiment Islam et al. [50] dataset contains 3 classes and 11851 samples, offering a standard sentiment analysis benchmark. Another extra dataset, VITD Saha et al. [51] was chosen

which contains 2700 samples across 3 labels and highly informal language. This dataset is slightly different from other datasets as it is the only dataset on violence-inciting texts, with the label for violence being split into 2 similar classes (direct and indirect) which makes it harder to understand and augment for BDA.

4.2 Evaluation Criteria:

The key characteristics of good augmented texts are lexical diversity, syntactic variance, and semantic preservation. To evaluate the effectiveness of the BDA Framework in producing high-quality augmented texts, we assessed these aspects using the following four criteria:

1. **F1 score Improvement:** The F1 score, being the harmonic mean of precision and recall, is less affected by class distribution compared to accuracy. This makes it a more reliable metric for evaluating performance, particularly in imbalanced datasets. Solving this issue is a primary concern of augmentation.
2. **Lexical Variance:** To check how different the words and their order are, use SacreBLEU (a standardized implementation of Bilingual Evaluation Understudy) Post [52] which checks n-gram overlaps between the normal and BDA-generated text.
3. **Semantic Similarity:** To address whether BDA's texts contain the inherent meaning, we used SBERT (Sentence BERT) Deode et al. [53] to generate embeddings of these sentences which are then compared using semantic similarity.

Chapter 5

Results

Our results on BDA consist of five different datasets on 3 clipping ranges (15%, 50%, 100%) using BanglaBERT and variants of SVM. For all experiments, we average results from all four of our approaches to determine how much improvement in F1 scores we can obtain.

5.1 BDA's Performance on Datasets

For each of the 5 datasets, we tested the performance of BanglaBERT and SVM as they cover a wide range of use cases SVM is a lightweight model whereas BanglaBERT represents the effect on larger models. Also, SVM tends to work better with more amount of samples provided whereas BanglaBERT generally improves according to the diversity and quality of the dataset, as shown by Baruah et al. [47]. The F1 score improvement across different clipping ranges are shown below:

Model	**Normal,** T **%**			**Augmented,** T' **%**			**Difference** $(T' - T)$ **%**		
	15	**50**	**100**	**15**	**50**	**100**	**15**	**50**	**100**
BanglaBERT	64.86	66.91	70.73	64.97	68.10	72.04	0.11	1.19	1.31
SVM	53.69	59.54	62.42	53.83	59.76	65.79	0.14	0.22	3.37
Average	59.28	63.22	66.58	59.40	63.93	68.91	0.12	0.71	2.33

Table 5.1: SentNoB Dataset

Model	**Normal, T %**			**Augmented, T' %**			**Difference $(T' - T)$ %**		
	15	**50**	**100**	**15**	**50**	**100**	**15**	**50**	**100**
BanglaBERT	42.82	68.98	71.57	61.94	69.39	70.56	2.33	0.41	-1.01
SVM	41.86	50.52	54.16	42.24	50.44	54.18	0.38	-0.08	0.02
Average	42.34	59.75	62.86	52.09	59.91	62.37	1.36	0.16	-0.49

Table 5.2: BemoC Dataset

Model	**Normal, T %**			**Augmented, T' %**			**Difference $(T' - T)$ %**		
	15	**50**	**100**	**15**	**50**	**100**	**15**	**50**	**100**
BanglaBERT	47.45	48.95	49.64	47.74	49.19	49.08	0.29	0.24	-0.56
SVM	36.38	37.90	39.51	37.01	38.70	39.90	0.63	0.80	0.39
Average	41.92	37.46	44.58	42.38	43.94	44.49	0.46	0.00	-0.09

Table 5.3: Bengali Sentiment Dataset

Model	**Normal, T %**			**Augmented, T' %**			**Difference $(T' - T)$ %**		
	15	**50**	**100**	**15**	**50**	**100**	**15**	**50**	**100**
BanglaBERT	27.98	49.27	54.83	41.90	56.05	57.45	**13.92**	6.78	2.62
SVM	34.80	45.24	44.43	35.75	46.74	50.54	0.95	1.50	6.11
Average	38.37	47.26	49.63	38.83	51.39	53.99	7.43	4.13	4.36

Table 5.4: ABSA Cricket Dataset

Model	**Normal, T %**			**Augmented, T' %**			**Difference $(T' - T)$ %**		
	15	**50**	**100**	**15**	**50**	**100**	**15**	**50**	**100**
BanglaBERT	18.60	37.01	44.73	26.14	45.59	55.28	7.54	**8.58**	**10.55**
SVM	24.32	30.57	36.53	25.55	31.67	36.72	1.23	1.10	0.19
Average	21.46	33.79	40.63	25.84	38.63	46.00	4.38	4.84	5.37

Table 5.5: ABSA Restaurant Dataset

5.2 Analysis of the Results

To determine how effective BDA is, we calculated the difference in F1 values for a particular model, between normal text, T, and its augmented version T'. The results are as follows:

SentNoB Dataset (Table 5.1)

- BanglaBERT: Shows slight improvements in F1 score as the dataset size increases with augmentation, especially from 15% to 100% dataset size (difference increases from 1.1% to 1.31%).
- SVM: Also shows notable improvements with augmentation, particularly when 100%

data is augmented we get 3.37% improvement, which aligns with the findings of Baruah et al. [47] which also showed higher values for SVM when majority classes were well understood.

BemoC Dataset (Table 5.2)

- BanglaBERT: Has the highest increase in difference for the smallest dataset size (15% dataset size shows a 2.33% increase) but shows a reduction for the largest dataset size.
- SVM: A similar trend is seen here where an increase in sizes shows a decrease (0.38% to 0.02% as size increases) in F1 after augmentation. This can be due to the dataset is already saturated and augmentation is trying to overfit the models.

Bengali Sentiment Dataset (Table 5.3)

- BanglaBERT: This shows a minor improvement through augmentation, but the improvement declines as the dataset becomes saturated.
- SVM: A similar scenario can be seen here as well, with minor improvements

ABSA Cricket Dataset (Table 5.4)

- BanglaBERT: Shows a consistent decrease in F1 score improvement with the dataset size; the largest improvement (13.92%) is seen in the 15% dataset size and it decreases as the dataset gets saturated.
- SVM: Shows the most substantial increase in performance(6.11%) in the 100% dataset, again demonstrating that more data through augmentation helps improve SVM's performance.

ABSA Restaurant Dataset (Table 5.5)

- BanglaBERT: It steadily increases as the dataset size increases, showing a massive 10.55% improvement when BDA is applied on 100% of the dataset.
- SVM: Also shows improvements with augmentation, but only when the dataset size is small.

We present the full results of the performance of all methods in each dataset in separate tables in the Appendix(A) section.

5.3 Augmentation Method Comparison

From the results shown in Tables 5.1-5.5 it is pretty evident that BDA methods contribute to performance gain. But how one method fares against others is still a matter of question. Table A.1 shows the comparison of the average model performance across all 5 datasets for each method. A detailed analysis of the performance of each method is shown in our ablation study.

Rule-based vs Model-based Methods

Rule-based approaches modify text via simple token-level perturbations, while model-based approaches leverage pretrained models for more data variation. Figure 5.1 shows both rule-based and model-based methods improve F1 scores as training data increases. Model-based methods consistently outperform rule-based methods, particularly with more data, due to their ability to generate diverse and contextually relevant modifications.

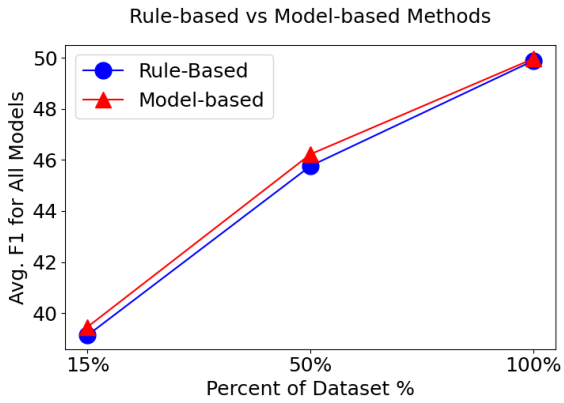


Figure 5.1: Performance Comparison of Rule-Based and Model-based Methods

5.4 Observations

1. **Effect of Augmentation:** Both models generally benefit from data augmentation, especially as the dataset size increases. Augmentation introduces variation in the dataset

allowing the model to generalize better. We see across all clipping ranges of the SentNoB and the ABSA Restaurant dataset, the F1 score improved with augmentation, along with other datasets showing improvement in the majority of the cases. Also, the highest amount of improvements can be seen in the 15% range as well, which is an indicator of BDA's ability to solve data scarcity.

2. **Model Comparison:** BanglaBERT generally shows a more robust improvement in F1 scores with augmentation across different datasets and sizes compared to SVM. For instance, in the BemoC dataset, BanglaBERT consistently outperforms SVM variants with or without augmentation. This is due to BanglaBERT's capabilities as a BERT model, which is better at capturing contextual information from augmented text. On the other hand, SVM's performance improves with a larger amount of data but falls short in comparison to BanglaBERT. A detailed study of each model's performance on different clipping ranges is shown in Table A.8.

3. **Impact of Dataset Size:** The augmentation tends to have a more significant impact as the dataset size increases, especially noticeable in more complex models like BanglaBERT. This could indicate that while smaller datasets benefit from augmentation, larger datasets provide a more substantial foundation allowing the models to leverage the augmented data effectively. The ABSA Restaurant Dataset supports this, showing increased performance gain with dataset size increase when using BanglaBERT (7.54%, 8.58% and 10.55%).

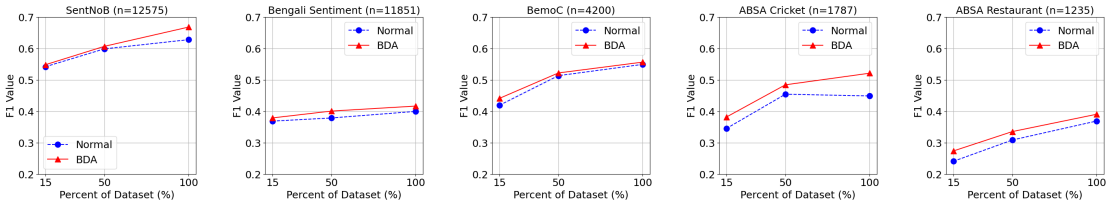


Figure 5.2: Improvement of F1 scores across various datasets, before and after augmentation

5.5 Effect of Training Set Size

Overfitting is known to be more pronounced with smaller datasets. We artificially created data scarcity to test this hypothesis by clipping the dataset size into 15%, 50%, and 100% ranges using a stratified approach to maintain class balance, which was used for both standard training and BDA-enhanced training. Our experiments, using a reduced portion of the training data, reveal that BDA's performance improvement is more pronounced with smaller training sets, as depicted in Figure 5.2. We hypothesize that BDA introduces impactful variation into constrained datasets with a limited number of original samples, leading to improved F1 scores. For instance, the F1 score of BanglaBERT without augmentation when trained on 100% dataset is 54.83%. When trained using BDA this number is surpassed by achieving a score of 56.05% while only using 50% of the available training data.

5.6 Conserving True Labels

A critical part of good-quality text augmentation is how well we can conserve the class or label of the data. Since the lexical variation can be modified using parameters of the BDA pipeline, we do not want to hard-limit these changes as long as they generate good augmented texts, instead, we chose to pass these generated texts through a filtering process as shown in Figure 5.3 that blocks the texts from appending to the generated dataset based on it's meaning retention.

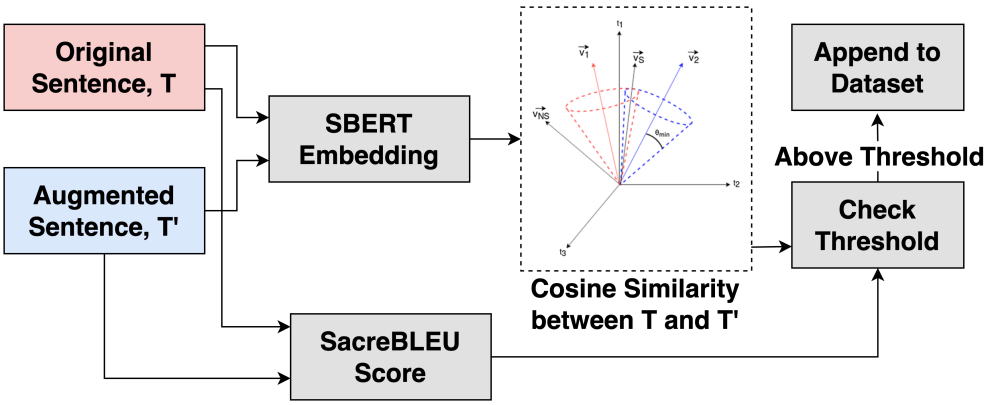


Figure 5.3: Filtering Process

The original sentence and the augmented sentence are passed through a multilingual Sentence-BERT model Deode et al. [53] to create vector representations of them. Then the cosine similarity is calculated to find how similar the texts are in terms of meaning. If it is above the threshold the texts are forwarded to the second phase where the lexical similarity is checked. In the case of lexical similarity, we use SacreBLEU to check n-gram overlaps in the sentences, the lower the score, the lower the number of similar words in the augmented sentence. If it is below the threshold, only then the texts are appended to the dataset. The threshold mostly depends on the dataset and the trade-off between lexical variance and the augmented dataset's sample count as augmented samples that defer from the original meaning will not pass through the filtering process.

5.7 Ablation Study: BDA Decomposed

The results obtained so far look promising. However one might speculate that only a particular part of the pipeline is primarily responsible for BDA's performance gains, so we isolate each operation to assess their independent contributions. We individually check all four operations(SR, RS, BT, PP) for augmentation of BDA across all five datasets. The results reveal that each of the four BDA operations plays a role in enhancing performance.

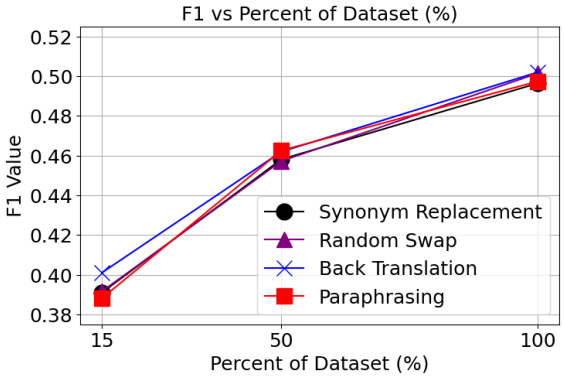


Figure 5.4: Performance Comparison of each Augmentation Method

The lexical and semantic analysis reveals that rule-based approaches, i.e. Synonym Replacement, and Random swapping result in very high SacreBLEU scores compared to Back-Translation or Paraphrasing, but are quite comparable in terms of meaning re-

tention, as shown in table 5.6.

Method	Similarity %	
	Lexical	**Semantic**
SR	59.61	83.86
RS	48.57	94.06
BT	11.29	79.15
PP	15.40	86.97

Table 5.6: Assessing the Quality of the Augmented Texts for each method

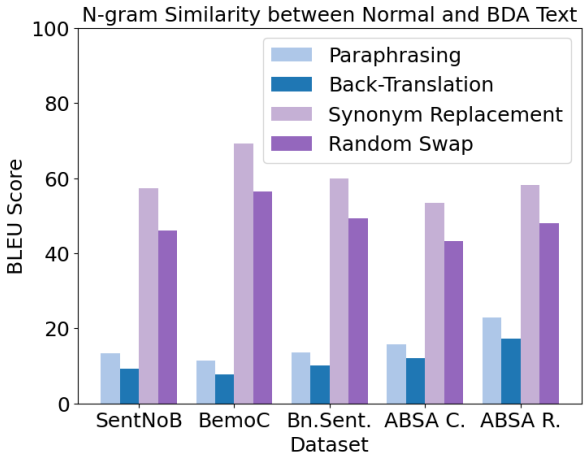


Figure 5.5: Lexical Similarity

This is because, while Back-Translation and Paraphrasing alter the complete structure of the sentences, Synonym Replacement and Random swapping mostly depend on the number of words chosen to be altered, using the provided parameters. Random Swapping (RS) retains the highest amount of semantic similarity. This is likely because, although the Sentence-BERT model Deode et al. [53] incorporates positional encoding to capture word order, RS only involves swapping existing words without introducing new ones. Therefore, the underlying semantics remain largely unchanged. To study the impact in terms of F1 score, we checked the average F1 score for all the datasets[1]. Here, in Figure 5.4, we can see that, there is no best augmentation method for all cases, but rather each method plays a valuable role in generating variations. Notably, in highly constrained datasets (15%), Back-Translation works the best as it usually introduces a different sentence structure with high variance, while Random Swap does well when the

[1]Detailed table of comparison presented in the Appendix

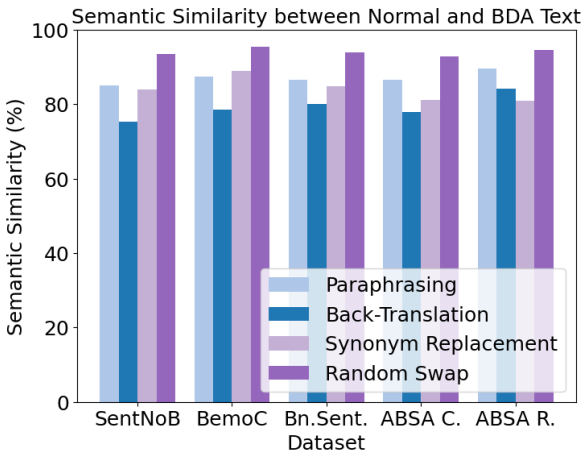


Figure 5.6: Semantic Similarity

dataset is saturated (100%) and has very little room to introduce variations. Synonym Replacement shows decent results on highly constrained (15%) scenarios but it does not work well when there are lots of samples in the dataset. Paraphrasing, on the other hand, performs consistently across all splits and shows the best performance with mildly constrained (50%) training data. But none of these are consistently the better method across all conditions. However one might prefer the rule-based approach (SR, RS) as a faster augmentation method.

Dataset Specific Observations

In the case of transformed-based approaches, we can expect a more realistic pattern of observation as rule-based methods will alter word and meaning similarity in a predetermined manner. Thus those can serve as a indicator of meaning similarity.

- **Sentiment and ABSA datasets:** F1 score is particularly high on these datasets, which could be due to the focused nature of the text (such as user reviews), where key semantic elements are clear and maintained across different augmentation techniques. But a concern is the high word similarity, which might lead to overfitting.
- **BeMOC:** This dataset shows good semantic preservation across methods, suggesting that its content, despite possible complex syntactic structures, is robust against semantic distortion.

Dataset	Word Similarity	Meaning Similarity
sentnob	13.433	0.850
bmoc	11.443	0.873
sentiment	13.561	0.865
absa cricket	15.709	0.866
absa restaurant	22.841	0.895

Table 5.7: Paraphrasing Similarity

Dataset	Word Similarity	Meaning Similarity
sentnob	9.211	0.752
bmoc	7.840	0.786
sentiment	10.100	0.801
absa cricket	12.054	0.779
absa restaurant	17.236	0.841

Table 5.8: Back Translation Similarity

Dataset	Word Similarity	Meaning Similarity
sentnob	57.404	0.838
bmoc	69.155	0.889
sentiment	59.954	0.848
absa cricket	53.421	0.811
absa restaurant	58.096	0.808

Table 5.9: Synonym Replacement Similarity

Dataset	Word Similarity	Meaning Similarity
sentnob	45.969	0.935
bmoc	56.454	0.954
sentiment	49.291	0.940
absa cricket	43.219	0.929
absa restaurant	47.926	0.946

Table 5.10: Random Swap Similarity

Both analyses underline the importance of selecting the right text augmentation method based on the specific needs of the dataset and the NLP task at hand. While syntactic fidelity and semantic integrity are both crucial, the choice of method might prioritize one over the other depending on the application, such as training data for machine learning models where syntax and semantics play differing roles depending on the model's use case.

5.8 Ablation Study

This graph compares the impact of four different data augmentation techniques—synonym replacement, random swap, back translation, and paraphrasing—on the F1 score across three dataset sizes (15%, 50%, 100%). As the dataset size increases, all techniques show

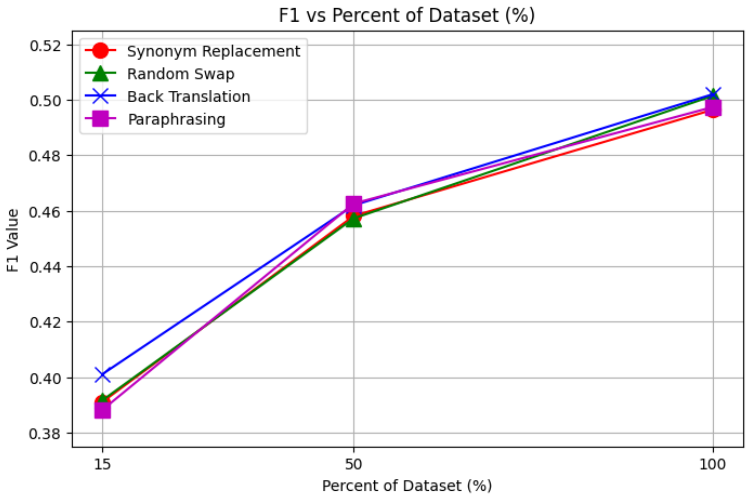


Figure 5.7: Performance of Each method

a progressive improvement in F1 scores, which indicates that augmentation helps in enhancing model understanding and generalization capabilities. The convergence of F1 scores at 100% dataset size suggests that the augmentation effects are maximized when the dataset is small and become less distinguishable as the data volume increases. This graph demonstrates that complex techniques like back translation and paraphrasing tend to perform slightly better, possibly due to introducing more diversity into the training data compared to simpler methods like synonym replacement.

5.9 Effectiveness of Augmentation

Augmentation generally improves model performance, particularly in scenarios where data is limited. This is evident from the consistent higher F1 scores in augmented datasets across most graphs.

- **Dataset Size Impact**: As the dataset size increases, the relative benefit of augmentation decreases, but it still contributes to higher performance, suggesting its utility in

overcoming data scarcity and enhancing model robustness.

- **Technique Specificity**: Different augmentation techniques may be more or less effective depending on the specific dataset and task. The choice of technique should be guided by empirical results like these.

Across all these diagrams, data augmentation demonstrates a consistent ability to enhance model performance, particularly in environments where data is scarce or the task is highly specific. These graphs illustrate the strategic importance of choosing the right augmentation techniques and applying them effectively to maximize model accuracy and robustness.

5.10 Where augmentation fails

The two graphs provide insights into the performance and challenges associated with BanglaBERT's vocabulary coverage and the effectiveness of text augmentation in the VITD model.

VITD Original vs Augmented F1 Score

The line chart compares the F1 scores of the VITD model on original and augmented datasets across various dataset usage percentages. The original dataset consistently outperforms the augmented dataset, with the original dataset achieving higher F1 scores across all percentages. For instance, at 10% dataset usage, the original dataset scores 0.50, whereas the augmented dataset scores 0.45. This trend continues, with the original dataset achieving an F1 score range of 0.72 to 0.81, while the augmented dataset lags behind with scores between 0.68 and 0.72.

From these graphs, we can draw the following conclusions to understand the reason behind the performance decline:

1. **Effectiveness of Text Augmentation:**

    - The ineffectiveness of text augmentation in improving the model's performance (Graph 2) implies that the augmented data may not be adding meaningful diver-

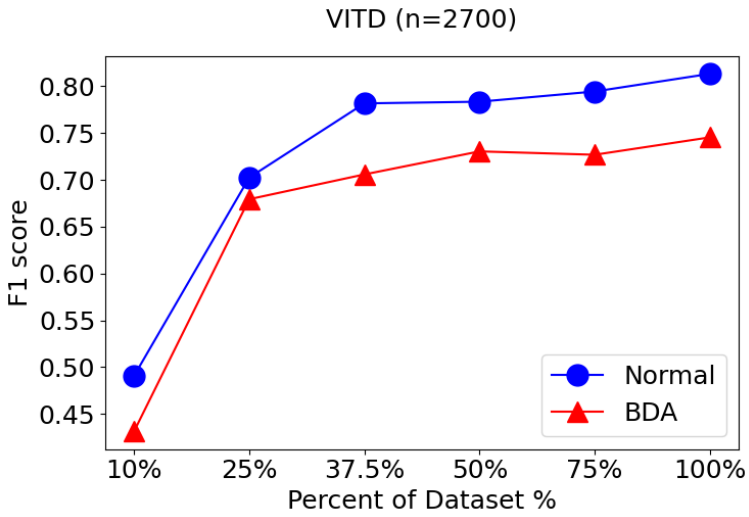


Figure 5.8: Performance hit (F1) after Augmentation in VITD dataset

sity. Instead, it might be amplifying existing issues like informal text and spelling mistakes.

- The relatively better coverage of the Dev dataset highlights the importance of having a well-aligned vocabulary. Augmentation techniques need to ensure that they enhance this alignment rather than disrupt it.

2. **Alignment and Noise:**

   - The Dev dataset's higher vocabulary coverage and slightly better alignment with BanglaBERT suggest that careful selection and preprocessing of data can significantly impact model performance.

   - The introduction of noise through augmentation demonstrates that not all data augmentation methods are beneficial, especially if they do not consider the specific vocabulary and linguistic characteristics of the dataset.

The comparative analysis of the two graphs underscores the crucial role of vocabulary alignment in NLP model performance and the potential pitfalls of text augmentation techniques if not applied judiciously. Improving vocabulary coverage and carefully designing augmentation methods are essential for enhancing the effectiveness of language models like BanglaBERT.

Chapter 6

Discussion

While our results indicate that BDA can improve the performance of models on a particular dataset, this is not always the case. The outcome depends largely on the dataset itself, and how much of the train set can be effectively augmented so that it represents the test. In the case of the VITD dataset, the results given in figure 6.1 show that BDA

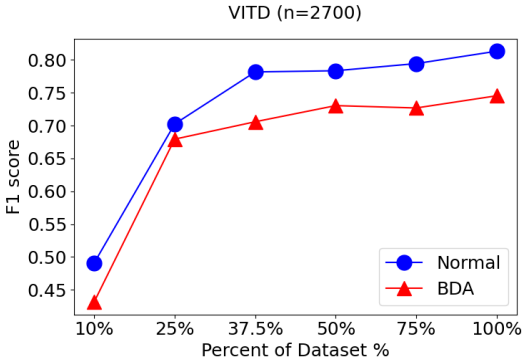


Figure 6.1: F1 score decreasing in VITD dataset

fails to improve the F1 score, and even leads to a decrease in F1 score compared to without BDA. To identify the reason, we analyzed the confusion matrix before and after applying BDA.

In the confusion matrix of the prediction set after augmenting via BDA shown in Figure 6.3 we can see that the amount of False-positive and false negatives increased for 'Non-violence' text from the normal dataset shown in Figure 6.2. This can occur due to noisy texts causing BDA to generate labels that alter the label of the text. Thus the filtering process might need to be improved to prevent such label flipping. Another

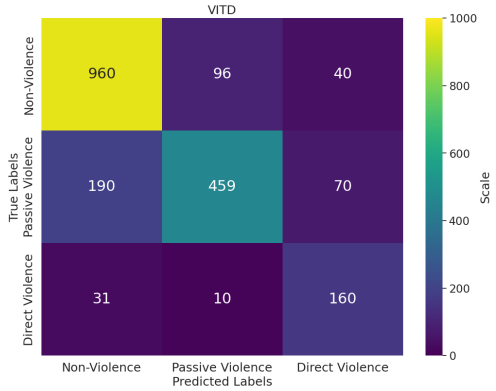


Figure 6.2: Before Augmenting via BDA (full dataset)

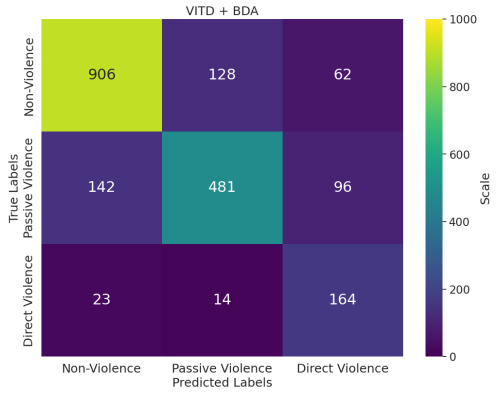


Figure 6.3: After Augmenting via BDA (full dataset)

reason can be overfitting the augmented dataset.

A deeper analysis of our experimental results reveals some key aspects of Bangla text augmentation. These are:

1. **Transformer-based augmentation pipelines generate the most natural augmented texts in general, according to similarity metrics.** This can be because these models alter the whole sentences instead of a set of tokens, thus leading to exposure to a wider array of lexical variations.

2. **Model vs. Rule-Based augmentation approaches are within roughly 1-2% of F1 scores across all datasets.** This suggests that both approaches can yield comparable results in some scenarios depending on the dataset. Transformer-based methods produce more diverse and contextually relevant augmentations, rule-based methods are faster and simpler.

3. **No universal best augmentation method.** The optimal augmentation technique may

vary depending on the specific dataset, task, and model architecture as shown in figure 5.4, highlighting the need for a unified approach using a combination of augmentation methods.

We propose BDA, a ”Bengali text Data Augmentation Framework,” a comprehensive system designed to artificially enhance and enrich Bengali text datasets. BDA integrates various text augmentation methods, encompassing existing state-of-the-art techniques and pioneering new ones. We evaluate its performance across five distinct datasets, presenting our results to substantiate our methodology. Additionally, we conduct syntactic and symmetric analyses to determine the most effective pipelines, noting that augmentation does not consistently improve performance. In cases like out-of-vocabulary (OOV), no performance gains were observed. Similarly, when datasets are saturated and exhibit no scarcity, augmentation proves less effective. Conversely, the best case for augmentation occurs when datasets are limited in size.

Our framework employs a range of augmentation techniques, including Back Translation (BT), Iterative Masked Filling (IMF), and Easy Data Augmentation (EDA) methods such as Synonym Replacement (SR), Random Deletion (RD), Random Insertion (RI), and Random Swap (RS). The augmentation threshold (AT) concept ensures the augmented texts maintain a balance between syntactic fidelity and semantic integrity. This is crucial for Bengali due to its rich syntactic and semantic nuances.Our evaluations indicate that not all augmentation techniques uniformly enhance model performance. Specifically, in datasets with an abundance of data (saturated datasets), the impact of augmentation diminishes. Conversely, in scenarios where data is sparse, augmentation techniques significantly improve model performance. This is particularly evident in low-resource settings where the diversity introduced by augmentation helps models generalize better.

Among the augmentation methods tested, BT and Paraphrase (PP) consistently outperformed others across various metrics. These methods excel in introducing meaningful variability while preserving the original context, thus enhancing model robustness. However, techniques like SR and RS, while faster, demonstrated competitive performance, especially in datasets characterized by high informality and out-of-vocabulary

terms. In such cases, RS, being a meaning-independent approach, showed better performance by introducing syntactic variations without altering semantic content. Our comparison using state-of-the-art classification models, including BanglaBERT and variants of SVM, reveals consistent improvements in baseline performance when augmented data is used. This underscores the efficacy of our augmentation framework in enhancing model training for Bengali NLP tasks.

Despite the successes, our research highlights that augmentation is not universally beneficial. In cases like auto vocabulary tasks referenced in our VITD paper, augmentation failed to yield significant improvements. This suggests that the efficacy of augmentation techniques is context-dependent and varies with the nature and quality of the dataset. Moreover, in datasets with significant noise and imbalance, augmentation needs to be carefully managed to avoid introducing further inconsistencies.

Chapter 7

Conclusion and Future Works

We developed an augmentation framework, BDA for the Bangla language to address the challenges inherent in processing a low-resource language like Bangla. We have proposed and evaluated a novel set of text augmentation techniques, leveraging State-of-the-Art (SOTA) models, to enhance the performance and robustness of Bangla NLP models. Our results outperformed the baseline (non-BDA optimized dataset) of five different datasets by generating high-quality synthetic samples and increasing the size of the dataset. We incorporated a combination of transformer-based and rule-based approaches for generating a wide array of synthetic samples of semantically equivalent variations. Through rigorous experimentation and evaluation, we have determined the optimal settings and parameters for our augmentation techniques. This has led to a framework that not only maintains the semantic integrity of the original text but also introduces the necessary variability and complexity required for robust NLP models. We are optimistic that the BDA framework can significantly contribute to the advancement of Bangla Natural Language Processing research to mitigate data scarcity issues in the Bangla language and enhance the robustness of models in low-resource settings. However, some limitations such as label flipping on datasets with very similar characteristics among classes, require further exploration.

Limitations

A limitation of this framework is its reduced effectiveness in augmenting noisy texts containing spelling errors or out-of-vocabulary (OOV) words. The exception is the Random Swap (RS) method, which remains unaffected by such noise due to its meaning-independent nature; it simply swaps word indexes without semantic consideration. In contrast, Synonym Replacement (SR), Back Translation (BT), and Paraphrasing (PP) require some level of semantic understanding, thus hindering their performance in noisy texts.

Informal word structures, such as the growing trend of Banglish (Bangla written in English characters) on social media, also pose a challenge. Back-translation, Paraphrasing, and Synonym Replacement within BDA do not apply to such datasets, necessitating the development of specialized methods. Currently, BDA can only augment these texts through Random Swapping.

Future Work

One of the primary issues that needs to be addressed is that we need a better model to check the similarity of the sentences. Even though the Sentence-BERT model by Deode et al. [53] does a decent job, but still, a better sentence similarity model can potentially have a better outcome as the results of the VITD dataset suffer from label flipping as shown in Figure 6.1, not prevented via filtering. There are several avenues for further research and development:

Privacy Preserving Augmentation

Future work will involve exploring methods for privacy-preserving data augmentation. This is especially pertinent given the increasing concerns regarding data privacy. Developing techniques that can augment data without compromising individual privacy will be a significant step forward in the field.

Bengali-Optimized Language Model Implementation

Another area of future research is the implementation of Bengali-Optimized Large Language Models (LLMs). These models, designed specifically for the Bengali language, will potentially offer greater accuracy and efficiency in generating synthetic augmented texts, thereby enhancing the overall effectiveness of the augmentation framework.

Domain-Specific Augmentation

The domain-specific augmentation of text data, particularly in specialized fields such as medicine and law, is another promising area of research. Tailoring augmentation techniques to suit the linguistic nuances and terminologies of these domains will be crucial in expanding the applicability and relevance of Bangla NLP models.

Intergration of efficient LLMs

Efficient, LLM-based models for Bangla can improve the quality of the generated synthetic texts as seen by the Back-Translation and Paraphrasing method since we can perform sequence-to-sequence generation, which has a higher chance of changing the structure of the sentence. Creating augmentation techniques tailored to specific domains like medical or legal documents can also be a potential area to explore as BDA does not specialize in these use cases.

Appendix A

Detailed Results from Test-bench

The table A.2 gives the in-depth performance of the BDA framework across the discussed 5 datasets, on 3 clipping ranges (15%, 50%, and 100%). We used BanglaBERT and variants of SVM as models for evaluation. The F1 scores are reported in order to verify the actual improvement in the robustness of the model, as it only improves if the model is truly getting better at identifying all classes. The tables A.3-A.7 demonstrate the F1 performance after augmentation using BDA using each method: Synonym Replacement, Random Swap, Back-Translation, and Paraphrasing on all the datasets.

Model	Syn. Replacement			Random Swap			Back-translation			Paraphrasing		
	15%	50%	100%	15%	50%	100%	15%	50%	100%	15%	50%	100%
BanglaBERT	49.11	57.13	60.39	49.36	57.76	61.06	49.45	57.83	61.33	46.22	57.93	60.75
Unigram (U)	39.94	45.85	50.08	39.57	45.43	49.04	41.28	46.26	49.15	38.80	46.89	49.08
Bigram (B)	33.16	39.25	44.28	33.10	38.89	43.59	33.58	40.25	44.89	32.82	39.69	44.39
Trigram (T)	23.46	29.68	35.18	23.60	31.27	35.28	23.66	30.22	36.13	23.66	30.05	35.73
U+B	39.68	45.44	49.25	39.86	45.30	50.61	40.75	45.81	50.02	38.55	45.37	49.08
B+T	32.34	38.48	43.56	32.67	38.47	43.04	33.36	39.22	44.40	35.77	39.60	44.26
U+B+T	40.03	45.76	50.93	40.18	45.76	51.34	41.22	47.48	51.44	39.41	45.91	50.23
Char 2-gram (C2)	39.63	46.01	47.54	39.20	45.90	48.72	40.59	45.47	48.05	38.74	45.66	48.04
Char 3-gram (C3)	40.90	46.92	50.13	40.34	46.89	50.95	41.70	47.79	50.41	40.39	47.06	50.50
Char 4-gram (C4)	39.43	46.53	50.57	40.13	46.21	51.50	40.23	46.79	50.95	39.01	46.72	50.25
Char 5-gram (C5)	38.47	45.32	49.41	38.62	45.23	50.37	39.39	46.14	50.91	37.87	45.93	49.48
C2+C3	40.76	47.68	50.09	39.96	47.19	50.91	41.77	47.52	50.79	39.79	47.87	51.03
C3+C4	39.94	47.57	51.11	40.25	47.48	51.71	40.92	47.58	51.69	39.49	47.89	50.82
C4+C5	39.27	46.44	50.42	39.46	45.96	51.19	40.13	46.74	51.34	38.82	46.99	49.81
C2+C3+C4	40.20	47.62	50.97	39.94	47.50	51.04	41.18	47.80	51.99	39.49	47.72	51.31
C3+C4+C5	39.26	47.33	51.41	39.67	46.95	51.56	40.36	47.66	51.62	39.12	47.95	51.19
C2+C3+C4+C5	39.60	47.21	51.23	39.88	47.06	51.42	40.87	47.86	51.63	39.27	48.30	51.38
U+B+C3+C4+C5	39.84	46.75	51.20	39.97	46.30	51.92	40.85	47.60	51.96	40.01	47.12	51.14
U+B+C2+C3+C4+C5	40.01	46.98	51.14	40.26	46.48	51.84	41.48	47.78	51.91	40.02	47.86	51.41
U+B+T+C2+C3+C4+C5	39.66	47.02	51.64	40.10	46.66	52.30	40.94	47.32	51.65	39.83	47.59	52.06
Embeddings (E)	42.39	45.37	45.75	42.59	45.22	47.42	43.42	44.43	46.37	42.61	45.48	46.03
U+B+C2+C3+C4+C5+E	41.19	48.53	52.52	41.14	48.64	53.08	42.80	48.43	52.79	41.48	49.17	52.75
U+B+T+C2+C3+C4+C5+E	41.03	48.67	52.81	40.88	49.00	53.22	42.43	48.42	53.05	41.60	49.08	53.15

Table A.1: Average F1 scores of all datasets for Each method

Model	SentNoB			BemoC			Bengali Sentiment			ABSA Cricket			ABSA Restaurant		
	15%	50%	100%	15%	50%	100%	15%	50%	100%	15%	50%	100%	15%	50%	100%
BanglaBERT	64.97	68.10	72.04	61.94	69.39	70.55	47.74	49.19	49.08	41.90	56.05	57.45	27.98	45.58	55.28
Unigram (U)	54.49	60.93	66.61	42.48	50.26	54.66	37.43	38.96	40.04	37.71	46.92	50.77	27.38	33.49	34.59
Bigram (B)	46.79	54.19	61.65	29.81	38.00	42.74	34.41	36.78	37.99	31.75	43.12	47.21	23.06	25.50	31.86
Trigram (T)	31.66	43.70	54.06	11.54	21.69	25.49	25.79	28.44	30.42	30.37	37.77	42.40	18.61	19.93	25.53
U+B	55.93	60.90	67.71	43.21	51.27	55.82	37.05	37.87	39.40	35.81	44.96	50.50	26.55	32.40	35.29
B+T	46.86	54.24	62.03	29.85	37.72	43.22	33.42	35.70	36.53	34.75	42.48	46.28	22.82	24.57	31.03
U+B+T	56.12	60.77	67.19	43.25	51.08	55.94	37.08	38.33	39.97	37.19	46.80	52.29	27.42	34.18	39.55
Char 2-gram (C2)	52.75	57.99	61.46	39.78	47.96	50.90	37.78	38.19	39.03	38.11	49.03	51.63	29.28	35.63	37.42
Char 3-gram (C3)	56.83	60.77	66.44	45.10	52.69	55.73	39.08	39.77	40.48	37.35	48.60	52.08	25.80	34.00	37.74
Char 4-gram (C4)	56.14	61.35	67.63	44.98	53.90	57.61	38.42	40.08	40.99	34.92	46.93	51.19	24.03	30.55	36.65
Char 5-gram (C5)	55.10	61.94	67.70	42.77	53.12	57.52	36.51	39.70	41.04	35.86	46.37	50.12	22.70	27.13	33.84
C2+C3	55.75	61.67	67.25	44.78	52.70	56.02	38.65	39.68	40.36	37.81	48.98	52.47	25.84	34.80	37.43
C3+C4	56.94	61.69	67.94	45.26	54.26	57.82	38.80	40.30	40.91	35.29	48.01	51.86	24.46	33.88	38.12
C4+C5	56.36	62.48	68.25	44.82	54.22	58.42	37.36	40.02	41.56	35.35	47.02	49.58	23.22	28.93	35.62
C2+C3+C4	56.24	62.01	68.47	45.45	54.26	57.78	38.52	40.09	40.97	35.75	48.17	52.03	25.05	33.79	37.36
C3+C4+C5	56.32	62.37	68.91	45.26	54.60	59.15	37.81	39.92	41.42	35.06	47.93	50.79	23.57	32.54	36.94
C2+C3+C4+C5	56.32	62.29	68.94	45.71	54.84	59.06	37.76	39.87	41.36	35.18	48.40	50.79	24.53	32.62	36.92
U+B+C3+C4+C5	54.42	61.92	67.99	45.57	53.83	56.71	37.31	39.52	40.98	36.08	47.00	52.75	27.47	32.46	39.34
U+B+C2+C3+C4+C5	54.87	62.11	68.20	45.78	53.62	56.62	37.42	39.64	40.82	36.22	47.60	52.55	27.92	33.42	39.68
U+B+T+C2+C3+C4+C5	55.01	62.10	68.50	45.89	54.06	57.24	37.40	39.25	41.20	35.61	47.92	52.48	26.75	32.42	40.14
Embeddings (E)	55.54	54.40	53.27	50.44	53.90	56.62	38.86	38.72	38.78	38.03	45.58	45.54	30.88	33.02	37.75
U+B+C2+C3+C4+C5+E	56.44	62.33	68.54	48.86	55.58	58.23	38.76	40.32	41.66	36.58	49.17	53.34	27.61	36.07	42.15
U+B+T+C2+C3+C4+C5+E	57.36	62.50	68.66	48.66	56.05	58.71	38.62	40.30	41.87	35.76	49.60	53.18	27.04	35.52	42.86

Table A.2: Table of average F1 scores for each dataset after using BDA

Model	Syn. Replacement			Random Swap			Back-translation			Paraphrasing		
	15%	50%	100%	15%	50%	100%	15%	50%	100%	15%	50%	100%
BanglaBERT	63.86	67.31	70.45	65.40	68.49	71.31	65.40	68.39	72.04	65.21	68.22	74.33
Unigram (U)	55.24	61.93	67.30	53.62	60.11	66.81	55.22	60.61	66.34	53.87	61.05	65.99
Bigram (B)	47.21	54.16	62.01	46.39	53.95	61.89	47.19	54.76	61.70	46.35	53.90	61.01
Trigram (T)	31.41	43.58	53.58	32.30	43.67	54.44	31.40	43.80	54.20	31.53	43.74	54.01
U+B	56.21	61.04	67.10	55.59	60.43	68.82	56.23	60.84	66.93	55.72	61.29	67.99
B+T	46.80	53.46	62.41	47.20	54.29	62.35	46.79	54.64	61.68	46.62	54.55	61.68
U+B+T	56.35	60.83	66.94	56.10	60.77	68.06	56.34	60.63	66.77	55.68	60.84	67.01
Char 2-gram (C2)	53.51	57.33	61.26	51.85	57.55	62.51	53.4	59.76	61.06	52.14	57.33	61.00
Char 3-gram (C3)	57.42	60.68	65.91	55.93	60.72	66.98	57.44	61.39	65.89	56.54	60.29	66.98
Char 4-gram (C4)	55.66	61.08	67.84	57.11	60.46	67.87	55.67	61.89	66.82	56.15	61.96	67.99
Char 5-gram (C5)	55.19	61.60	67.85	55.06	61.25	68.19	55.17	62.39	67.78	54.96	62.53	67.01
C2+C3	56.40	61.61	66.87	54.38	61.62	67.66	56.41	61.83	66.48	55.81	61.61	67.99
C3+C4	56.86	61.43	68.03	57.22	61.57	68.18	56.66	61.80	67.57	56.81	61.96	68.01
C4+C5	56.55	62.68	68.61	56.45	62.26	68.39	56.65	62.41	68.00	55.87	62.55	67.99
C2+C3+C4	56.02	61.81	68.02	56.47	61.09	68.28	56.01	62.73	68.58	56.44	62.40	68.99
C3+C4+C5	56.01	62.63	69.39	57.03	62.16	69.24	56.01	62.16	68.03	56.21	62.54	68.98
C2+C3+C4+C5	56.52	62.41	69.19	56.12	61.42	69.14	56.54	62.10	68.45	56.13	63.23	69.02
U+B+C3+C4+C5	53.96	61.21	67.85	55.73	61.62	68.70	53.99	62.37	67.40	54.04	62.48	67.99
U+B+C2+C3+C4+C5	54.87	61.46	68.23	55.35	61.82	68.74	54.85	62.51	67.85	54.38	62.64	67.99
U+B+T+C2+C3+C4+C5	54.91	61.39	68.67	55.17	61.86	68.84	54.92	62.36	67.48	55.05	62.77	69.01
Embeddings (E)	55.67	54.86	55.18	56.14	54.94	55.86	53.88	53.65	54.05	56.49	54.17	48.01
U+B+C2+C3+C4+C5+E	56.06	61.96	68.40	56.30	61.99	69.30	56.61	62.50	67.45	56.79	62.87	68.99
U+B+T+C2+C3+C4+C5+E	56.77	62.02	69.08	57.35	62.14	68.71	57.65	62.80	67.85	57.65	63.02	68.99

Table A.3: F1 scores of SentNoB dataset for Each method

Model	Syn. Replacement			Random Swap			Back-translation			Paraphrasing		
	15%	50%	100%	15%	50%	100%	15%	50%	100%	15%	50%	100%
BanglaBERT	60.99	69.01	70.47	64.11	69.61	70.61	60.87	69.63	70.62	61.78	69.31	70.52
Unigram (U)	42.16	49.70	53.87	41.92	49.75	54.20	45.02	50.51	55.14	40.82	51.08	55.44
Bigram (B)	29.25	36.24	41.50	30.04	37.64	42.41	30.02	38.76	41.90	29.94	39.38	45.15
Trigram (T)	11.23	18.67	24.07	10.85	27.07	24.44	12.04	20.26	26.37	12.03	20.77	27.09
U+B	42.66	50.70	55.32	43.06	51.43	56.44	44.68	51.91	55.67	42.45	51.04	55.83
B+T	29.49	36.40	41.97	29.25	36.75	42.55	30.34	38.09	43.02	30.33	39.64	45.36
U+B+T	42.42	50.13	55.47	42.74	51.01	55.81	45.17	52.03	56.15	42.67	51.13	56.31
Char 2-gram (C2)	38.43	48.45	50.97	39.06	47.90	49.86	41.57	47.79	51.49	40.05	47.70	51.29
Char 3-gram (C3)	44.37	52.79	55.24	45.23	51.71	55.09	45.69	53.58	56.40	45.11	52.68	56.18
Char 4-gram (C4)	45.00	54.07	58.18	44.94	54.63	58.09	45.64	53.81	56.84	44.34	53.07	57.34
Char 5-gram (C5)	42.38	52.71	57.08	43.64	53.48	57.12	42.33	53.38	57.62	42.74	52.93	58.24
C2+C3	43.80	53.55	55.59	44.39	52.00	55.53	46.06	52.93	56.84	44.89	52.34	56.11
C3+C4	44.88	54.17	57.27	45.16	54.69	58.18	45.94	54.37	58.57	45.06	53.80	57.27
C4+C5	44.18	54.07	58.60	44.80	54.59	58.35	45.74	54.24	58.18	44.57	53.98	58.57
C2+C3+C4	45.20	54.40	57.33	45.06	54.75	57.26	46.38	54.34	58.96	45.17	53.55	57.59
C3+C4+C5	45.03	54.22	59.06	45.00	54.35	58.78	45.25	55.06	59.31	45.75	54.77	59.46
C2+C3+C4+C5	44.77	54.62	58.61	46.21	55.10	59.07	46.07	54.91	59.33	45.80	54.73	59.25
U+B+C3+C4+C5	45.04	53.76	56.28	45.45	52.77	56.78	45.94	54.77	56.85	45.84	54.02	56.92
U+B+C2+C3+C4+C5	45.07	53.23	56.00	46.11	52.28	56.92	45.80	55.00	56.79	46.14	53.97	56.78
U+B+T+C2+C3+C4+C5	44.91	54.06	56.99	46.09	52.91	57.33	46.05	54.86	56.67	46.52	54.43	57.98
Embeddings (E)	49.40	54.42	56.15	50.23	53.43	55.97	51.64	53.04	56.66	50.49	54.72	57.70
U+B+C2+C3+C4+C5+E	48.44	55.57	57.73	47.81	55.44	58.53	50.18	56.00	58.39	49.03	55.29	58.26
U+B+T+C2+C3+C4+C5+E	47.93	55.73	58.10	47.61	56.36	59.04	49.77	56.27	58.87	49.31	55.83	58.84

Table A.4: F1 scores of BemoC dataset for Each method

Model	Syn. Replacement			Random Swap			Back-translation			Paraphrasing		
	15%	50%	100%	15%	50%	100%	15%	50%	100%	15%	50%	100%
BanglaBERT	48.58	49.20	49.59	47.93	50.28	49.32	37.68	48.51	48.70	48.00	48.76	48.79
Unigram (U)	37.73	38.14	39.38	37.72	38.62	39.92	34.52	39.87	39.87	36.57	39.22	39.61
Bigram (B)	34.20	36.38	36.97	34.49	36.66	37.73	25.90	37.34	37.34	34.43	36.74	37.46
Trigram (T)	25.66	28.21	30.32	25.86	28.08	30.32	37.11	29.78	29.78	25.75	27.68	29.89
U+B	37.12	37.69	38.91	37.33	37.25	38.69	33.50	39.29	39.29	36.63	37.26	38.98
B+T	32.93	35.74	35.52	33.44	35.60	36.49	37.28	35.78	35.78	33.80	35.68	35.54
U+B+T	36.74	37.78	39.32	37.20	38.54	39.37	36.87	39.39	39.39	37.08	37.60	39.65
Char 2-gram (C2)	38.63	37.99	38.95	38.01	37.71	38.14	39.03	38.38	38.38	37.60	38.68	38.39
Char 3-gram (C3)	39.31	39.20	40.34	39.47	39.65	40.05	38.61	40.49	40.49	38.50	39.73	40.12
Char 4-gram (C4)	38.12	39.55	41.02	38.98	39.59	41.40	36.97	40.94	40.94	37.98	40.26	40.59
Char 5-gram (C5)	36.54	38.56	40.35	36.53	39.75	40.39	39.11	40.91	40.91	36.01	39.58	40.68
C2+C3	39.20	39.46	40.59	38.89	39.10	40.07	38.96	40.40	40.40	37.41	39.74	39.98
C3+C4	38.64	40.00	40.24	39.41	39.73	40.91	37.72	40.50	40.50	38.20	40.99	40.53
C4+C5	37.33	39.41	41.06	37.22	39.85	40.43	39.00	40.78	40.78	37.16	40.05	41.10
C2+C3+C4	38.53	39.54	39.98	38.87	39.82	40.81	38.27	40.67	40.67	37.66	40.31	40.36
C3+C4+C5	37.80	39.31	40.98	38.18	39.31	41.13	38.21	40.86	40.86	37.00	40.18	40.97
C2+C3+C4+C5	37.85	38.98	40.83	38.04	39.28	40.79	36.92	40.92	40.92	36.95	40.31	40.96
U+B+C3+C4+C5	37.52	38.70	40.61	37.17	38.72	40.74	37.36	41.22	41.22	37.63	39.43	40.71
U+B+C2+C3+C4+C5	37.47	39.41	40.35	37.12	38.83	40.54	37.32	41.29	41.29	37.75	39.03	40.51
U+B+T+C2+C3+C4+C5	37.20	38.91	40.67	37.47	38.69	40.84	38.90	40.16	41.64	37.62	39.24	40.98
Embeddings (E)	38.57	39.14	38.87	39.13	38.98	38.18	39.11	38.19	37.13	38.85	38.59	37.85
U+B+C2+C3+C4+C5+E	38.48	40.16	41.15	38.49	40.01	40.55	38.77	40.96	41.39	38.98	40.14	41.26
U+B+T+C2+C3+C4+C5+E	38.37	40.19	40.90	38.40	40.12	41.00	36.90	41.13	41.71	38.94	39.77	41.51

Table A.5: F1 scores of Bengali Sentiment for Each method

Model	Syn. Replacement			Random Swap			Back-translation			Paraphrasing		
	15%	50%	100%	15%	50%	100%	15%	50%	100%	15%	50%	100%
BanglaBERT	45.72	55.85	57.28	45.57	56.01	59.78	48.32	56.72	57.90	48.32	55.60	54.83
Unigram (U)	37.42	47.27	51.69	37.42	46.37	49.95	39.53	47.05	52.10	39.53	46.97	49.35
Bigram (B)	32.05	43.24	48.14	32.05	44.41	47.26	31.60	41.82	46.55	31.60	43.01	46.87
Trigram (T)	30.37	38.04	42.54	30.37	37.62	42.36	30.37	37.29	42.36	30.37	38.13	42.36
U+B	36.15	44.96	49.53	36.15	45.11	52.15	36.92	45.74	51.68	34.01	44.02	48.62
B+T	30.37	43.50	46.75	30.37	43.16	46.17	31.64	40.63	46.31	46.62	42.64	45.87
U+B+T	37.32	45.18	52.61	37.32	46.64	53.26	38.82	49.46	52.06	35.29	45.90	51.24
Char 2-gram (C2)	38.47	50.03	50.75	38.47	50.58	53.32	39.28	46.80	50.35	36.23	48.70	52.09
Char 3-gram (C3)	37.09	47.72	52.19	37.09	47.60	53.10	38.15	48.40	52.12	37.07	50.70	50.93
Char 4-gram (C4)	34.41	47.11	48.77	34.41	45.59	53.22	35.97	46.90	53.82	34.91	48.12	48.96
Char 5-gram (C5)	35.91	46.42	49.07	35.91	44.66	50.53	38.02	47.13	52.12	33.60	47.28	48.76
C2+C3	38.04	48.51	51.86	38.04	48.83	52.94	37.97	47.77	53.02	37.21	50.81	52.06
C3+C4	35.01	46.52	51.42	35.01	47.12	52.52	36.70	48.27	53.57	34.45	50.14	49.93
C4+C5	35.04	46.78	47.84	35.04	44.75	50.53	36.71	47.62	52.15	34.63	48.92	47.81
C2+C3+C4	35.38	47.71	51.59	35.38	47.38	52.36	37.43	48.19	52.84	34.83	49.40	51.32
C3+C4+C5	34.35	47.04	49.39	34.35	46.92	50.72	37.09	47.77	53.21	34.45	50.00	49.84
C2+C3+C4+C5	34.77	47.37	49.39	34.77	46.83	51.00	36.74	49.34	52.18	35.26	50.07	50.58
U+B+C3+C4+C5	35.36	47.57	52.96	35.36	45.38	53.16	38.33	47.60	53.26	34.96	47.45	51.62
U+B+C2+C3+C4+C5	35.36	47.92	52.88	35.45	45.54	52.22	39.20	48.05	52.97	34.16	48.87	52.12
U+B+T+C2+C3+C4+C5	35.45	48.07	52.76	36.94	46.09	53.24	37.39	48.73	52.27	37.63	48.77	51.65
Embeddings (E)	36.94	45.96	43.84	36.07	44.81	47.81	40.63	45.81	43.36	35.84	45.75	47.13
U+B+C2+C3+C4+C5+E	36.07	49.19	53.63	35.11	50.42	52.90	38.35	48.56	53.09	35.91	48.52	53.75
U+B+T+C2+C3+C4+C5+E	35.11	49.23	53.02	35.11	50.60	53.25	36.90	48.54	53.07	35.91	50.04	53.39

Table A.6: F1 scores of ABSA Cricket for Each method

Model	Syn. Replacement			Random Swap			Back-translation			Paraphrasing		
	15%	50%	100%	15%	50%	100%	15%	50%	100%	15%	50%	100%
BanglaBERT	26.42	44.30	54.14	23.78	44.39	54.35	26.22	45.91	57.37	28.15	47.74	55.28
Unigram (U)	27.16	32.23	37.92	27.17	32.32	34.61	28.94	33.27	30.82	26.24	36.15	35.00
Bigram (B)	23.08	26.22	32.31	22.53	21.80	28.94	24.55	28.57	34.71	22.08	25.41	31.48
Trigram (T)	18.63	19.92	25.81	18.60	19.90	25.29	18.60	19.95	25.73	18.60	19.93	25.29
U+B	26.25	32.80	35.34	27.18	32.27	36.68	28.82	31.27	35.14	23.96	33.26	33.99
B+T	22.13	23.32	31.13	23.11	22.54	28.61	24.53	26.95	31.51	21.49	25.47	32.85
U+B+T	27.34	34.87	40.00	27.53	31.86	39.93	28.49	35.90	41.30	26.32	34.09	36.95
Char 2-gram (C2)	29.11	36.27	36.33	28.60	35.76	39.51	31.72	34.61	36.39	27.68	35.89	37.45
Char 3-gram (C3)	26.30	34.23	37.18	23.97	34.78	39.44	28.21	35.08	36.09	24.72	31.91	38.26
Char 4-gram (C4)	23.98	30.84	37.45	25.19	30.76	37.74	25.29	30.39	35.06	21.67	30.21	36.34
Char 5-gram (C5)	22.35	27.33	32.36	21.96	26.99	35.34	24.42	26.87	34.92	22.06	27.32	32.72
C2+C3	26.34	35.26	36.14	24.08	34.40	38.45	29.33	34.66	36.12	23.63	34.87	39.02
C3+C4	24.33	35.72	38.32	24.46	34.31	39.15	26.12	32.94	36.67	22.93	32.55	38.36
C4+C5	23.27	29.27	35.93	23.80	28.35	37.57	23.94	28.67	35.41	21.85	29.44	33.56
C2+C3+C4	25.89	34.66	37.53	23.90	34.45	36.92	27.06	33.09	36.74	23.35	32.94	38.26
C3+C4+C5	23.11	33.46	38.23	23.79	31.99	38.07	25.20	32.47	34.79	22.19	32.25	36.66
C2+C3+C4+C5	24.07	32.66	37.99	24.27	32.66	36.95	26.79	32.02	35.66	23.00	33.14	37.09
U+B+C3+C4+C5	27.33	32.53	38.20	26.14	32.66	40.25	29.10	32.03	40.49	27.30	32.23	38.44
U+B+C2+C3+C4+C5	27.26	32.86	38.07	27.37	32.86	40.83	30.16	32.06	40.17	26.89	34.81	39.63
U+B+T+C2+C3+C4+C5	25.84	32.68	38.79	26.33	33.77	41.13	29.04	30.47	39.97	25.78	32.74	40.68
Embeddings (E)	31.38	32.48	35.74	30.51	33.94	39.61	32.06	31.47	36.20	29.59	34.18	39.47
U+B+C2+C3+C4+C5+E	26.88	35.78	41.56	27.05	35.33	43.40	29.75	34.13	42.16	26.78	39.02	41.47
U+B+T+C2+C3+C4+C5+E	26.98	36.18	42.32	25.92	35.76	43.61	29.08	33.37	42.48	26.18	36.75	43.01

Table A.7: F1 scores of ABSA Restaurant for Each method

Model	Normal, T			Augmented, T'		
	15%	50%	100%	15%	50%	100%
BanglaBERT	40.34	51.84	58.30	48.54	57.66	60.88
Unigram (U)	39.26	45.46	47.60	39.90	46.11	49.34
Bigram (B)	32.57	38.86	41.37	33.16	39.52	44.29
Trigram (T)	23.34	28.88	33.67	23.60	30.30	35.58
U+B	38.52	43.20	46.71	39.71	45.48	49.74
B+T	32.29	39.36	42.92	33.54	38.94	43.82
U+B+T	39.42	45.12	47.95	40.21	46.23	50.98
Char 2-gram (C2)	38.52	44.91	46.50	39.54	45.76	48.09
Char 3-gram (C3)	40.34	47.20	48.09	40.83	47.16	50.50
Char 4-gram (C4)	38.92	46.28	48.69	39.70	46.56	50.82
Char 5-gram (C5)	37.64	44.54	47.75	38.59	45.66	50.04
C2+C3	39.96	47.21	48.47	40.57	47.56	50.70
C3+C4	39.37	46.88	49.01	40.15	47.63	51.33
C4+C5	39.07	45.68	48.20	39.42	46.53	50.69
C2+C3+C4	39.58	47.00	48.64	40.20	47.66	51.33
C2+C3+C4+C5	39.42	46.69	48.81	39.60	47.47	51.44
U+B+C3+C4+C5	39.64	46.27	49.18	39.90	47.61	51.42
U+B+T+C2+C3+C4+C5	39.50	46.51	49.78	40.17	46.94	51.55
Embeddings (E)	42.59	45.08	47.53	40.44	47.28	51.57
U+B+C2+C3+C4+C5+E	41.01	48.19	51.82	41.65	48.69	52.78
U+B+T+C2+C3+C4+C5+E	41.06	48.29	52.07	41.48	48.79	53.06

Table A.8: Comparison of F1 scores across all five datasets between Normal, T, and BDA augmented T' datasets

Bibliography

[1] M. Kabir, O. Bin Mahfuz, S. R. Raiyan, H. Mahmud, and M. K. Hasan, "BanglaBook: A large-scale Bangla dataset for sentiment analysis from book reviews," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 1237–1247. [Online]. Available: https://aclanthology.org/2023.findings-acl.80

[2] K. D. Dhole, V. Gangal, S. Gehrmann, A. Gupta, Z. Li, S. Mahamood, A. Mahendiran, S. Mille, A. Srivastava, S. Tan, T. Wu, J. Sohl-Dickstein, J. D. Choi, E. H. Hovy, O. Dusek, S. Ruder, S. Anand, N. Aneja, R. Banjade, L. Barthe, H. Behnke, I. Berlot-Attwell, C. Boyle, C. Brun, M. A. S. Cabezudo, S. Cahyawijaya, E. Chapuis, W. Che, M. Choudhary, C. Clauss, P. Colombo, F. Cornell, G. Dagan, M. Das, T. Dixit, T. Dopierre, P. Dray, S. Dubey, T. Ekeinhor, M. D. Giovanni, R. Gupta, R. Gupta, L. Hamla, S. Han, F. Harel-Canada, A. Honore, I. Jindal, P. K. Joniak, D. Kleyko, V. Kovatchev, and et al., "Nl-augmenter: A framework for task-sensitive natural language augmentation," *CoRR*, vol. abs/2112.02721, 2021. [Online]. Available: https://arxiv.org/abs/2112.02721

[3] O. Sen, M. Fuad, M. N. Islam, J. Rabbi, M. Masud, M. K. Hasan, M. A. Awal, A. Ahmed Fime, M. T. Hasan Fuad, D. Sikder, and M. A. Raihan Iftee, "Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning-based methods," *IEEE Access*, vol. 10, p. 38999–39044, 2022. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2022.3165563

[4] T. Mohiuddin, M. S. Bari, and S. Joty, "AugVic: Exploiting BiText vicinity for low-resource NMT," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 3034–3045. [Online]. Available: https://aclanthology.org/2021.findings-acl.267

[5] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. H. Hovy, "A survey of data augmentation approaches for NLP," *CoRR*, vol. abs/2105.03075, 2021. [Online]. Available: https://arxiv.org/abs/2105.03075

[6] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for nlp," *arXiv preprint arXiv:2105.03075*, 2021.

[7] M. Tareq, M. F. Islam, S. Deb, S. Rahman, and A. Al Mahmud, "Data-augmentation for bangla-english code-mixed sentiment analysis: Enhancing cross linguistic contextual understanding," *IEEE Access*, 2023.

[8] A. J. Keya, M. A. H. Wadud, M. Mridha, M. Alatiyyah, and M. A. Hamid, "Augfake-bert: handling imbalance through augmentation of fake news using bert to enhance the performance of fake news classification," *Applied Sciences*, vol. 12, no. 17, p. 8398, 2022.

[9] N. R. Bhowmik, M. Arifuzzaman, and M. R. H. Mondal, "Sentiment analysis on bangla text using extended lexicon dictionary and deep learning algorithms," *Array*, vol. 13, p. 100123, 2022.

[10] P. Simard, Y. LeCun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition-tangent distance and tangent propagation," in *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*. Berlin, Heidelberg: Springer-Verlag, 1998, p. 239–27.

[11] B. Li, Y. Hou, and W. Che, "Data augmentation approaches in natural language processing: A survey," *CoRR*, vol. abs/2110.01852, 2021. [Online]. Available: https://arxiv.org/abs/2110.01852

[12] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[13] J. Wei, C. Huang, S. Xu, and S. Vosoughi, "Text augmentation in a multi-task view," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds. Online: Association for Computational Linguistics, Apr. 2021, pp. 2888–2894. [Online]. Available: https://aclanthology.org/2021.eacl-main.252

[14] Y. Li, T. Cohn, and T. Baldwin, "Robust training under linguistic adversity," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 21–27. [Online]. Available: https://aclanthology.org/E17-2004

[15] O. Kashefi and R. Hwa, "Quantifying the evaluation of heuristic methods for textual data augmentation," in *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, W. Xu, A. Ritter, T. Baldwin, and A. Rahimi, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 200–208. [Online]. Available: https://aclanthology.org/2020.wnut-1.26

[16] B. Hariharan and R. B. Girshick, "Low-shot visual object recognition," *CoRR*, vol. abs/1606.02819, 2016. [Online]. Available: http://arxiv.org/abs/1606.02819

[17] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, A. Kumar, R. Feris, R. Giryes, and A. M. Bronstein, "Δ-encoder: an effective sample synthesis method for few-shot object recognition," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 2850–2860.

[18] M. Paschali, W. Simson, A. G. Roy, M. F. Naeem, R. Göbl, C. Wachinger, and N. Navab, "Data augmentation with manifold exploring geometric transformations

for increased performance and robustness," *CoRR*, vol. abs/1901.04420, 2019. [Online]. Available: http://arxiv.org/abs/1901.04420

[19] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6256–6268. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/44feb0096faa8326192570788b38c1d1-Paper.pdf

[20] H. Chen, Y. Ji, and D. Evans, "Finding Friends and flipping frenemies: Automatic paraphrase dataset augmentation using graph theory," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 4741–4751. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.426

[21] G. G. Şahin and M. Steedman, "Data augmentation via dependency tree morphing for low-resource languages," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 5004–5009. [Online]. Available: https://aclanthology.org/D18-1545

[22] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 86–96. [Online]. Available: https://aclanthology.org/P16-1009

[23] A. Kumar, S. Bhattamishra, M. Bhandari, and P. Talukdar, "Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation," in *Proceedings of the 2019 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3609–3619. [Online]. Available: https://aclanthology.org/N19-1363

[24] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, M. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 452–457. [Online]. Available: https://aclanthology.org/N18-2072

[25] Y. Yang, C. Malaviya, J. Fernandez, S. Swayamdipta, R. Le Bras, J.-P. Wang, C. Bhagavatula, Y. Choi, and D. Downey, "Generative data augmentation for commonsense reasoning," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 1008–1025. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.90

[26] F. Gao, J. Zhu, L. Wu, Y. Xia, T. Qin, X. Cheng, W. Zhou, and T.-Y. Liu, "Soft contextual data augmentation for neural machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5539–5544. [Online]. Available: https://aclanthology.org/P19-1555

[27] Y. Nie, Y. Tian, X. Wan, Y. Song, and B. Dai, "Named entity recognition for social media texts with semantic augmentation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 1383–1391. [Online]. Available: https://aclanthology.org/2020.emnlp-main.107

[28] N. Ng, K. Cho, and M. Ghassemi, "SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 1268–1283. [Online]. Available: https://aclanthology.org/2020.emnlp-main.97

[29] S. Y. Feng, A. W. Li, and J. Hoey, "Keep calm and switch on! preserving sentiment and fluency in semantic text exchange," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2701–2711. [Online]. Available: https://aclanthology.org/D19-1272

[30] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling, "Not enough data? deep learning to the rescue!" *CoRR*, vol. abs/1911.03118, 2019. [Online]. Available: http://arxiv.org/abs/1911.03118

[31] H. Quteineh, S. Samothrakis, and R. Sutcliffe, "Textual data augmentation for efficient active learning on tiny datasets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 7400–7410. [Online]. Available: https://aclanthology.org/2020.emnlp-main.600

[32] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, "Adversarial example generation with syntactically controlled paraphrase networks," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, M. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana: Association

for Computational Linguistics, Jun. 2018, pp. 1875–1885. [Online]. Available: https://aclanthology.org/N18-1170

[33] V. Gangal, S. Y. Feng, E. H. Hovy, and T. Mitamura, "NAREOR: the narrative reordering problem," *CoRR*, vol. abs/2104.06669, 2021. [Online]. Available: https://arxiv.org/abs/2104.06669

[34] T. Dreossi, S. Ghosh, X. Yue, K. Keutzer, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Counterexample-guided data augmentation," *CoRR*, vol. abs/1805.06962, 2018. [Online]. Available: http://arxiv.org/abs/1805.06962

[35] N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych, "Augmented SBERT: data augmentation method for improving bi-encoders for pairwise sentence scoring tasks," *CoRR*, vol. abs/2010.08240, 2020. [Online]. Available: https://arxiv.org/abs/2010.08240

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

[37] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT sentence embedding," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 878–891. [Online]. Available: https://aclanthology.org/2022.acl-long.62

[38] A. Bhattacharjee, T. Hasan, W. U. Ahmad, and R. Shahriyar, "Banglanlg and banglat5: Benchmarks and resources for evaluating low-resource natural language generation in bangla," 2023.

[39] M. Tareq, M. F. Islam, S. Deb, S. Rahman, and A. A. Mahmud, "Data-augmentation for bangla-english code-mixed sentiment analysis: Enhancing cross linguistic contextual understanding," *IEEE Access*, vol. 11, pp. 51 657–51 671, 2023.

[40] H. T. Kesgin and M. F. Amasyali, *Iterative Mask Filling: An Effective Text Augmentation Method Using Masked Language Modeling*. Springer Nature Switzerland, Dec. 2023, p. 450–463. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-50920-9_35

[41] T. Hasan, A. Bhattacharjee, K. Samin, M. Hasan, M. Basak, M. S. Rahman, and R. Shahriyar, "Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 2612–2623. [Online]. Available: https://aclanthology.org/2020.emnlp-main.207

[42] A. Akil, N. Sultana, A. Bhattacharjee, and R. Shahriyar, "BanglaParaphrase: A high-quality Bangla paraphrase dataset," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Y. He, H. Ji, S. Li, Y. Liu, and C.-H. Chang, Eds. Online only: Association for Computational Linguistics, Nov. 2022, pp. 261–272. [Online]. Available: https://aclanthology.org/2022.aacl-short.33

[43] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6382–6388. [Online]. Available: https://aclanthology.org/D19-1670

[44] T. Hasan, A. Bhattacharjee, K. Samin, M. Hasan, M. Basak, M. S. Rahman, and R. Shahriyar, "Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*

(EMNLP). Online: Association for Computational Linguistics, Nov. 2020, pp. 2612–2623. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.207

[45] A. Bhattacharjee, T. Hasan, W. Ahmad, K. S. Mubasshir, M. S. Islam, A. Iqbal, M. S. Rahman, and R. Shahriyar, "BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla," in *Findings of the Association for Computational Linguistics: NAACL 2022*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 1318–1327. [Online]. Available: https://aclanthology.org/2022.findings-naacl.98

[46] K. I. Islam, S. Kar, M. S. Islam, and M. R. Amin, "SentNoB: A dataset for analysing sentiment on noisy Bangla texts," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3265–3271. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.278

[47] A. Baruah, K. Das, F. Barbhuiya, and K. Dey, "Aggression identification in English, Hindi and Bangla text using BERT, RoBERTa and SVM," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, R. Kumar, A. K. Ojha, B. Lahiri, M. Zampieri, S. Malmasi, V. Murdock, and D. Kadar, Eds. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 76–82. [Online]. Available: https://aclanthology.org/2020.trac-1.12

[48] A. Iqbal, A. Das, O. Sharif, M. Hoque, and I. Sarker, "Bemoc: A corpus for identifying emotion in bengali texts," *SN Computer Science*, vol. 3, 03 2022.

[49] M. Rahman and E. Dey, "Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation," *Data*, vol. 3, p. 15, 05 2018.

[50] K. I. Islam, M. S. Islam, and M. R. Amin, "Sentiment analysis in bengali via trans-

fer learning using multi-lingual bert," in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, 2020, pp. 1–5.

[51] S. Saha, J. A. Junaed, M. Saleki, M. Rahouti, N. Mohammed, and M. R. Amin, "BLP-2023 task 1: Violence inciting text detection (VITD)," in *Proceedings of the First Workshop on Bangla Language Processing (BLP-2023)*, F. Alam, S. Kar, S. A. Chowdhury, F. Sadeque, and R. Amin, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 255–265. [Online]. Available: https://aclanthology.org/2023.banglalp-1.33

[52] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 186–191. [Online]. Available: https://www.aclweb.org/anthology/W18-6319

[53] S. Deode, J. Gadre, A. Kajale, A. Joshi, and R. Joshi, "L3Cube-IndicSBERT: A simple approach for learning cross-lingual sentence representations using multilingual BERT," in *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, C.-R. Huang, Y. Harada, J.-B. Kim, S. Chen, Y.-Y. Hsu, E. Chersoni, P. A, W. H. Zeng, B. Peng, Y. Li, and J. Li, Eds. Hong Kong, China: Association for Computational Linguistics, Dec. 2023, pp. 154–163. [Online]. Available: https://aclanthology.org/2023.paclic-1.16