# Investigating the Use of Deep Learning for Textual Entailment in BRACU_NLI Dataset

by

FARAH BINTA HAQUE
24141090
MD YASIN
20301310
SHISHIR SAHA
20301320
MD MAZED HOSSAIN
21301569

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
January 2024

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

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

| | |
|---|---|
| Farah Binta Haque | MD Yasin |
| 24141090 | 20301310 |
| Shishir Saha | Md Mazed Hossain |
| 20301320 | 21301569 |

# Approval

The thesis/project titled "Investigating the Use of Deep Learning for Textual Entailment in BRACU_NLI Dataset " submitted by

1. Farah Binta Haque (24141090)
2. MD YASIN (20301310)
3. Shishir Saha (20301320)
4. MD MAZED HOSSAIN (21301569)

Of fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January, 2024.

**Examining Committee:**

Supervisor:
(Member)

Dr. Farig Yousuf Sadeque

Assistant Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

# Ethics Statement

So, we confirm that the information presented in this thesis is based only on our own research results. Each extra source used in this work has been undeniably acknowledged. Further, we also certify that this thesis and not one of its compositions have never been checked in or presented to get a degree from some other university or educational establishment.

# Abstract

This work aims to analyze the potential of deep neural models for text-based entailment in Bangla Language. Entailment is the method of determining whether one text infers or goes against another text. The study concentrates on the application of deep learning methods, such as Recurrent Neural Networks (RNNs), BERT, GPT for solving text-based entailment. The neural network method is trained to foretell the relationship between two text sequences, such as whether one text sequence entails the other or whether one text sequence provides evidence for the other. Other tasks, such as question answering, can also be tackled by fine-tuning these models on specific datasets. The findings of this work will contribute to the development of further developed NLP systems that can perform complex reasoning and entailment tasks.

**Keywords:** Deep Learning; Machine Learning; Text Entailment; Text Summarizing, Text Generation, Transformers

# Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Dr. Farig Yousuf Sadeque sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The application and implication of Natural Language Processing(NLP) have witnessed remarkable headways in recent times. Fueled by the fast advancement in Deep Learning methods, Neural Network models have delivered excellent undertakings across different NLP errands such as language translation, text prediction, sentiment analysis etc. Text Entailment is one of the core problems in NLP. Text Entailment refers to a logical connection between two bits of text. One is premises and another is hypothesis. Text Entailment plays a pivotal part in different segments of NLP such as information extraction, questioning answer, summarization, machine translation etc.
The undertaking of Text Entailment involves evaluating whether the meaning of the hypothesis can be logically deduced from the premise or not. It requires figuring out the semantic and contextual subtle distinction within the text and making competent assumptions. Conventional methods of Text Entailment depended extensively on rule-based frameworks. It frequently struggled to seize the intricacy and variability of natural language.
Deep learning has revolutionized the domain of NLP by empowering the advancement of end-to-end models. It can automatically gain significant depictions from data. Different Neural Network(NN) models such as CNN, RNN, BERT, and mBERT have shown phenomenal success in several NLP errands. Deep learning methods would be a great stake for tending to the challenges of Text Entailment by utilizing their capabilities to capture intricate patterns and connections between textual data.
In this work, we will examine and analyze different deep learning architectures, such as RNN, CNN and transformer-based models to conduct the Text Entailment case. We will examine different methods for representation learning, pre-training on large-scale corpora and fine-tuning entailment-specific data. Moreover, we will investigate the influence of several model structures, optimizing algorithms, and preparing systems on the performance of the entailment models.

## 1.1 Research Problem

With the advent of the Internet, people have become extremely communicative in recent years. Text, image, audio, and video are used in a lot of conversations and communications. Every day, this generates a lot of data. Automated systems are required to detect counterfeit or misinformation, and fraudulent claims, and

judge the integrity of textual content in social media, online news feeds, and tweets [18]. Given a subject of interest, a standard search frequently fetches innumerable information. A considerable lot of them are not to the user's benefit. Summarization tools save time by permitting readers to quickly decide whether to read a document [2]. Text Entailment is an asymmetric relation between two fragments of text that describe whether one part can be construed from the other[1].
Textual Entailment is used in various errands of NLP such as Machine Translation, Text Summarization, and Question Answering. Good research in Textual Entailment will benefit other NLP fields. Numerous NLP applications, including sentiment analysis, machine translation, and others, are benefiting from the application of deep learning techniques. It has also begun to be used in Textual Entailment (TE), where various methods based on neural networks have recently been developed [20].
The logical relationship between a premise and a hypothesis can be difficult to determine due to the Text Entailment issue. Conventional approaches to Text Entailment relied on handcrafted features and rule-based frameworks, which frequently struggled to capture the intricacy and inconstancy of natural language. Although deep learning models have illustrated remarkable execution in different NLP undertakings,their viability in tending to the Text Entailment issue is still an open question.
The examination issue of this study is to research the utilization of deep learning techniques in Text Entailment and how to capture semantic relationships between premises and hypotheses using deep learning. The following research questions will be addressed by the study:

- In Text Entailment , can deep learning architectures capture connections between premises and hypotheses in Text Entailment ?
- How does the selection of representation learning methods, including pretraining on large-scale corpora and fine-tuning on entailment-specific data, influence the implementation of deep learning models for Text Entailment ?
- What effects do different model architectures, optimization algorithms, and training techniques have on the robustness and accuracy of entailment models?
- In terms of execution and computational productivity, how do deep learning models for Text Entailment differ from conventional approaches?

This study aims to shed light on the effectiveness of deep learning methods for Text Entailment and provide understanding into the development of more precise and robust models for this crucial NLP task by addressing these research questions.

## 1.2 Research Objectives

The objective of this study is to evaluate the significance of deep learning methods for Text Entailment in capturing the logical connections between hypotheses and premises. By investigating different deep learning architectures, representation learning techniques, and training strategies, this study aspires to advance the state-of-the-art in Text Entailment and contribute to the development of more precise and robust models for this important NLP task. The research objectives of this topic are as per the following:

- To review and analyze literature on Text Entailment , deep learning, and NLP tasks.
- To investigate different deep learning architectures evaluated for Text Entailment .
- To evaluate representation learning techniques on Text Entailment performance.
- To compare the performance and computational efficiency of deep learning models outperforming traditional approaches for Text Entailment .
- To evaluate the generalisability of deep learning models by testing them on a variety of datasets and evaluating how well they perform under a variety of use cases
- To provide recommendations and guidelines for using deep learning methods to create models for Text Entailment that are more accurate and robust.

## 1.3 Significance of this work

This study investigates the use of deep learning approaches for textual entailment in Bengali Natural Language Processing, solving problems and laying the groundwork for future applications.

- **Addressing a Critical Gap**: The thesis seeks to fill a gap in Bengali NLP research by pioneering deep learning for Bengali textual entailment, contributing to the creation of Bengali NLP tools, and laying the framework for future advances.
- **Leveraging Deep Learning's Potential:** The thesis assesses deep learning performance on Bengali entailment datasets, investigates GPT-3.5 data augmentation effectiveness, and demonstrates deep learning's practicality in addressing Bengali linguistic problems.
- **Paving the Way for Real-World Applications:** This research has the potential to increase textual entailment capabilities, resulting in practical applications such as better Bengali chatbots, more accurate summarization systems, improved search engines, and instructional tools.
- **Inspiring Further Research and Development:** The thesis proposes increasing Bengali NLP databases, creating domain-specific models, including cultural and contextual data, and investigating advanced techniques such as sentiment and discourse analysis.
- **Encouraging Local Talent and Research:** The thesis, which focuses on a regional language, attempts to motivate and empower local academics and students to perform NLP research using their linguistic heritage.
- **Cross-Linguistic Implications:** The findings can help to improve linguistic diversity and inclusion in AI and NLP by guiding comparable studies in other underrepresented languages.

- **Real-World Applications:**The study's findings have practical implications for Bengali-language applications such as machine translation, sentiment analysis, chatbots, and instructional tools, hence improving technological accessibility and user experience.
- **Benchmarking and Evaluation Framework:** The article offers a standard for deep learning models' performance in Bengali textual entailment tasks, as well as a methodology for assessing and comparing their efficacy in future research.

Overall, This thesis investigates the use of deep learning for textual entailment, which advances Bengali NLP research and development, paves the way for future applications, and empowers Bengali language technology.

# Chapter 2

# Related Work

This paper works on the trueness of the news. For this, it focuses on the stance (relation between headlines and body text) detection problem with textual entailment. According to the paper, it could be an important initial move towards supporting human fact-checkers to distinguish the misleading cases.The paper uses the Fake News Challenge stage-I(FNC-1) dataset to solve the problem. This dataset contains the headline, body text and stance. The stance is classified into the terms - agree, disagree, discuss and unrelated. It uses both supervised Machine Learning and Deep Learning methods. Support Vector Machine (SVM) and Multilayer perception (MLP) are used to develop a ML-based system with different TE-based features- Overlapping Tokens, Longest Common Overlap, Modal verbs, Polarity, Numerals, Named Entities, and Cosine Similarity. Two DL-based methods - Universal Sentence Encoder (USE) and USE Incorporated with ML Features are used in this paper. Transformer-based USE is used and the DAN encoder is ignored for this work. USE produces the representation of the headlines and body which are concatenated and employed as input for the feed-forward neural network with the ReLU activation function. Four layers have been used with the softmax activation function to acquire the highest performance. A standard metric is created due to an imbalance in the dataset. The dataset comprises two stages. Among the ML approaches, MLP performs better than SVM as the problem is a multi-class problem. Although the result addressed good in this paper. Among the DL approaches, USE incorporated with ML features performs better than the others. Some hyperparameters are tuned to get a better result. Although the models have some drawbacks such as the performance is not up to the mark when the headlines and bodies are question answering type. [18] The Stanford Natural Language Inference corpus is a sizable annotated corpus of 570,000 phrase pairs that has been labelled with entailment, contradiction, or neutral. Lexicalized classifiers were able to outperform complex models because to the SNLI corpus's two times greater size than other resources, and a neural network-based model was able to compete on benchmarks for natural language inference. Several NLI research projects, including model creation, algorithm evaluation, theoretical investigation, machine translation, and question-answering, make use of the SNLI corpus.[3]

For legal reasoning, ensuring predictability and consistency in decision-making, and identifying relevant precedents, there is a great significance for determining entailment relationships between case law documents. One of the tasks in the Competition

on Legal Information Extraction and Entailment (COLIEE) is the description in this paper of a method for determining case law entailment. Using four components - calculating the resemblance between texts, applying a transformer-based method, using a threshold-based classifier, and post-processing the results, it was ranked first among the competition with an F1-score of .70. The COLIEE competition has four types of challenges: retrieval of case law, entailment of case law, retrieval of statute law, and entailment of statute law. At first, approaches zeroed in on shallow text highlights, but later, it has included the use of word embeddings, logical models, and machine learning. The task involves figuring out which paragraph(s) of a pertinent second case entails a fragment of text held inside a base case, given the base case and the subsequent case. This paper treats it as a binary classification problem. It makes a classifier which utilizes two proportions of likeness and the result of BERT for a text entailment task as elements.One is a cosine measure that evaluates multiple word tokens and another evaluates noun-phrase. The classifier figures a score consolidating these qualities and uses observationally characterized sift olds to create halfway result, which is post-handled with respect to the likelihood of the preparation dataset. BERT performs the best in terms of isolation. Although the exhibition can be improved via preparing a huge corpus of legal documents. The combination of BERT and similarity-based techniques can provides better result. There is a scope to check in the event that BERT is equipped for seeing law related information via prepared with a bigger legitimate corpus.[15]

For probabilistic reasoning, a new deep-learning approach called the neural association model (NAM) is proposed in this work. The objective of this work is to identify the association between two events by taking one event as input and computing a conditional probability of the other event. For several probabilistic reasoning tasks, including textual entailment, the two NAM structures known as deep neural networks (DNN) and relation-modulated neural nets (RMNN) are the subjects of study.After utilising popular WordNet, FreeBase, and ConceptNet datasets, DNN and RMNN outperformed other conventional networks. In case of knowledge transferring, RMNN performed better. As large amount of data mining is challenging for model training, this paper works with a collection of data of primary level.The paper begins by identifying the characteristics of events and all possible associations between them. To crack commonsense reasoning problems, modelling the event association could play a vital role. For probabilistic reasons, NAM is used as a general modeling framework. For multi-relational data, two NAM structure is used which is called RMNN. Sentences are represented by their word vectors and, the vectors aer initialized from a pre-trained word embedding model. The dimensions of word embedding models are set to 100 and, 50 for the relation code. The activation function is ReLU. Negative samples are used for the NAM's learning process. Two hidden layers is used for the final model structure. The learning rate and the dropout rate are 0.1 and 0.2 respectively. The Stochastic Gradient Descend (SGD) algorithm is used to train both DNN and RMNN. The proposed RMNN model can be promptly acclimated to one more association without relinquishing the performance in the original relations. The Winograd Schemas and other more complicated commonsense reasoning problems are common applications for the NAM model's efficiency. An explicit approach should be used to collect associate phrase pairs. Overall, it is found that NAM model solve some Winograd Schema problems successfully. But

still there is a lot of scope to work with. [6]

Sentence Connectivity refers to a textual characteristic that defines if sentences are logically coherent or not. This paper works with text summarization based on sentence connectivity. A graph-based algorithm, Weighted Minimum Vertex Cover (WMVC) is used to solve this problem. Textual Entailment is used to define sentence connectivity. This method performed better than other traditional methods. As the quantity of information in the digital world is mounting, users face difficulty to obtain it. Text summarization is the best possible solution. The primary goal of this paper is to summarize text using extraction. As wMVC is a graph-based algorithm, vertices represent the sentence and edges represent the relation between the sentence. And the degree of relation is defined via textual entailment. The associated vertices would resemble a document's main content's cover. Text entailment is an lopsided connection between two text fragments indicating whether one part can be induced from the other. Textual Entailment is used to determine the connection between two nodes of the graph and wMVC is used to determine the summary of sentences. The proposed methodology is divided into four segments - compute textual entailment scores, connectivity scores based on entailments, entailment connectivity graph construction and finally applying wMVC algorithm. To adjust the scores to the WMVC algorithm, that looks for a negligible arrangement, the scores are converted iinto positive weights in rearranged request.. Finally, the wMVC is applied to define the minimum vertex cover. Integer Linear Program(ILP) is used to find minimum cover. DUC 2002 dataset is used in this work consists of 60 news article. A transformation-based TE system called BIUTEE (Stern and Da- gan, 2012) is used to calculate the textual entailment score between pairs of sentences. It is trained with 600 text-hypnothesis pairs of the RTE dataset. This method's performance is compared with three re-implemented methods- Sentence selection with tf-idf, LTT, and ASTEC. This method performed better than the previous ones, which used a frequency-based baseline and TE for summarization. There is a scope for the future to apply this method to smaller segments of the sentences. [2]

The study discussed in the paper tackles the issue of textual entailment, which entails figuring out if one text (the hypothesis) may be inferred or implied from another text (the premise). The authors hope to improve the accuracy of textual entailment by incorporating knowledge-awareness into the work. KCITEN is a Knowledge-Context Interactive Textual Entailment Network (KCITEN) that uses external knowledge graphs and graph attention networks to learn graph level phrase representations. Research on the SciTail dataset demonstrates that KCI-TEN outperforms cutting-edge techniques. The authors provide a cutting-edge method for capturing associations between words and entities in text that makes use of graph attention networks. The suggested approach successfully incorporates outside knowledge sources to improve text comprehension by modeling text information as a graph and utilizing attention mechanisms. The authors devise a graph attention network architecture to address the issue, which can learn the significance of various textual components and incorporate outside knowledge. The graph nodes in the model are given weights based on how relevant they are to the textual entailment task using attention processes. The model builds a thorough representation of the text by merging data from the graph nodes, which is subsequently applied to the prediction

of text entailment. The examinations did by the creators show that the recommended strategy works. Precision, recall, F1-score, and accuracy are some of the evaluation criteria employed. The knowledge-aware graph attention network beats cutting-edge models in textual entailment tests, demonstrating its potential for use in information retrieval and natural language processing. This research uses graph attention networks to provide a knowledge-aware method for text entailment. This emphasizes the significance of knowledge awareness and demonstrates the capability of graph attention networks to enhance textual entailment task accuracy. These results add to the field of natural language processing and provide new research and development opportunities. [13]

The study discussed in this paper focuses on employing deep learning approaches for entity matching, which is the process of locating and connecting records from various data sources that refer to the same real-world entities. The study's objective is to investigate several deep learning model design options for entity matching and assess how well they perform. Researchers have presented a design space exploration approach that methodically examines various deep learning model parts and configurations in order to achieve this goal. The framework includes a number of crucial components, such as the model architecture, feature representation, similarity metric, and training methods that are selected. The performance of various deep learning models in comparison to conventional entity matching methods was compared in-depth tests on numerous real-world datasets in order to assess the proposed framework. Precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) were some of the evaluation metrics employed. Experiment results demonstrate that deep learning models can perform as well as or better than more conventional entity matching techniques. The researchers discovered that several deep learning architectures, such as recurrent neural networks and Siamese neural networks, have promising results in accurately matching items across various datasets. Overall, this study demonstrates how deep learning techniques have the potential to improve the precision and effectiveness of entity matching tasks. The design space exploration approach helps academics and practitioners make well-informed decisions when implementing deep learning in this field by offering important insight into numerous design decisions and considerations that can impact the performance of deep learning models for entity matching.[11]

The task of text recognition, which entails deciding whether a specific text (or hypothesis) can be logically derived from another text (or hypothesis), is the main emphasis of this research study. The purpose of the study is to determine whether integrating machine learning and dependency analysis techniques for this assignment is successful. Researchers offer a novel method that uses dependency analysis to glean syntactic connections between words in base and inference texts in order to address the goal. In a machine learning context, these dependence relationships are employed as features to train a classifier that can identify textual items.Researchers test their suggested approach against current state-of-the-art methods using benchmark datasets and compare the results. Common assessment metrics are used to evaluate the model's performance like accuracy, precision, recall, and F1-score. According to the experiment's findings, combining dependency analysis and machine learning techniques greatly enhances text recognition performance over conventional approaches. The suggested approach successfully identifies logical inferences in texts with high accuracy and an F1-score. The model obtains a deeper grasp of text struc-

ture by successfully capturing syntactic relationships through dependency analysis, enabling more precise predictions of textual entailment. By demonstrating the viability of fusing linguistic analysis with machine learning techniques, this research advances natural language understanding tasks. In conclusion, this study offers a successful strategy for identifying textual entailment by fusing machine learning with dependency analysis. The results show that this strategy works better than other ways and offers insightful information about how to leverage linguistic elements to improve text-related task performance.[5]

The challenging process of text entailment is the focus of the research presented in this paper, which involves figuring out if one text (referred to as an inference) may be logically inferred from another text (referred to as a premise). The authors seek to considerably increase text accuracy and dependability by introducing the idea of knowledge-awareness into their methodology. Researchers suggest a probabilistic model that includes numerous language elements to handle the task. The model calculates the probability of entry between the base and the hypothesis text using a probabilistic estimation approach. The suggested method exhibits competitive performance in textual entailment recognition through trials on common benchmark datasets. The model is able to capture semantic links and produce precise predictions by using probabilistic inference. The model develops a greater grasp of the relationship between the text of the premise and the inference by utilizing linguistic aspects and probabilistic inference. By proving the efficiency of probabilistic inference in identifying text entailment, this study advances the field of natural language understanding. The work successfully demonstrates a way for identifying textual entailment using probabilistic inference methods. The results demonstrate the competitive performance of the suggested approach and offer insightful information about the usefulness of potential models for enhancing accuracy on text-related tasks.[4]

This paper focuses on the task of extractive multi-document summarizing, which entails identifying significant phrases from many documents and fusing them to create succinct summaries. In order to handle this job, the study suggests a method that combines text entailment, phrase compression, and the knapsack problem. The researchers created a framework that uses text joining techniques to assess the underlying relationships between a collection of candidate phrases and summaries in order to accomplish this goal. Additionally, repetition is minimized and the effectiveness of a few well-chosen sentences is increased by using language reduction. To select the most informative sentence while adhering to the length restriction, the knapsack problem, an integrated optimization method, is still used. A benchmark dataset is used to compare the proposed method to well-known extractive summarization techniques. The ROUGE score, which is one of the evaluation criteria used, measures how closely the generated summary resembles the reference summary. According to experimental findings, the suggested strategy performs better in terms of ROUGE score than the standard procedures. Thanks to a combination of text entailment, sentence compression, and knapsack problems, the system efficiently extracts important information from input sources to provide brief but useful summaries. This paper overview of the technique for extracting multi-document summaries described in this thesis paper, including text entailment, phrase compression, and the knapsack problem. The results show how effective this technique is in generating high-quality summaries. This work expands on extractive summarization techniques

and provides information on the benefits of using a range of summaries.[12]
The paper provides an extensive overview of Recognizing Textual Entailment (RTE) as a crucial tool for evaluating Natural Language Processing (NLP) systems. It emphasizes how RTE is instrumental in contrasting the semantic comprehension abilities of various NLP systems. Over the past 30 years, this paper explores different approaches employed to evaluate and compare NLP systems, recognizing the necessity of evaluating NLP systems to enhance language understanding. The authors conduct an in-depth analysis of the work undertaken during the last three decades in the development of RTE datasets and their utilization in evaluating NLP models. A significant distinction is made between Natural Language Inference (NLI) and Recognizing Textual Entailment (RTE), shedding light on their often interchangeable usage. While NLI is employed for tasks requiring inferences from natural language, RTE is specifically designed to classify or predict whether one sentence logically follows another. The term "recognizing" in RTE aptly encapsulates its role in classifying sentence truth value. The paper rationalizes the preference for RTE over NLI as a more accurate representation of current NLP work. The authors also delve into the various aspects involved in evaluating NLP systems, including the differentiation between General Purpose and Task-Specific Evaluations, as well as Intrinsic and Extrinsic Evaluations. They conduct research on Probing Deep Learning NLP Models, exploring how auxiliary or diagnostic classifiers can be effectively utilized to interpret and analyze the inner workings of NLP models. In summary, Recognizing Textual Entailment serves as a robust method for assessing the inference capabilities of NLP models. This survey highlights recent advancements in RTE datasets, particularly those focusing on specific linguistic phenomena. It also revisits past methodologies for evaluating NLP systems, emphasizing the dichotomy between intrinsic and extrinsic evaluations and general-purpose versus task-specific evaluations. Lastly, the paper underscores the ongoing efforts to create RTE datasets tailored to specific linguistic phenomena, reinforcing RTE's pivotal role as an evaluation framework in the realm of NLP. [17]
The creation of the SCITAIL dataset aims to elevate the accuracy of automated science question answering systems and foster advancements in natural language processing models. SCITAIL comprises pairs of scientific sentences, encompassing a hypothesis that can either entail, contradict, or remain neutral to the first sentence. The dataset represents a natural entailment dataset, originating from multiple-choice question answering, and introduces a novel model leveraging linguistic structure within hypotheses to surpass existing techniques.
Recognizing textual entailment (RTE) stands as a formidable challenge in the domain of natural language understanding. While large datasets like RTE-n, SICK, and SNLI have emerged to facilitate the development of robust RTE systems, they do not always encompass the breadth of entailment queries encountered in real-world tasks. SCITAIL, presented as the largest entailment dataset, is directly derived from an end task, featuring naturally occurring text serving as both a premise and a hypothesis. SCITAIL aims to capture the intricate reasoning essential for textual question answering. Although current RTE systems, including neural entailment models, exhibit moderate performance on this dataset, the introduction of an asymmetric Decomposed Graph Entailment Model (DGEM) elevates accuracy to 77.3%. SCITAIL's creation involves an annotation scheme designed to construct an entailment dataset specific to the science question answering task, drawing from multiple-

choice questions within the SciQ dataset. Notably, SCITAIL shares statistical similarities with existing datasets, eliminating any straightforward advantages based on overlap proportions or token lengths. The "entails" class exhibits higher word overlap, while SCITAIL's proportions align more closely with SNLI. On average, entailment examples tend to feature longer premises. Comparative evaluations involve two state-of-the-art neural entailment systems and a straightforward overlap-based model trained on the SCITAIL dataset. In summary, the authors introduce SCITAIL as a novel natural dataset for textual entailment, posing a significant challenge for current state-of-the-art models. They propose an innovative neural entailment architecture capable of utilizing graph-based syntactic/semantic structures from hypotheses, resulting in a notable 5% improvement on the dataset. The authors anticipate that this model establishes a robust baseline for future advancements on SCITAIL, facilitating progress in reasoning with complex, natural language in the field of automated science question answering. The creation of the SCITAIL dataset aims to elevate the accuracy of automated science question answering systems and foster advancements in natural language processing models. SCITAIL comprises pairs of scientific sentences, encompassing a hypothesis that can either entail, contradict, or remain neutral to the first sentence. The dataset represents a natural entailment dataset, originating from multiple-choice question answering, and introduces a novel model leveraging linguistic structure within hypotheses to surpass existing techniques.

Recognizing textual entailment (RTE) stands as a formidable challenge in the domain of natural language understanding. While large datasets like RTE-n, SICK, and SNLI have emerged to facilitate the development of robust RTE systems, they do not always encompass the breadth of entailment queries encountered in real-world tasks. SCITAIL, presented as the largest entailment dataset, is directly derived from an end task, featuring naturally occurring text serving as both a premise and a hypothesis. SCITAIL aims to capture the intricate reasoning essential for textual question answering. Although current RTE systems, including neural entailment models, exhibit moderate performance on this dataset, the introduction of an asymmetric Decomposed Graph Entailment Model (DGEM) elevates accuracy to 77.3%. SCITAIL's creation involves an annotation scheme designed to construct an entailment dataset specific to the science question answering task, drawing from multiple-choice questions within the SciQ dataset. Notably, SCITAIL shares statistical similarities with existing datasets, eliminating any straightforward advantages based on overlap proportions or token lengths. The "entails" class exhibits higher word overlap, while SCITAIL's proportions align more closely with SNLI. On average, entailment examples tend to feature longer premises. Comparative evaluations involve two state-of-the-art neural entailment systems and a straightforward overlap-based model trained on the SCITAIL dataset. In summary, the authors introduce SCITAIL as a novel natural dataset for textual entailment, posing a significant challenge for current state-of-the-art models. They propose an innovative neural entailment architecture capable of utilizing graph-based syntactic/semantic structures from hypotheses, resulting in a notable 5% improvement on the dataset. The authors anticipate that this model establishes a robust baseline for future advancements on SCITAIL, facilitating progress in reasoning with complex, natural language in the field of automated science question answering.[10]

This paper delves into recent developments in constructing neural network-based

entailment systems employing Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and attention mechanisms. Notably, it highlights the introduction of high-quality datasets like SNLI and MNLI, which serve as crucial resources for training deep learning models.

Textual entailment, a concept examining the relationship between two texts to determine if one implies the meaning of the other, holds significant importance in various Natural Language Processing (NLP) applications. These applications span Question Answering, Text Summarization, and Machine Translation evaluation. Deep learning techniques have proven highly effective in multiple NLP domains, including machine translation, sentiment analysis, and textual entailment, which forms the focal point of this paper.

The paper provides an insightful exploration of the deep learning models developed for building entailment systems, emphasizing their role in comprehending sentence and text fragment meanings—a core aspect of Natural Language Inference (NLI) research in NLP. The broader applications of language inference encompass Machine Translation Evaluation, Information Retrieval, Question Answering, Information Extraction, and Text Summarization, where it aids in deciphering the essence of given textual content.

The results obtained from neural network models designed for textual entailment recognition exhibit great promise. Current deep learning methodologies, including Word Embeddings, Recurrent Neural Networks (RNNs), Attention mechanisms, and Memory Nets, are seamlessly integrated into this field. These improved approaches have led to enhanced accuracy rates for NLI datasets like SNLI and MNLI. The utilization of superior datasets, such as MNLI, has further propelled the development of superior neural network entailment models.

In conclusion, this paper underscores the significant strides made in textual entailment through the adoption of deep learning techniques. The incorporation of advanced methodologies and high-quality datasets has not only elevated the accuracy of entailment recognition but also expanded its applicability across diverse NLP domains.[20]

This paper introduces a novel neural model designed to assess entailment by analyzing two sentences using Long Short-Term Memory (LSTM) units. It goes a step further by incorporating a word-by-word neural attention mechanism, which encourages nuanced reasoning over word and phrase entailments. The results demonstrate its superiority over prior neural models and classifiers when applied to a textual entailment dataset.

Recognizing textual entailment (RTE) involves the task of determining the relationship between two natural language sentences—whether they contradict each other, bear no relation, or if the first sentence (referred to as the premise) implies the second sentence (referred to as the hypothesis). Traditional systems for RTE have heavily relied on engineered Natural Language Processing (NLP) pipelines, manual feature creation, and external resources. Efforts to develop end-to-end differentiable neural architectures for RTE have fallen short, primarily due to the scarcity of large, high-quality datasets. Achieving an end-to-end differentiable solution is desirable, as it avoids specific assumptions about the underlying language.

This paper builds upon the foundation laid by Bowman et al. (2015), who introduced the Stanford Natural Language Inference (SNLI) corpus along with a neural

network incorporating Long Short-Term Memory units (LSTM), achieving an RTE accuracy of 77.6% on this dataset. The authors propose an attentive neural network that can reason over entailments of word and phrase pairs by processing the hypothesis within the context of the premise. The baseline LSTM model showcased an accuracy of 80.9%, surpassing a straightforward lexicalized classifier customized for RTE by 2.7 percentage points. Furthermore, an extension featuring word-by-word neural attention outperformed the robust LSTM benchmark by an additional 2.6 percentage points.

The versatility of LSTMs in the context of RTE is showcased through neural attention and word-by-word attention mechanisms. These mechanisms enable attention both over the premise conditioned on the hypothesis and vice versa. Notably, the SNLI corpus, being two orders of magnitude larger than existing RTE datasets like SICK, features sentence pairs curated by human annotators, rendering it a valuable resource for training neural architectures.

This research underscores the potential of end-to-end differentiable models to advance the state-of-the-art in textual entailment recognition, particularly on a large, meticulously curated, and annotated corpus like SNLI. LSTM recurrent neural networks outperform conventional approaches such as classifiers with manually created features and neural baselines. The incorporation of attention mechanisms elevates the predictive capabilities of these models, resulting in a new state-of-the-art accuracy for entailment recognition on the Stanford Natural Language Inference corpus. Future research avenues include exploring transfer learning tasks, extending attention strategies to larger text units through hierarchical attention processes, and investigating alternative differentiable memory strategies to further enhance RTE performance.[7]

This study employs a deep learning (DL) architecture and natural language inference (NLI) to find contradictions and inconsistencies in the medical literature. The authors investigate well-known NLI models and phrase embeddings, improving these models' capacity to recognize inferences by fusing Deep learning architecture combined with typical machine learning (ML) traits. The suggested approach seamlessly combines lexical, contextual, and compositional semantics to faithfully reflect biomedical language. Over the past ten years, there has been a considerable advancement in clinical and medical research, which has increased the number of published research dramatically. In 1975, there were 10 daily publications; by 1995, there were 55; and by 2015, there were 95. As of 2017, the well-known database PubMed had a vast collection of 27 million articles, 2 million medical reviews, 500,000 clinical trials, and 70,000 systematic reviews. Our work suggests a fresh strategy in response to this enormous volume of data. Using cutting-edge phrase encoders, carefully constructed language cues, domain-specific characteristics, and a model, this technique categorizes sentence pairs as being either entailed, contradicting, or neutral. This method stands out because it combines deep neural networks—including encoders—with traditional machine learning characteristics. The 20 human-engineered features, which make up typical NLP characteristics, are made to capture the subtleties of the text's context, lexicon, and semantics. The effectiveness of a number of Scikit-learn classification techniques, comprising Naive Bayes, Gradient Boost, Random Tree, Support Vector Machine, and Linear Regression Model, is assessed throughout the experimental period. The deep learning (DL) model has a siamese-like architecture and includes parallel duplicates, several intermediate dense layers,

rectified linear activation (ReLU) functions, a dropout layer with a 0.3 dropout rate, and a prediction layer with three nodes and a softmax activation function. Additionally, a l2 regularization weight of 0.01 and an exponentially decreasing learning rate are applied. The developed hybrid design exhibits promising results in terms of both size and secret layers. But because the model's influence must be properly generalized, additional study of a larger corpus is necessary. Additionally, it could be feasible to enhance encoder performance by retraining the sentence encoders using domain-specific sources like research publications and clinical notes. A more effective boosting vector to assist the neural network may be discovered by taking a larger variety of features into consideration and doing feature ablation analysis. The results of this work pave the way for a stronger understanding of textual entailment in the biological field. This development provides a potent tool for navigating the ever growing body of medical literature.[16]

The authors of the study "Identifying Attack and Support Argumentative Relations Using Deep Learning" (Cocarascu  Toni, 2017) look into how deep learning techniques can be used to find attack and support in argumentative Relations. The main goal of this research is to create an automated system that can recognize and understand the intricate structure of arguments.

The authors use Long Short Term Memory based neural networks (LSTMs), a type of deep learning technique, to analyze argumentative texts and capture the inherent patterns and features indicative of attack and support relationships. They took 2 pieces of text (up to 50 words) and feed it to two different LSTMs and produced a 100-dimensional embedding. Then merged both of those embeddings and gave the combined embedding to a softmax layer. Softmax finally outputs if one part of the text is supportive, attacking or neutral to the other part.

The researchers run trials with a dataset made up of argumentative writings to evaluate how well their approach performs. The outcomes show that their deep learning model performs better than conventional approaches, displaying a high level of accuracy in recognizing the complex relationships between arguments. The development of argument mining and natural language understanding is aided by these findings. They achieved 89.53% accuracy in the test after training.

To conclude, the research introduces a novel deep learning-based approach for identifying attack and support argumentative relations. The study highlights how deep learning techniques can be used to analyze and comprehend arguments, which has implications for many different applications, including automated debate systems, sentiment analysis, and opinion mining.[8]

The paper, titled "Recognizing Partial Textual Entailment," addresses the task of identifying partial textual entailment, which involves determining what can be inferred from one text to another. The authors present a novel approach that focuses on capturing subtle and nuanced relationships between sentences. In order to properly recognize partial text subjects in different NLP applications, including querying, summarization, and information retrieval, this study begins by stressing its significance. The difficulties presented by the partial nature—where only a portion of the knowledge in one text may be deduced from another—are highlighted by the writers. The authors provide a framework that integrates lexical, syntactic, and semantic elements to capture links between phrases in order to overcome these difficulties. To extract pertinent characteristics, they employ syntactic parsing, semantic role labeling, and distributional similarity metrics. The study also presents a brand-new

dataset created especially for measuring partial text entailment. The authors' experimental findings demonstrate the effectiveness of the strategy they suggest. A considerable performance improvement is seen when compared to the previous approach. The study also examines the results of various feature combinations and outlines the advantages and disadvantages of the suggested framework. The authors also go through possible applications and implications of their work, emphasizing how partial text entailment recognition might help with various NLP tasks. They also provide recommendations for future lines of inquiry, such as looking at more language sources and applying deep learning methods to boost performance even further. This research concludes by presenting a unique framework for identifying incomplete text content. Experimental findings show the usefulness of the technique, showing its potential for enhancing diverse natural language processing applications. The suggested method leverages lexical, syntactic, and semantic aspects to capture complex interactions between sentences. This study makes a contribution to the field of natural language comprehension as a whole and paves the way for future developments in the understanding and modeling of partial textual entailment.[1]
This paper works with a Question Answering system based on legal information. The system is evaluated by the dataset of (COLIEE)-2017. COLIEE-2017 has two phases - legal information retrieval and textual entailment. For phase-1, this paper works with the TF-IDF approach and for phase-2, it compares the approximate meanings of queries. It builds a logic-based representation as a semantic analysis result and classifies the questions into two categories - easy and difficult. The answer of the easy category question is obtained from the entailment answer and for the others, it uses unsupervised learning methods. After evaluation, it is ranked highest in the phase-2 of this competition. The answer to a question is typically determined by estimating semantic comparability between question and answer. At first, this work extracts features from logical representation and then determines the semantic match between the question and corresponding articles. After negation analysis, an unsupervised model is constructed to get the yes/no answer. The COLIEE legal IR task has several sets of questions with the Japan civil regulation articles as records (1044 articles in total). For a dry run, it uses 10 sets of data that include 581 queries and for testing, it uses 78 queries. The data is translated from Japanese to English language.TF-IDF outperforms language model-based IR models in this phase. The yes/no question answering is divided into four segments - condition, conclusion and exceptional case detection, detect negation, semantic representation and unsupervised machine learning. After this segmentation, the last segment is section is viewed as an conclusion, and the remainder of the sentence is considered as a condition. Negation detection is done by considering affix, words and concepts. In semantic representation phase, 47.85% of the data were assigned as easy questions, and 52.15% were assigned as non-easy questions. The unsupervised machine learning algorithm considers the features - negation, semantic representation and lexical semantic. For unsupervised learning, it makes use of the K-means clustering algorithm. This system scored highest in COLIEE-2017 competition. Semantic similarity error is part where it can be work in future.[9]
The paper titled "UKP-Athene: An Introduction" provides an overview of the UKP-Athene system, which addresses the problem of verifying claims using textual entailments that span multiple sentences. Textual entailments for claim verification that span multiple sentences. Verification of statements establishes their veracity.

The authors present a novel method that assesses assertions for coherence and consistency with the evidence offered in a given text using textual entailments. The architecture and parts of the UKP-Athene system are described in general in the article. It has modules for textual entailment, claim extraction, and evidence retrieval. The evidence retrieval module retrieves pertinent sentences as evidence while the claim extraction module extracts claims from input texts. The logical connection between claim and evidence sentences is then thoroughly examined by the textual entailment module. The authors ran tests to gauge UKP-Athene's efficacy using the FEVER dataset, a benchmark for claims validation. The findings demonstrate that, when compared with cutting-edge models, the suggested technique achieves equivalent accuracy in claim verifications. The report also analyzes the shortcomings of the UKP-Athene system, including its dependence on categorical evidence and possible lexical overlap issues. Future study directions are suggested by the authors, including the use of outside information sources and the investigation of cutting-edge methods to improvement system performance. The UKP-Athene system is a multi-sentence textual entailment technique for claim verification that is introduced in the paper's conclusion. By examining the coherence and consistency of claims and evidence, the system achieves encouraging results in assessing the truthfulness of assertions. The findings expand the field of natural language processing and offer guidance for future improvements to claim verification systems.[14]

The research paper "BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla" by Bhattacharjee et al., published in 2022, marks a significant advancement in Natural Language Processing (NLP) for the Bangla language. The paper details the development of a big pretraining dataset, 'Bangla2B+', from multiple sources in Bangla and the pretraining process of BanglaBERT using the ELECTRA framework. The Bangla Language Understanding Benchmark (BLUB) comprises of many NLU tasks, such as named entity identification, sentiment classification, natural language inference, and question answering, on which this model is built.

There are limitations or biases in the way data are collected and processed, which can affect model performance or generalizability. I also identified a flaw in their approach, that they used an NLI dataset that does not retain the authentic structure of Bengali sentences. Instead, they relied on a translation library to convert the content, which may not accurately reflect the natural syntax and semantics of the Bengali language. This can potentially lead to errors in the dataset, affecting the model's ability to learn and understand the natural structure of Bengali sentences. BanglaBERT is the state-of-the-art model that performs better than multilingual and monolingual models. The authors have made the model, datasets, and a leaderboard available to the public. This research makes a major strand towards NLP abilities for Bangla, offering resources and points of reference for future efforts in the field. [19]

# Chapter 3

# Challenges of Textual Entailment in BRACU_NLI dataset

The Bengali language presents unique challenges in textual entailment, such as ambiguity, paradox, and inference, which might impede proper detection due to its specific characteristics. Some of the key challenges are :

- **Morphological Complexity:** Bengali has a strong agglutinative morphology, with morphemes (meaningful units) combining to produce words. This adaptability results in innumerable inflections and derivations, often with subtle semantic nuances. Traditional entailment models may struggle to capture these nuances and draw reliable conclusions based on morphological differences.
- **Lack of Large-Scale Annotated Datasets:** Bengali is experiencing a shortage of high-quality entailment datasets, which are critical for training and assessing deep learning models, potentially contributing to model bias and poor generalisation.
- **Cultural and Contextual Dependencies:** Bengali entailment is dependent on cultural context, shared knowledge, social conventions, literary references, and historical events, rendering existing models insufficient for appropriately judging Bengali literature.
- **Pronoun Ambiguity and Ellipsis:** Bengali's use of pronouns and ellipsis for brevity can confuse subject-verb agreement and coreferential noun identification, perhaps leading to model misinterpretation or inference issues.
- **Syntactic Variations and Word Order:** Bengali word order is more flexible than English, allowing for different sentence patterns and rhetorical goals; yet, inflexible syntactic models may misread non-standard word order meanings.
- **Code-switching and Loanwords:** Bengali text regularly contains words from English and Sanskrit, complicating models' understanding of the semantic meanings of loanwords inside Bengali language.
- **Dialectal Variations:** Textual entailment is complicated by dialectal variances in Bengali across regions such as Bangladesh and Indian states, which include differences in lexicon, syntax, and pronunciation.

- **Technical Constraints:** Processing Bengali text can be difficult due to low computing resources, especially in locations where Bengali is widely spoken.
- **Semantic Ambiguities**: Bengali, like many languages, has numerous meanings depending on the situation, resulting in semantic ambiguity. Resolving these uncertainties is critical for understanding entailment.
- **Lack of Standardization:** Due to varying educational backgrounds, digital Bengali texts sometimes lack spelling and grammar standardisation, making natural language processing jobs more difficult.

To address these issues, Bengali entailment datasets should be enlarged, morphology-aware models should be merged, cultural and contextual information should be included, pronoun resolution and ellipsis handling approaches should be created, and models should be adaptive.

# Chapter 4

# Workflow

The work process for this work can be divided into a few key stages:

1. Collect datasets with diverse linguistic characteristics for Text Entailment and preprocess them to ensure data quality and consistency.
2. Design and implement deep learning architectures for Text Entailment such as RNN, CNN, BERT etc.
3. Investigate the effect of pre-preparing and fine-tuning tweaking on deep learning execution.
4. Implement optimization algorithms and training strategies to train deep learning models. Train models using preprocessed data and evaluate performance metrics such as accuracy, precision, F1-score etc.
5. Evaluate generalization and transferability of deep learning models.
6. Generate synthetic datasets using large language model and evaluate this dataset with transformer based models.
7. Summarize the most important findings of the study, such as insights into the effectiveness of deep learning for text-based comprehension and potential areas for further investigation.

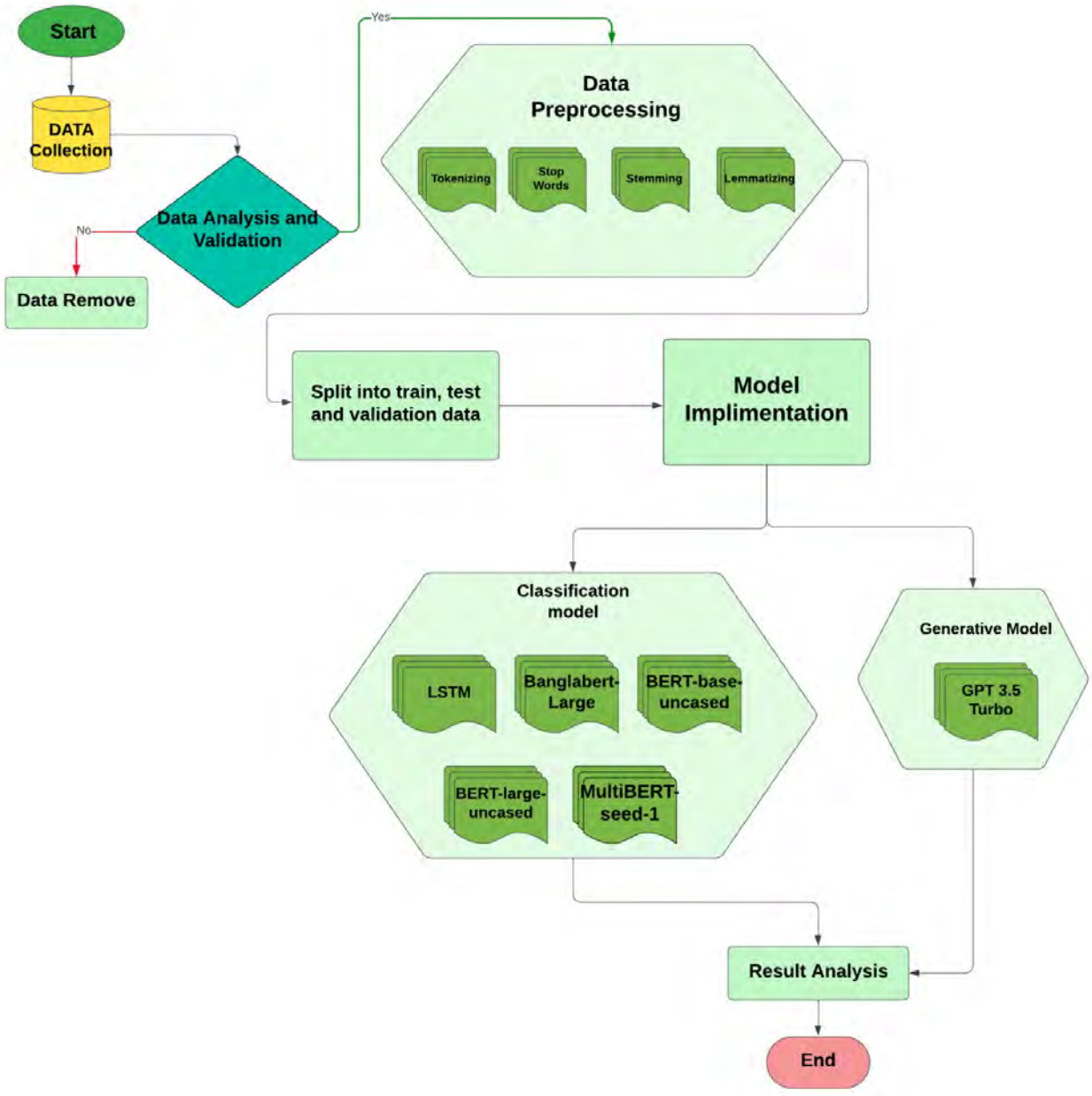



Figure 4.1: The flow chart of our work

# Chapter 5

# Data

## 5.1 Data Collection

We collected Data in two ways. Firstly, we picked 3292 sentences as premise from the Stanford Natural Language Inference (SNLI) Corpus and translated it to Bengali[3], maintaining proper Bengali sentence structure. Then 60 students of *Natural language processing II* course from BRAC University CSE contributed this dataset. Each individual task required the worker to provide hypothesis for each of our three labels — entailment, neutral, and contradiction—in order to force the data to be evenly distributed among these classes. The worker was given premise scene descriptions from a pre-existing corpus.Then, they were given google forms with proper instructions to write the hypothesis.They submitted the hypothesis by filling up the google forms. Finally, total 9876 sentence pairs generated by them in this dataset.

Secondly, To make the dataset bit larger, we picked 16779 sentence pairs randomly from the Stanford Natural Language Inference (SNLI) Corpus and then translated it to Bengali and corrected the sentence sequence as Bangla grammatical rules.This time we picked both the premise, hypothesis and containing label.Then we merged both and make a dataset containing 23067 data in total. Of these, 7689 are contradictions, 7654 are neutrals, and 7673 are entailments.

There were no existing Bengali dataset of entailment till now except one.But that dataset was not following Bengali sentence structure and sequence. This is the first time Bengali entailment dataset was made with the help of some NLP course's students who ensured that the sentence sequence and Bengali sentence structure is maintained properly. We randomly selected 16779 sentence pairs from the Stanford Natural Language Inference Corpus, translated them to Bengali, corrected them using Bangla grammatical rules, and merged them to create a total dataset of 23067 data. We assigned it the name BRACU_NLI dataset. Moreover, the students who contributed are Bengali people and speak in Bangla as their mother tongue. So, The main significance of this dataset is, following proper Bangla grammatical rules and structure.

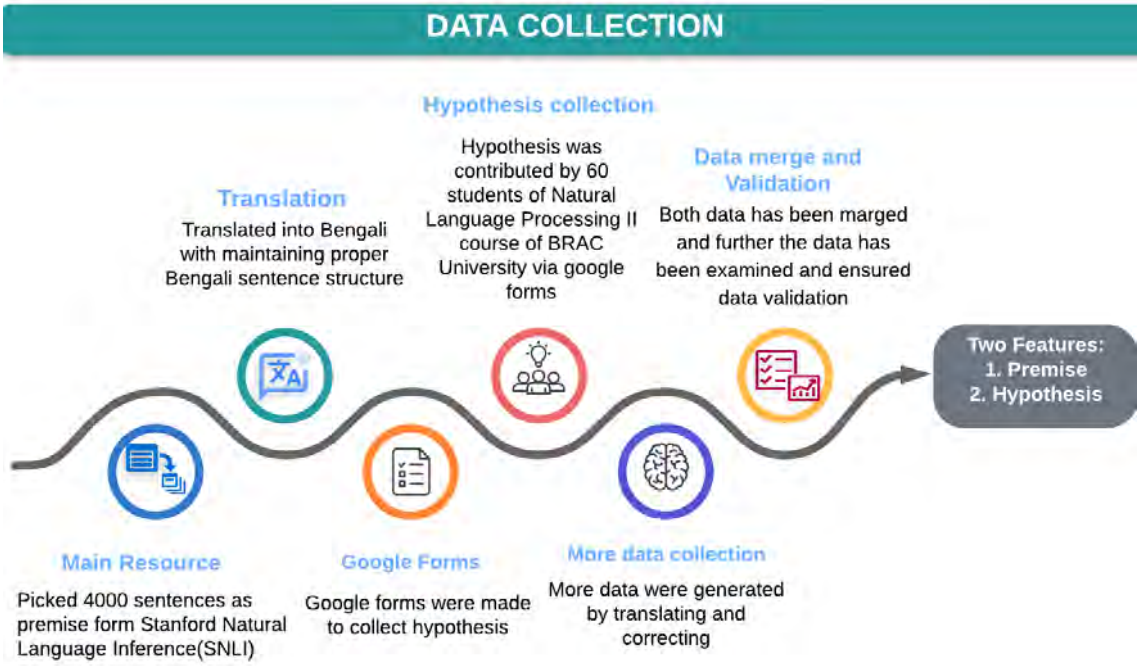



Figure 5.1: Dataset

| Premise | Hypothesis | label | Similarity |
|---|---|---|---|
| শিশুরা হাসতে হাসতে ক্যামেরার দিকে হাত নাড়াচ্ছিল | শিশুরা হাসিমুখে ক্যামেরার দিকে তাকিয়ে হাত নাড়া... | entailment | 0.923020 |
| শিশুরা হাসতে হাসতে ক্যামেরার দিকে হাত নাড়াচ্ছিল | মেয়েরা গম্ভীর মুখে তার কথা ভিডিও করছিল। | contradiction | 0.110095 |
| শিশুরা হাসতে হাসতে ক্যামেরার দিকে হাত নাড়াচ্ছিল | শিশুরা হাসতে হাসতে খেলা করছিল। | neutral | 0.622482 |
| একজন বৃদ্ধ লোক একটি রেস্তরায় বসে কমলার জুস পান... | একজন লোক জুস খাচ্ছে। | entailment | 0.575961 |
| একজন বৃদ্ধ লোক একটি রেস্তরায় বসে কমলার জুস পান... | দুই মহিলা একটি মাঠে মদ খাচ্ছেন। | contradiction | 0.182121 |
| একজন বৃদ্ধ লোক একটি রেস্তরায় বসে কমলার জুস পা... | একজন বয়স্ক লোক একটি দোকানে জুস খাচ্ছেন। | neutral | 0.709780 |

Figure 5.2: Dataset

## 5.2 Data Preprocessing

For pre-processing, we translate the premise and hypothesis sentences into English to determine the Semantic Textual Similarity (STS) value. We used Google Translator package for Python.
**STS**: Semantic Textual Similarity (STS) is a powerful method that helps computers understand complex language, compare texts effectively, and create creative content like paraphrases or summaries.

$$\text{LinSim}(u, v) = \frac{2 \cdot \log P(\text{LCS}(u, v))}{\log P(u) + \log P(v)}$$

- The LCS represents the longest sequence of elements appearing in the same order in both sequences.
- The probability of the LCS occurring is represented by the term "P(LCS)".
- The probabilities of sequences u and v occurring are represented by the terms "P(u)" and "P(v)".
- The natural logarithm is used to represent the natural logarithm.
- The numerator of the formula emphasizes the contribution of the LCS to the similarity measure, scaled by a factor of 2.
- The denominator represents the sum of the logarithms of the probabilities of individual sequences.
- The formula is designed to work with probabilistic models or statistical information about the sequences.

We used *en_stsb_bert_large* model of Spacy library to determine STS value. The value was between 0 and 1. We produced graphs for entailment, contradiction, and neutrality by plotting these values.

By analyzing these graphs and other factors, we developed two logic. We dropped the column based on these logics:

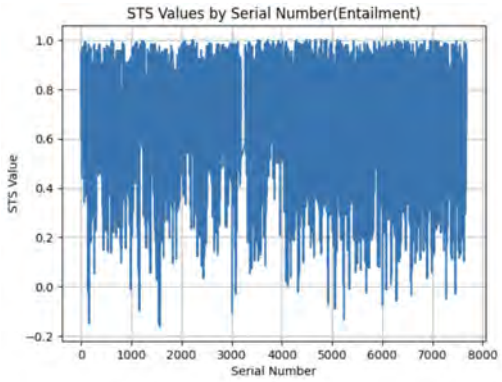
**First Thresholding**

```
where labels == 'entailment' and sts_value > .60
where labels == 'neutral' and sts_value > .40
where labels == 'contradiction' and sts_value < .30
```

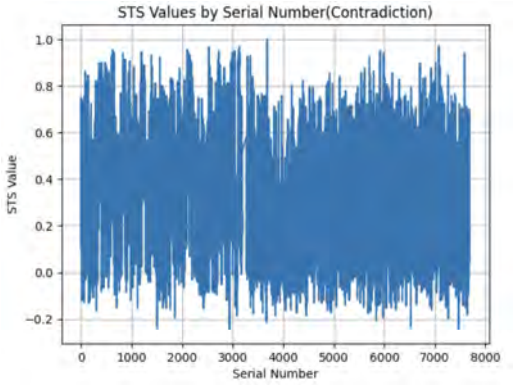
Finally, we got 15792 sentence pairs. Among them we took 12144 data for training ,1729 data for validation and 1919 data for test for the transformer based models.

We have conducted additional thresholding tests under differnet conditions of STS value. Subsequently, we compared these two thresholding and demonstrated which model operates under which circumstances.

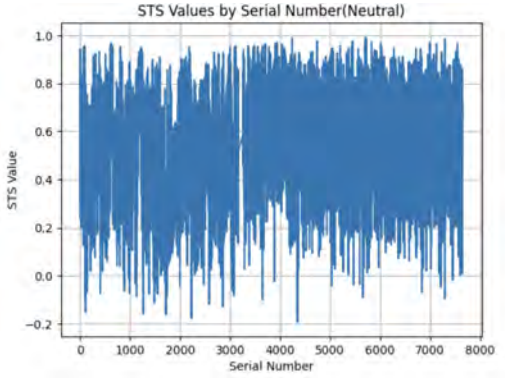
**Second Thresholding**




(a) STS value of Entailment




(b) STS value of Contradiction




(c) STS value of Neutral

```
where labels == 'entailment' and sts > .70
where labels == 'neutral' and (sts > .30 and sts < .70)
where labels == 'contradiction' and sts < .30
```

We got 13321 sentence pairs. Among them we took 11000 data for training ,1172 data for validation and 1149 data for test for the transformer based models.

We did tokenizing, removing stopwords, stemming and lemmatizing using NLTK and *Bengali Natural Language Processing(BNLP)* package. We used Label Encoder in label column to convert into numerical value and concatenated *premise* and *hypothesis* value.

### 5.2.1 Tokenizing

In order to give NLP algorithms a consistent representation of text, a tokenizer in NLP separates text into tokens like words or letters. We used word tokenizer of *BNLP* package.
**Word Tokenizer** : Tokenizers are frequently used to split down text into words for applications like text classification and sentiment analysis.

### 5.2.2 Stop Words

Stop words are terms that are often used in a language and are eliminated during pre-processing to improve the effectiveness of natural language processing (NLP) operations. We used *Bengali* stopwords of NLTK package.

### 5.2.3 Stemming

Stemming, a method of natural language processing used for text classification and information retrieval, reduces words to their fundamental form by stripping away frequent prefixes and suffixes. We used *bangla-stemmer* package for stemming.

### 5.2.4 Lemmatizing

Lemmatization, an NLP approach that groups a word's inflected variants, is frequently used in conjunction with stemming to increase the precision of text analysis. We used *banglakit-lemmatizer* package for lemmatizing.

# Chapter 6

# Methodology

We used a Deep Learning model *long short-term memory networks(LSTM)* to validate and test our dataset which is a variety of Recurrent Neural Network(RNN). We also used four transformer based models - *bert-base-uncased*, *banglabert_large*, *MultiBERTs* and *bert-large-uncased*. Then we constructed a synthetic dataset of size 1,000. To generate this synthetic dataset, we used a pre-trained transformer based Large Language Model(LLM) - GPT-3.5. Finally, we evaluated our generated dataset using the BERT model to evaluate its accuracy.

## 6.1 LSTM

LSTM excels in learning long-term dependencies, making it ideal for sequence prediction tasks like speech recognition and machine translation. To forecast the future, LSTMs utilize a continuously updated cell state, which serves as a memory of previous inputs. Three gates—*forget*, *input*, and *output*—are used by LSTMs to regulate the flow of data into and out of the cell state. The machine is able to retain and retrieve information for long periods of time because LSTM gates are controlled by learnable weights that are modified during training.
In both *premise* and *hypothesis*, we determined the maximum sequence length. The length of the longest sequence in the dataset serves as the maximum sequence length. All sequences must be the same length for the LSTM model, which is achieved through padding, which involves appending zeros to the end of shorter sequences.
We splited the data into three section - training set (60%), validation set(20%) and testing set(20%) and random state of 42. We set 15 epochs and batch size of 64. We creatd word embeddings for the text sequences size of 128. The LSTM model is able to acquire knowledge the relationships between words because of to word embeddings, which represent words as vectors. We add two LSTM layers, one dropout layer value of 0.2, a *'ReLU'* activation fucntion for input layer and a *'softmax'* activation function for dense layer. We use *sparse_categorical_crossentropy* for loss evaluation and *adam* for optimizer.

## 6.2 BERT

The BERT language model, Bidirectional Encoder Representations from Transformers, can comprehend both individual meanings and hidden word relationships.

BERT is a powerful machine learning model that uses contextual information to understand the meaning of words based on the words around them and their arrangement. It can analyze entire sentences at once, providing a broader understanding of linguistic relationships. Due to its pre-trained architecture, BERT can be used for various tasks, including document summarization and question answering.
We splited the data into three section training set (80%), validation set(10%) and testing set(10%) and encoded text labels into numerical value using label encoder. We used the *bert-base-uncased* model and it's tokenizer. We used AdamW optimizer and learning rate of *1e5*. The number of epochs was set to 5. For training datasets, the batch size is set to 32, for testing and validation dataset, it is set to 64. The max_length is set to 128. For training datasets, the optimizer zeroes out gradients, feeds the batch into the model, calculates loss, backpropagates it, and updates model parameters to minimize loss. For validation dataset, the model is set to evaluation mode, initializes lists for tracking labels and predictions, loops through validation batches, calculates validation loss, extracts logits, converts logits to class predictions, and collects true labels and predictions for accuracy calculation. Finally, we compared predictions with true labels and computed accuracy.

## 6.3 MultiBERTs

MultiBERTs-Seed is a collection of large language models for conducting comprehensive studies on the BERT architecture. It provides numerous BERT replicas and 140 intermediate checkpoints for analysing training progression and model stability. This resource assists researchers in identifying unique discoveries for distinct models, distinguishing generalizable features, and investigating the impact of random initialization. We used *google/multiberts-seed_1* model and it's tokenizer and rest of the configuration are similar with BERT's.

## 6.4 BERT large

The BERT-large-uncased model is a widely used and efficient natural language processing (NLP) tool. It is made up of 24 transformer layers, with each layer having 1024 hidden units, and 32 attention heads. This model is helpful in tasks such as question answering, sentiment analysis, and natural language inference. Additionally, it is used for fine-tuning specific tasks. The model's enormous pre-trained text dataset and uncased design provide accuracy and adaptability, making it suitable for various NLP tasks. We used *bert_large_uncased* model and it's tokenizer and rest of the configuration are similar with BERT's.

## 6.5 Banglabert

BUET's pre-trained Bengali language model, BanglaBert, is used for text interpretation, question answering, summarization, sentiment analysis, and natural language inference in Bengali with a focus on understanding context and providing appropriate responses. After being pre-trained on a large corpus of Bengali text, this model outperforms other models in multiple NLP tasks. Researchers and developers can access it as an open-source tool. BUET Bangla BERT enhances Bengali

NLP, enabling efficient use of Bengali language data across various sectors, including education, media, and communication in Bangladesh.
We splited the data into three section training set (80%), validation set(10%) and testing set(10%) and encoded text labels into numerical value using label encoder. We used the *csebuetnlp/banglabert_large* model and it's tokenizer. We provide item access, idx, and inputs, which are used for training and inference. We used AdamW optimizer and learning rate of *1e5*. The max_length is set to 128. The number of epochs was set to 5. For training datasets, the batch size is set to 32, for testing and validation dataset, it is set to 64. For training datasets, the optimizer zeroes out gradients, feeds the batch into the model, calculates loss, backpropagates it, and updates model parameters to minimize loss. For validation dataset, the model is set to evaluation mode, initializes lists for tracking labels and predictions, loops through validation batches, calculates validation loss, extracts logits, converts logits to class predictions, and collects true labels and predictions for accuracy calculation. Finally, we compared predictions with true labels and computed accuracy.

## 6.6 GPT-3.5

Generative Pre-trained Transformer 3.5, also known as GPT-3.5, is an improved large language model that offers better performance, flexibility, and control in comparison to prior GPT models. GPT-3.5 is an enormous model that can generate high-quality text and code. Trained on 175 billion parameters, it can produce human-like writing, translate languages, and offer helpful responses. Its fine-tuning capabilities improve its accuracy and performance, making it ideal for real-world applications. It also allows for customisable outputs. GPT-3.5 is capable of generating realistic chatbot conversations, unique written forms, accurate language translations, summarizing large documents, and answering open-ended, difficult or unusual queries.
We used openai library of version 0.28. We used the model to predict GPT-3.5 entailment by creating a prompt for GPT-3.5-turbo, submitting it to GPT-3.5, restricting the output length to 280 tokens, and tweaking inventiveness. The engine is set to *text-davinci-003* and temperature is 0.7. This function extracts premises and hypotheses from training data. It generates a hypothesis for each premise of test data, creates a list, and adds it to generated datasets. The total number of entailment in this dataset is 307, contradiction is 260 and neutral is 432.

| A sentence | A output | A label | # Similarity |
|---|---|---|---|
| একজন মহিলা একটি স্টোরের বাইরে চেয়ার এবং টেবিলের কাছে দাঁড়িয়ে আছেন। | তারা ব্যক্তিগত সমস্যাগুলি আলোচনা করছে । | contradiction | 0.06843010618130085 |
| দু'জন যুবতী ফুটপাতের টেবিলে বসে আছেন। | তারা হচ্ছে ফুটবলার। | contradiction | 0.07446998240708609 |
| দু'জন নার্স একটি বেঞ্চে বসে একটি বাসের জন্য অপেক্ষা করছে। | একজন নার্সের নাম হচ্ছে আইয়া। | contradiction | 0.2717542354221822 |
| তিন মহিলা একটি সৈকতে হাত ধরে। | তারা সঙ্গীত শুনতে গিয়েছেন । | contradiction | -0.008731126421353666 |
| পাতাল রেলপথে চার জন মজা করছেন। | তাদের সাথে একজন আত্মীয় ছিল । | contradiction | 0.09805336703259689 |
| দু'জন লোক সৈকতে সকার বল লাথি মারছে। | দুই লোক সৈকতে সকার বল মারছে। | entailment | 1 |

Figure 6.1: Dataset generated by GPT-3.5

# Chapter 7

# Results

## 7.1 Human Generated Data

We utilized the *bert_base_uncased* model on a dataset generated by human annotators and achieved an accuracy of 63.05%. This result is particularly telling about the model's capabilities in distinguishing between different types of textual relationships. Notably, the model demonstrated a higher accuracy in classifying text as entailment and contradiction compared to its performance in classifying neutral text. This suggests that the model is more adept at identifying clear relationships, whether they are of agreement (entailment) or disagreement (contradiction), rather than texts where no such clear relationship is apparent.
The model's recall for entailment was 0.76, indicating that it correctly identified 76% of the samples that were actual instances of entailment. This high recall rate is significant as it shows the model's strength in correctly retrieving a high proportion of relevant instances, thereby reducing the chances of missing out on true instances of entailment.
However, the precision for neutral text classification stood at 0.56. This means that only 56% of the text samples that the model classified as neutral were actually neutral, highlighting a relative weakness in the model's ability to accurately identify texts that do not have a clear entailment or contradiction relationship. The lower precision in this category could be due to the inherent subtlety and complexity in categorizing texts as neutral, which often requires a deeper understanding of context and the absence of bias.

Table 7.1: Accuracy of BERT base for human generated data

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Entailment | 0.77 | 0.76 | 0.76 | 148 |
| Contradiction | 0.62 | 0.68 | 0.65 | 280 |
| Neutral | 0.56 | 0.51 | 0.53 | 273 |
| Accuracy | | | 0.63 | 701 |
| Macro avg | 0.65 | 0.65 | 0.65 | 701 |
| Weighted avg | 0.63 | 0.63 | 0.63 | 701 |

*Total accuracy : 0.630528*

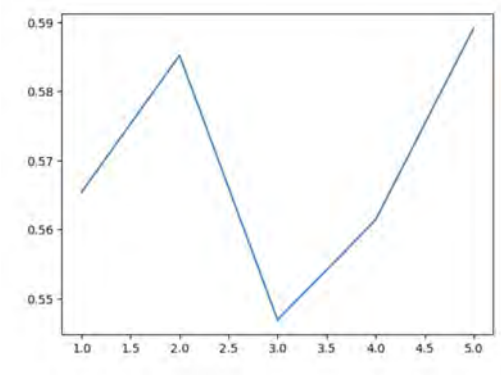
### 7.1.1 LSTM

we implemented a Long Short-Term Memory (LSTM) network to examine its learning capabilities with our dataset. The results revealed a validation accuracy of 44.19%, which is notably lower than the model's training accuracy of 59.86%. This discrepancy suggests a potential overfitting to the training data or a lack of generalization to new, unseen data. Additionally, the overall data loss recorded was 1.0845, indicating the average loss per batch during the training process, which is a critical factor in evaluating the model's efficiency in learning and making predictions.
One of the key metrics we employed to assess the model's performance was the F1-score, which is particularly valuable as it balances precision (the proportion of true positive predictions among all positive predictions) and recall (the proportion of true positive predictions among all actual positives). A higher F1-score is indicative of a model's better performance in predicting both positive and negative cases accurately.
For this LSTM model, the F1-scores obtained for different classifications were insightful. The model achieved F1-scores of 0.45 for both entailment and contradiction, which suggests a moderate level of prediction accuracy in these categories. These scores indicate that the model is relatively competent in identifying instances of clear textual relationships, whether they are agreements entailment or disagreements (contradiction).
However, the F1-score for neutral classification was significantly lower, at 0.31. This lower score implies that the model struggles more with accurately predicting neutral cases, which are instances where neither clear agreement nor disagreement is expressed. This is a common challenge in NLP tasks, as neutral cases often require a more nuanced understanding of the text, free from biases towards positive or negative categorizations.

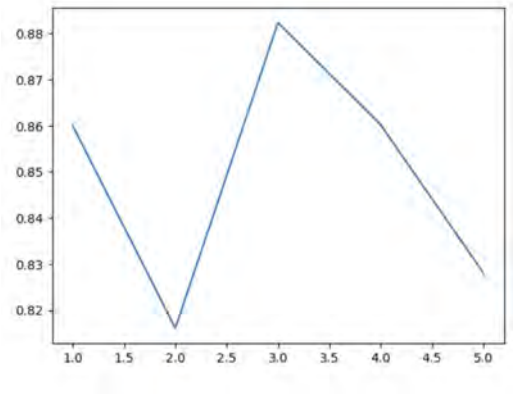
```
40/40 [==============================] - 0s 6ms/step - loss: 1.0845 - accuracy: 0.4420
Final Test Accuracy: 44.1971%
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_4 (Embedding)     (None, 52, 128)           654976

 dense_9 (Dense)             (None, 52, 1)             129

 dropout_4 (Dropout)         (None, 52, 1)             0

 lstm_8 (LSTM)               (None, 52, 128)           66560

 lstm_9 (LSTM)               (None, 64)                49408

 dense_10 (Dense)            (None, 4)                 260

=================================================================
Total params: 771,333
Trainable params: 771,333
Non-trainable params: 0
```

Figure 7.1: Accuracy

The model's training accuracy improves with each epoch, suggesting that it has learned from the training data. However, its validation accuracy plateaus at roughly 70%, indicating overfitting to training data. A small difference between training and validation accuracy indicates overfitting. The model's overall patterns, such as reduced training and validation loss, suggest learning and progress. Overfitting, a large gap between lines, and early ending may all indicate overfitting, implying early intervention.




(a) Accuracy graph of human generated data




(b) Loss graph of human generated data


Figure 7.3: Model Report

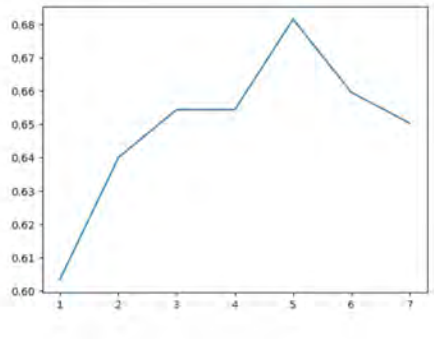
Table 7.2: Model Report of LSTM

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Entailment | 0.51 | 0.41 | 0.45 | 623 |
| Contradiction | 0.38 | 0.56 | 0.45 | 630 |
| Neutral | 0.37 | 0.26 | 0.31 | 632 |
| *Total accuracy : 0.441971* | | | | |

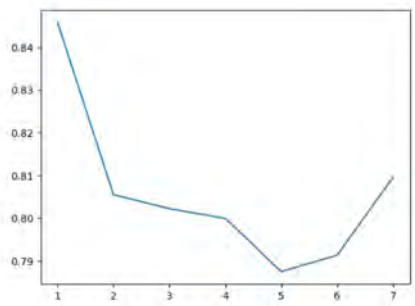
## 7.2 Extended Dataset with First Thresholding

### 7.2.1 BERT Base

After the application of the *bert_base_uncased* model to our dataset, we observed an overall accuracy of 68.45%. This level of performance is particularly significant in the context of textual entailment tasks. The model exhibited its strongest performance in classifying entailment, with a noteworthy accuracy in this category. Specifically, it correctly identified 73% of sentences as entailment, reflecting a precision rate of 0.73 for this label. Precision, in this context, refers to the proportion of true positive predictions among all positive predictions, indicating the model's effectiveness in accurately identifying genuine instances of entailment.
However, the model faced challenges in determining the precision for neutral classifications. This difficulty highlights a common issue in natural language processing (NLP), where categorizing sentences as neutral often requires a nuanced understanding of language, beyond clear-cut positive or negative delineations.
In terms of recall, which measures the proportion of true positive predictions among all actual positives, the model also scored 0.73 for entailment. This means it correctly




(a) Accuracy graph of BERT base




(b) Test and validation loss of BERT base

classified 73% of the actual entailment sentences present in the dataset. A recall of this level demonstrates the model's robustness in retrieving a high proportion of relevant instances.
Furthermore, the model achieved a harmonic mean of precision and recall (the F1-score) of 0.73 for entailment, indicating a balanced and strong performance between precision and recall. This metric is crucial as it provides a more holistic view of the model's effectiveness, considering both false positives (precision) and false negatives (recall).

Table 7.3: Accuracy of BERT Base

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Entailment | 0.73 | 0.73 | 0.73 | 569 |
| Contradiction | 0.71 | 0.70 | 0.70 | 633 |
| Neutral | 0.63 | 0.64 | 0.63 | 716 |
| Accuracy | | | 0.68 | 1918 |
| Macro avg | 0.69 | 0.69 | 0.69 | 1918 |
| Weighted avg | 0.69 | 0.68 | 0.68 | 1918 |

*Total accuracy : 0.684567*

### 7.2.2 BanglaBERT

The BanglaBert model, tailored specifically for the Bengali language, achieved an overall accuracy of 51.82%. Notably, the model reached its peak performance with a maximum test accuracy of 53.02% during the third epoch of training. However, the variation in validation loss across the training epochs suggests that there is considerable room for improvement in the model's training process.
Throughout the five epochs, the test accuracy of the model demonstrated fluctuations, ranging from 47.02% to 53.02%. The highest accuracy achieved on epoch 3 points to the model's potential, but it also underscores the need for further optimization or exploration of hyperparameters. Such fluctuations in accuracy over different epochs may be indicative of the model's sensitivity to certain aspects of the dataset or training regimen.
The validation loss, a critical metric for assessing the model's performance on unseen data, also exhibited variability. It recorded a minimum value of 1.0155 during epoch

4. This fluctuation in validation loss is particularly noteworthy as it suggests the model's sensitivity to specific data characteristics. This can be a concern because a model that is too sensitive to the training data may not generalize well to new, unseen data.
Moreover, the observed fluctuations in validation loss highlight the importance of close monitoring of the model during training. It is crucial to ensure that the model does not overfit to the training data, which would diminish its effectiveness on real-world data. Overfitting occurs when a model learns the details and noise in the training data to the extent that it negatively impacts the model's performance on new data.

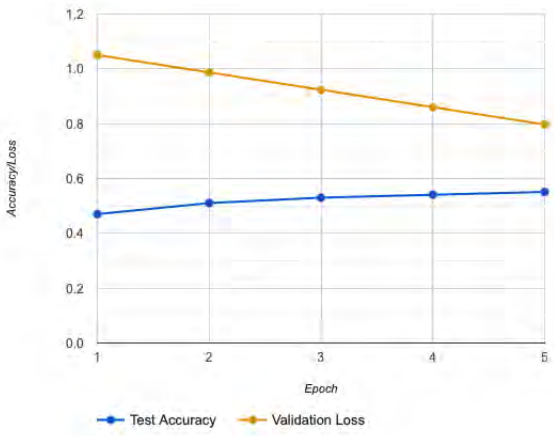



Figure 7.5: Test and Validation loss of BanglaBERT

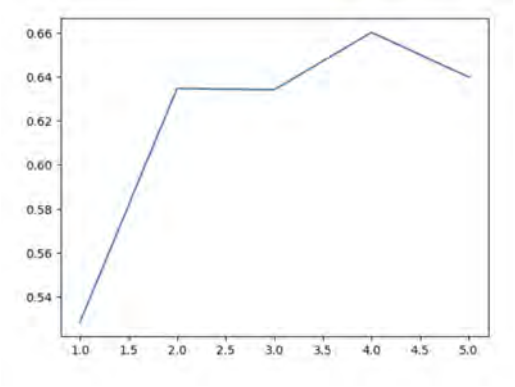
Table 7.4: Accuracy of BanglaBERT

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Entailment | 0.56 | 0.58 | 0.58 | 569 |
| Contradiction | 0.53 | 0.52 | 0.53 | 633 |
| Neutral | 0.48 | 0.49 | 0.48 | 716 |
| Accuracy | | | 0.65 | 1918 |
| Macro avg | 0.53 | 0.52 | 0.52 | 1918 |
| Weighted avg | 0.52 | 0.52 | 0.52 | 1918 |

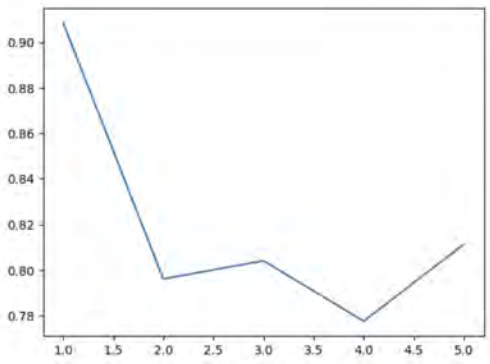
*Total accuracy : 0.518219*

### 7.2.3 BERT large

The BERT large model, tailored for complex language processing tasks, achieved an overall accuracy of 64.7%. This performance is particularly noteworthy in the context of entailment classification, where the model exhibited its strongest capabilities. Specifically, it accurately classified 73% of sentences as entailment. However, the precision for this label was slightly lower, at 0.65. Precision in this context refers to the proportion of true positive predictions among all positive predictions,




(a) Accuracy graph of BERT large




(b) Test and validation loss of BERT large

suggesting that while the model is generally reliable in identifying entailment, there is still room for improvement in reducing false positives.
One area where the model faced challenges was in determining the precision for neutral text classifications. This difficulty is not uncommon in natural language processing, as neutral text often requires nuanced interpretation and can be easily confused with subtle instances of entailment or contradiction.
In terms of recall, the model performed exceptionally well, correctly classifying 85% of the actual entailment sentences in the dataset, as indicated by a recall rate of 0.85. Recall measures the model's ability to find all relevant instances within a dataset, so a high recall in entailment indicates the model's effectiveness in identifying true instances of entailment.
The model also achieved an impressive harmonic mean of precision and recall (F1-score) of 0.85 for entailment. The F1-score is a critical metric as it provides a balanced measure of the model's precision and recall, making it a reliable indicator of the model's overall performance in correctly classifying entailment.

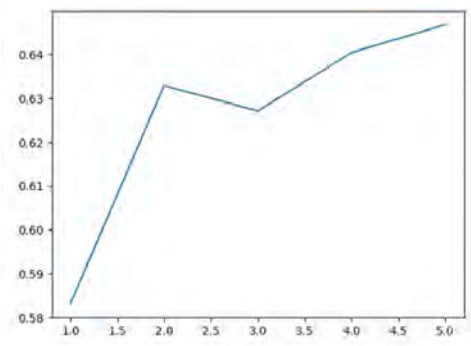
Table 7.5: Accuracy of BERT Large

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Entailment | 0.65 | 0.85 | 0.73 | 569 |
| Contradiction | 0.64 | 0.77 | 0.70 | 633 |
| Neutral | 0.67 | 0.38 | 0.49 | 716 |
| Accuracy | | | 0.65 | 1918 |
| Macro avg | 0.65 | 0.67 | 0.64 | 1918 |
| Weighted avg | 0.65 | 0.65 | 0.63 | 1918 |

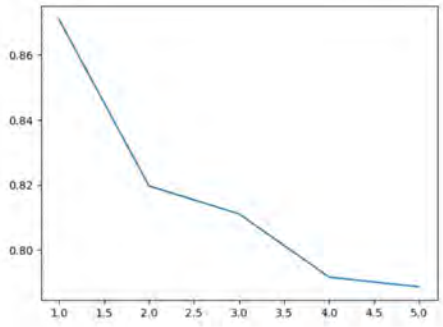
*Total accuracy : 0.684567*

### 7.2.4 MultiBERT

The MultiBERT model, a variant of the BERT architecture designed for enhanced performance across multiple tasks, achieved an overall accuracy of 66.73% in our analysis. This level of accuracy is noteworthy, particularly in the context of textual entailment, where the model displayed its strongest capabilities.




(a) Accuracy graph of MultiBERT




(b) Test and validation loss of MultiBERT

In the specific task of classifying entailment, the MultiBERT model accurately identified 72% of sentences as entailment, which is indicative of its adeptness in handling this category. The precision for entailment was recorded at 0.71. This metric, which represents the proportion of true positive predictions among all positive predictions, suggests that while the model is generally effective at identifying true instances of entailment, it also includes a moderate number of false positives in its predictions.
However, the model encountered some challenges in determining the precision for neutral text. This difficulty in accurately classifying neutral sentences is a common challenge in natural language processing, requiring a nuanced understanding of context and subtlety in language.
The model demonstrated a high recall of 0.72 for entailment, indicating that it correctly classified 72% of the actual entailment sentences in the dataset. Recall is a crucial metric as it measures the model's ability to identify all relevant instances within the dataset. A high recall in entailment suggests the model's effectiveness in capturing true instances of this category.
Moreover, the MultiBERT model achieved an impressive harmonic mean of precision and recall (F1-score) of 0.72 for entailment. The F1-score is an important measure as it provides a balanced view of the model's precision and recall, making it a reliable indicator of overall performance in correctly classifying entailment.

Table 7.6: Accuracy of MultiBERT

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Entailment | 0.71 | 0.72 | 0.72 | 569 |
| Contradiction | 0.74 | 0.62 | 0.67 | 633 |
| Neutral | 0.59 | 0.67 | 0.62 | 716 |
| Accuracy | | | 0.67 | 1918 |
| Macro avg | 0.68 | 0.67 | 0.67 | 1918 |
| Weighted avg | 0.68 | 0.67 | 0.67 | 1918 |

*Total accuracy : 0.667362*

### 7.2.5 Comparison of Models

Out of the five models tested in our study, the transformer-based and the BERT-based models demonstrated superior accuracy, significantly outperforming the others, particularly the LSTM model, which showed noticeably poorer accuracy. This

disparity highlights the advancements in language modeling techniques, where newer architectures like transformers have proven to be more effective in understanding and processing complex language patterns.
We also made an interesting observation regarding the performance of different BERT variations. Each variant of BERT, with the notable exception of Banglabert large, showed enhanced performance metrics. This suggests that while the general architecture of BERT is advantageous for textual entailment tasks, specific adaptations or versions may not always yield the same level of efficacy, especially in the context of the Bengali language. The underperformance of BanglaBERT large, in particular, raises questions about the scalability and adaptability of certain BERT models for specific linguistic challenges.
Furthermore, a critical finding of our study was the notable improvement in the performance of the BERT-based model when the datasets were expanded. This correlation between dataset size and accuracy is particularly significant. It underscores the importance of large, diverse, and well-structured datasets in training more effective machine learning models for language processing. Larger datasets likely provide a more comprehensive representation of the language's nuances, idiomatic expressions, and syntactical variations, allowing models like BERT to learn and generalize better.

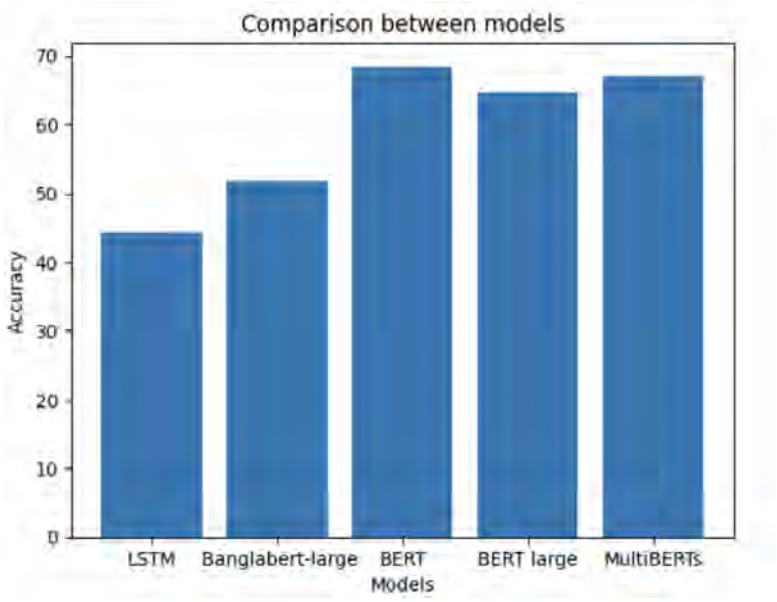
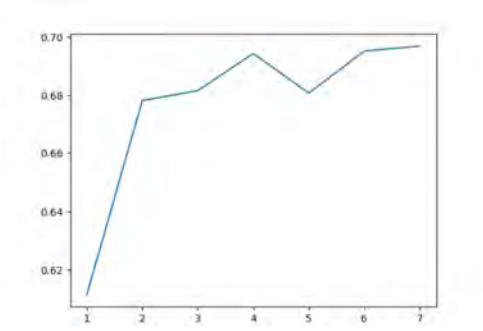



Figure 7.8: Comparison of Models with first threshold

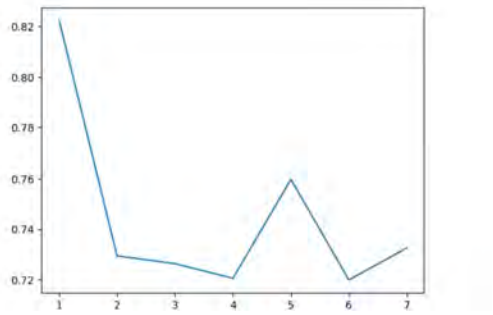
## 7.3 Extended Data with Second Thresolding

### 7.3.1 BERT base

After implementing BERT base with the second thresholding condition, we achieved a commendable 70.90% accuracy, showcasing the model's robust performance. The emphasis on the "entailment" category was particularly noteworthy, with an impressive F1-score of 0.76, recall of 0.73, and precision of 0.80, underscoring the model's proficiency in capturing nuanced relationships. Equally noteworthy were




(a) Accuracy of BERT base with second thresholding




(b) Validation loss of BERT with second thresholding

the positive outcomes in the "contradiction" category, where the model exhibited a well-balanced performance, achieving an F1-score of 0.76, recall of 0.76, and precision of 0.75. However, the model encountered challenges in the "neutral" category, demonstrating a comparatively lower F1-score of 0.62, recall of 0.64, and precision of 0.60. These insights provide valuable context for further refinement, suggesting potential areas of focus for enhancing the model's overall performance and applicability.

Table 7.7: Accuracy of BERT Base with second thresholding data

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Contradiction | 0.75 | 0.76 | 0.76 | 350 |
| Entailment | 0.80 | 0.73 | 0.76 | 379 |
| Neutral | 0.60 | 0.64 | 0.62 | 388 |
| Accuracy | | | 0.71 | 1117 |
| Macro avg | 0.72 | 0.71 | 0.71 | 1117 |
| Weighted avg | 0.71 | 0.71 | 0.71 | 1117 |

*Total accuracy : 0.653285*

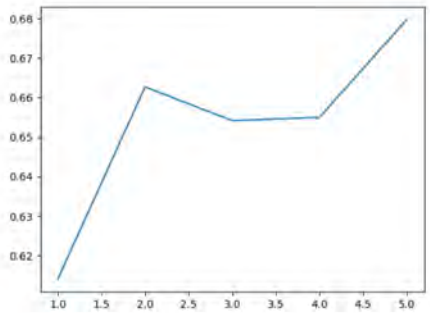
### 7.3.2 BERT Large

Achieving a commendable 67.5% accuracy post the integration of the BERT large model at the specified threshold is a notable success. The detailed evaluation metrics further affirm the model's proficiency, particularly excelling in the "entailment" category with a precision of 0.78, recall of 0.68, and an F1-score of 0.73. Despite a comparatively lower performance in the "neutral" category, the model compensates with its strong performance in the "contradiction" domain. This nuanced analysis underscores the model's robust ability to effectively classify text into the three designated groups. The balance between precision, recall, and F1-score reflects the model's versatility and suggests its reliability in handling diverse text inputs. Overall, the outcomes underscore the model's competency in text categorization, paving the way for its potential application in various natural language processing tasks.

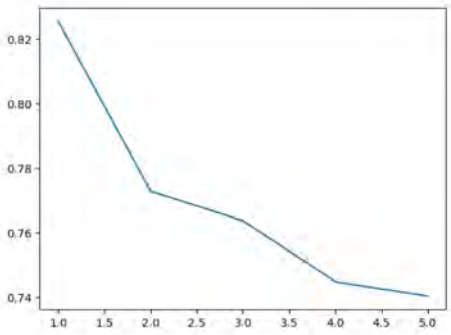
Table 7.8: Accuracy of BERT Large with second thresholding data

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Contradiction | 0.68 | 0.83 | 0.75 | 350 |
| Entailment | 0.78 | 0.68 | 0.73 | 379 |
| Neutral | 0.58 | 0.53 | 0.55 | 388 |
| Accuracy | | | 0.68 | 1117 |
| Macro avg | 0.68 | 0.68 | 0.67 | 1117 |
| Weighted avg | 0.68 | 0.68 | 0.67 | 1117 |

*Total accuracy : 0.675022*




(a) Accuracy of BERT large with second thresholding




(b) Validation loss of BERT large with second thresholding

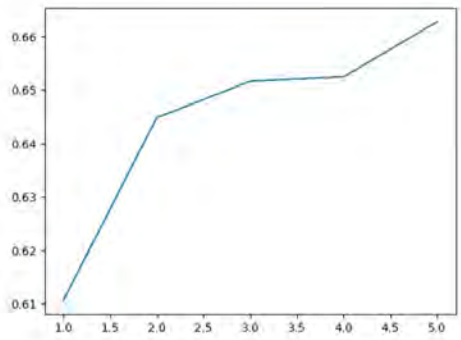
### 7.3.3 MultiBERT

The model exhibits commendable performance across various categories, showcasing its prowess in text classification. Particularly noteworthy is its exceptional performance in the "entailment" category, boasting a precision, recall, and F1-score all registering at an impressive 0.86. Furthermore, the model demonstrates proficiency in handling contradictions, achieving a respectable F1-score of 0.71, recall of 0.75, and precision of 0.67 in the "contradiction" category. However, its effectiveness diminishes in the "neutral" category, where it yields an F1-score of 0.54, recall of 0.48, and precision of 0.63. Despite this, the model maintains an overall accuracy of 69.02%, showcasing its ability to reliably categorize text into the three specified categories. This nuanced evaluation underscores the model's strengths and areas for potential improvement, contributing to a comprehensive understanding of its classification capabilities.

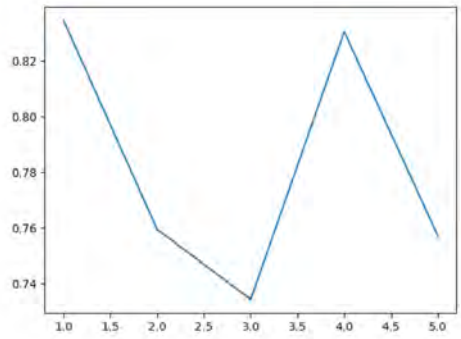
Table 7.9: Accuracy of MultiBERT with second thresholding data

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Contradiction | 0.67 | 0.86 | 0.75 | 350 |
| Entailment | 0.76 | 0.75 | 0.75 | 379 |
| Neutral | 0.63 | 0.48 | 0.54 | 388 |
| Accuracy | | | 0.69 | 1117 |
| Macro avg | 0.69 | 0.70 | 0.68 | 1117 |
| Weighted avg | 0.69 | 0.69 | 0.68 | 1117 |

*Total accuracy : 0.690242*




(a) Accuracy of MultiBERT with second thresholding




(b) Validation loss of MultiBERT large with second thresholding

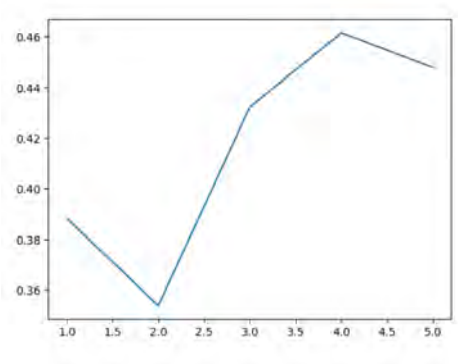
### 7.3.4 BanglaBERT

The overall accuracy of the model stands at 44.4%, reflecting its ability to correctly classify texts within the test set. Specifically, in the "contradiction" category, the precision is at 79%, indicating that 79% of texts identified as contradictory by the model were indeed contradictory. However, the model's performance is hindered by a recall of only 32% in the "contradiction" category, signifying its limited capability to identify true inconsistencies within the test set. The F1-score, an amalgamation of precision and recall, reinforces the model's struggle in the "contradiction" category, yielding a score of 0.46. These metrics collectively underscore the model's suboptimal performance in accurately capturing and categorizing contradictions, revealing areas for potential improvement in its classification abilities.

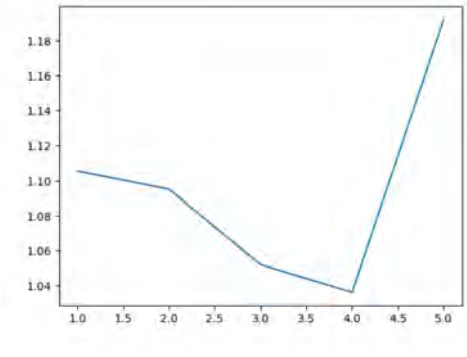
Table 7.10: Accuracy of BanglaBERT with second thresholding data

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Contradiction | 0.79 | 0.32 | 0.46 | 350 |
| Entailment | 0.25 | 0.11 | 0.16 | 379 |
| Neutral | 0.42 | 0.88 | 0.57 | 388 |
| Accuracy | | | 0.44 | 1117 |
| Macro avg | 0.49 | 0.44 | 0.48 | 1117 |
| Weighted avg | 0.48 | 0.44 | 0.40 | 1117 |

*Total accuracy : 0.444047*




(a) Accuracy of BanglaBERT with second thresholding




(b) Validation loss of BanglaBERT large with second thresholding

Based on the results, the model's performance is not very good. It has low precision, recall, and F1-scores for both contradiction and entailment detection. Additionally, the overall accuracy is quite low.

### 7.3.5 LSTM

Despite achieving a commendable 45% accuracy with the implemented LSTM model and maintaining a relatively consistent F1-score of approximately 0.5 across all three categories, a notable challenge arises in accurately categorizing texts labeled as "neutral." The model exhibits a lower recall rate of 0.18 for this specific category, indicating a tendency to misclassify neutral texts as either contradiction or entailment. While the overall performance appears promising, addressing this particular issue is crucial for enhancing the model's precision and reliability, especially in scenarios where neutrality plays a pivotal role. Further refinement and fine-tuning of the model may be necessary to bolster its capability in distinguishing and correctly categorizing texts within the neutral category, thereby improving its overall efficacy.

Table 7.11: Accuracy of LSTM with second thresholding data

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Contradiction | 0.53 | 0.52 | 0.53 | 414 |
| Entailment | 0.41 | 0.66 | 0.50 | 414 |
| Neutral | 0.41 | 0.18 | 0.25 | 426 |
| Accuracy | | | 0.45 | 1254 |
| Macro avg | 0.45 | 0.45 | 0.43 | 1254 |
| Weighted avg | 0.45 | 0.45 | 0.42 | 1254 |

*Total accuracy : 0.451235*

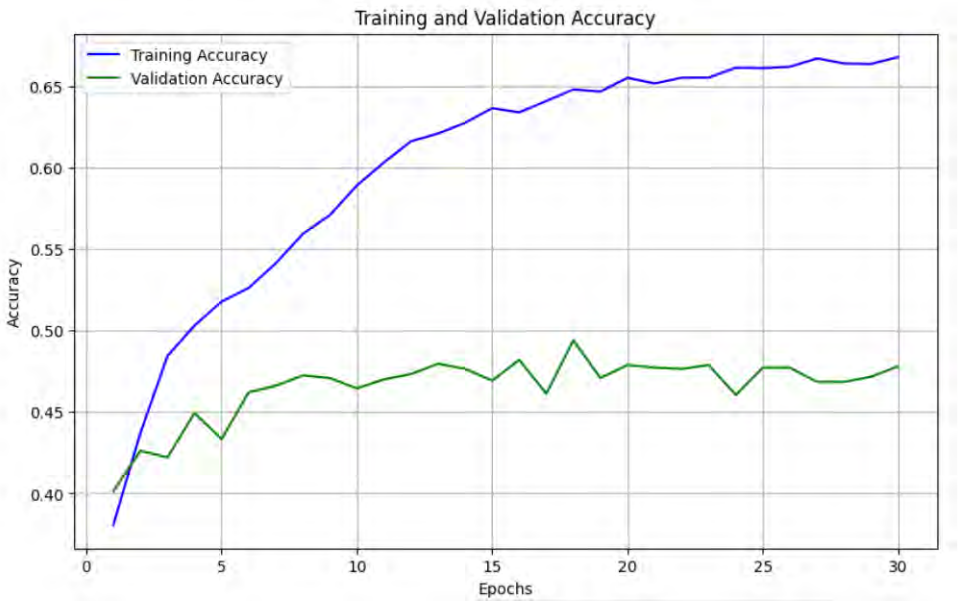



Figure 7.13: Accuracy of LSTM with second thresholding

### 7.3.6 Comparison of Models

The research outcomes underscore the remarkable strides made in language modeling techniques, revealing the supremacy of transformer-based and BERT models over the LSTM model in terms of accuracy. The variations within the BERT model demonstrated notable advancements, showcasing enhanced performance metrics. However, the study unearthed a significant challenge with the scalability and adaptability of BanglaBERT large, prompting a critical examination of its limitations. Questions arise concerning its underperformance and the potential hurdles in achieving broader applications.

Moreover, an intriguing revelation emerged when examining the impact of dataset size on model performance. The study emphasized the substantial improvement in BERT-based model accuracy with the expansion of datasets. This underscores the pivotal role of large, diverse, and well-structured datasets in training effective machine learning models for language processing tasks. Notably, the BERT base model emerged as the top performer in the study, outshining others in this particular context. Additionally, BERT large and MultiBERT demonstrated commendable performance, further solidifying the effectiveness of BERT-based approaches. In contrast, BanglaBERT lagged behind, emerging as the least performing model within the specified threshold, prompting further exploration into its limitations and potential areas for improvement in future language modeling endeavors.

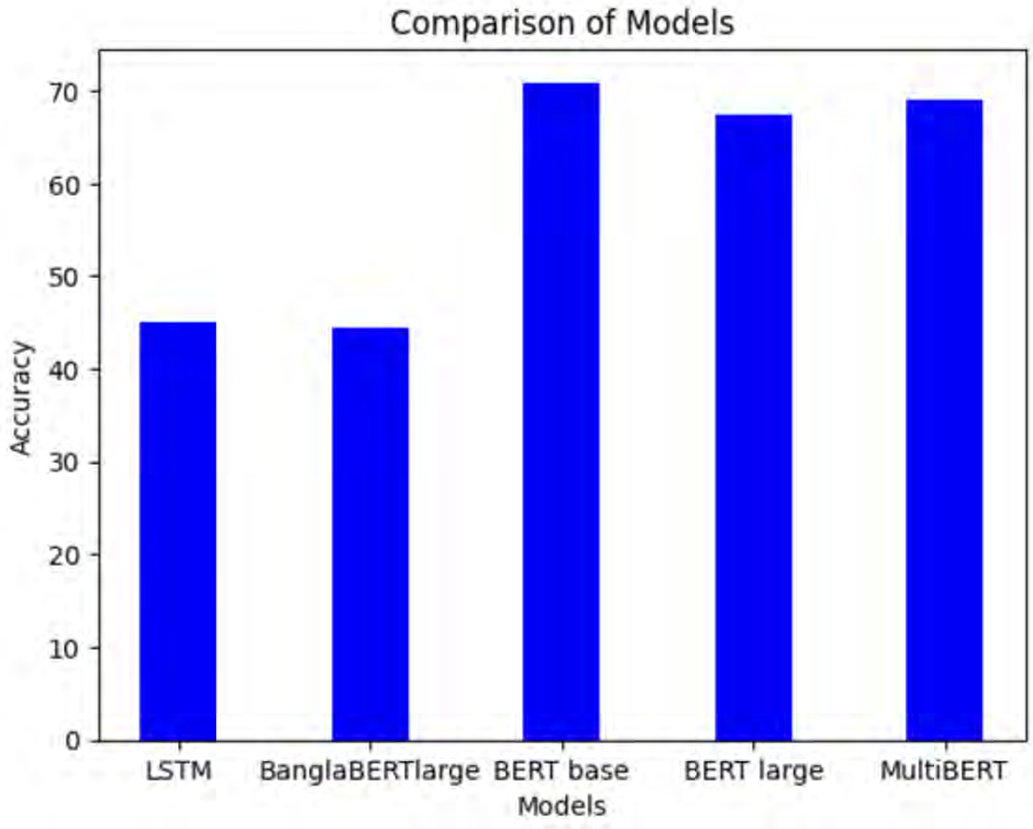



Figure 7.14: Comparison of Models with second threshold

## 7.4 Synthetic Data

We apply the transformer-based model BERT to assess the accuracy of the synthetic dataset.

### 7.4.1 BERT base with first threshold

After applying the *bert_base_uncased* model to our dataset, we observed an overall accuracy of 59.35%. This level of performance indicates that the model is particularly adept at recognizing and interpreting entailment in sentences. Notably, it accurately classified 75% of sentences as entailment, demonstrating a precision of 0.75 for this specific label. This precision rate is noteworthy as it underscores the model's capability in correctly identifying true positive cases of entailment, a crucial aspect in the context of textual entailment tasks. However, the model exhibited some challenges in determining the precision of sentences labeled as neutral. This suggests a potential area of focus for further refinement, as the ability to distinguish neutral cases from entailment and contradiction is vital for comprehensive understanding in NLP tasks.
In terms of recall, the model correctly classified 52% of sentences that were true entailments in the dataset, which is reflected in a recall rate of 0.52. While this indicates room for improvement, it's important to consider that recall measures the model's ability to find all relevant cases within the dataset, a task that often presents complexity in nuanced linguistic environments.
The harmonic mean of precision and recall (F1 score) for entailment stood at 0.73,

signifying a balanced and good performance between precision and recall. This metric is particularly relevant as it provides a more holistic view of the model's effectiveness, considering both false positives and false negatives.
Overall, these results suggest that the *bert_base_uncased* model has demonstrated a commendable performance on the test dataset, especially in identifying and classifying entailment. However, the findings also highlight potential areas for further research and model tuning, particularly in enhancing the model's ability to discern neutral cases and in improving recall for entailment. This insight forms a valuable basis for future efforts aimed at optimizing the model for better performance in textual entailment tasks, especially in the context of the Bengali language.

Table 7.12: Accuracy of BERT for generated data with first threshold

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Contradiction | 0.73 | 0.32 | 0.45 | 260 |
| Entailment | 0.75 | 0.52 | 0.61 | 307 |
| Neutral | 0.52 | 0.81 | 0.64 | 432 |
| Accuracy | | | 0.59 | 999 |
| Macro avg | 0.67 | 0.55 | 0.56 | 999 |
| Weighted avg | 0.65 | 0.59 | 0.58 | 999 |

*Total accuracy : 0.593594*

### 7.4.2 BERT base with second threshold

The model's evaluation metrics, including precision, recall, and F1-score, provide valuable insights into its performance across different categories. Impressively, the model exhibits strong capabilities in correctly classifying instances falling under the "entailment" and "contradiction" categories, with all values exceeding 0.6. This suggests a high degree of accuracy in identifying text that either supports or contradicts a given statement.
However, a notable contrast emerges when assessing the model's performance in the "neutral" category. Both precision and recall values fall below the 0.6 threshold, indicating a suboptimal ability to correctly classify neutral texts. The lower precision suggests that instances labeled as "neutral" by the model may include a significant number of misclassifications, while the lower recall indicates that the model misses a notable portion of true "neutral" instances. This discrepancy underscores a challenge in the model's understanding of neutral statements, leading to occasional misclassification as either "contradiction" or "entailment."
The overall accuracy of the model, reported at 60%, further reinforces this observation. While the model achieves a commendable accuracy rate, it also indicates that around 40% of the sentences in the test dataset are misclassified. This discrepancy primarily stems from the model's struggle with the "neutral" category, as reflected in the lower precision and recall values.
Examining the macro and weighted averages of the precision, recall, and F1-score provides a broader perspective on the model's performance. The consistency of these metrics around the 0.6 mark for the "contradiction" and "entailment" categories implies a balanced performance across both classes. However, the challenge persists in the "neutral" category, where the model's difficulty is further highlighted.

In summary, while the model excels in categorizing texts as "entailment" or "contradiction," it encounters challenges in accurately classifying instances as "neutral." Addressing this issue could significantly enhance the model's overall performance and broaden its applicability across diverse text classifications.

Table 7.13: Accuracy of BERT for generated data with second threshold

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| Contradiction | 0.73 | 0.31 | 0.43 | 260 |
| Entailment | 0.69 | 0.67 | 0.68 | 307 |
| Neutral | 0.54 | 0.73 | 0.62 | 432 |
| Accuracy | | | 0.60 | 999 |
| Macro avg | 0.65 | 0.57 | 0.58 | 999 |
| Weighted avg | 0.63 | 0.60 | 0.59 | 999 |

*Total accuracy : 0.602603*

# Chapter 8

# Conclusion

This thesis delves into the fascinating field of textual entailment, in which deep learning models try to comprehend the complex links between propositions. This thesis examined the effectiveness of deep learning algorithms for dealing with textual entailment in the Bengali language. We used a two-pronged strategy to examine the accuracy of current entailment datasets when used with some deep learning models BERT, BanglaBERT, MultiBERT, BERT large. This early research offered an important baseline, establishing AI's current skills in recognising the various intricacies of textual meaning and inference. The study explored the generation of synthetic Bengali entailment data. We have found that some of our models performed really well for Bengali language.
Building on this foundation, we moved into previously unexplored areas with GPT-3.5, utilising its generative capabilities to produce novel datasets optimised for textual entailment research. This brave step pushed the field's frontiers, demonstrating AI's ability to not just analyse but also augment the data on which it lives. Evaluating the accuracy of these generated datasets on deep learning models such as BERT provided insight into GPT-3.5's effectiveness and potential biases.
The findings of our research have important implications for the future of natural language processing. We have laid the ground for future advances in AI understanding and reasoning by examining the benefits and limits of deep learning and GPT-3.5 in dealing with textual entailment. Our work opens the doors to:

- Analysing performance on created datasets can help optimise deep learning models, enhancing accuracy and generalizability for real-world applications.
- Evaluating the created datasets reveals important information about the potential biases inherent in GPT-3.5's text production, ultimately guiding responsible development and deployment of this powerful language model.
- Investigating the synergy between AI models, our research highlights the potential for combining deep learning and generative models, such as GPT-3.5, to create new AI architectures capable of enhanced natural language understanding.

Deep learning models demonstrated promising accuracy for Bengali textual entailment on existing datasets, while GPT-3.5-generated data augmentation may overcome constraints in real-world datasets. Deep learning has demonstrated the potential for Bengali textual entailment, but obstacles remain. GPT-3.5 data augmentation is useful, however further research is needed to improve generation methods.

This study improves Bengali NLP by leveraging deep learning for textual entailment, paving the way for Bengali-specific NLP tools and applications that will assist machine translation, information retrieval, and sentiment analysis.

## 8.1 Key Findings

Here are the key findings of this work :

- Accuracy from only human generated data which was done by the students was nearly 64%
- After increasing the data it went to 68.45% using the same model.
- Increasing the data is increasing the accuracy accordingly.
- According to the f1 score of all classification models , it seems that detection of neutral is complicated and it gives the lowest accuracy rate.
- The reason behind neutral's low percentage might be, it is confusing and people also get confused with the neutrals classification. Because it holds both the sense of entailment and contradiction.
- In the generated dataset with GPT 3.5 , accuracy of the contradiction class is the lowest.
- The contradiction data generation with the generative model might not be good

## 8.2 Novelty

This thesis explores new ground by combining text generated by GPT-3.5 with manually chosen data. We employ a mixed approach to gather data, blending curated data for high-quality training with data generated by GPT-3.5 to diversify our dataset. Additionally, we investigate methods to refine GPT-3.5 for text generation that is pertinent to textual entailment. We also perform a comparative analysis of BERT model performance using data collected manually and data generated by GPT-3.5.
Additionally, this work examines the advantages of comparing performance across different sets of data and highlights the advantages of each model. It explains how transformer or BERT architectures were modified or adjusted for a particular task, emphasising if these adjustments increased performance or created new research opportunities.
Furthermore, there isn't a textual entailment dataset for Bengali currently in existence. The absense of Bengali data has hindered previous research in this field, despite the fact that Bengali is spoken by almost 250 million people worldwide. Our datasets will pave the way for further advancements in Bengali NLP.

## 8.3 Future Works

There is potential to expand on the current work by exploring the use of other generative models alongside GPT-3.5 to find the best fit for BERT's entailment task. Additionally, it would be beneficial to fine-tune prompts and parameters, and assess data quality through either human judgment or automatic metrics to ensure accuracy.
The findings can be used to create applications based on entailment, analyze bias and fairness, and explore how texts in different languages are related in terms of entailment. Additionally, there is an option to develop methods that can help reduce bias and ensure fair performance across different domains and contexts.

In conclusion, our thesis has made an important contribution to the field of textual entailment. By studying the interaction between deep learning with GPT-3.5, we have shed light on the intricacies of AI's understanding of meaning and inference, providing significant insights for future advances in natural language processing. The trail we have blazed promises to reveal the delicate dance between AI and human language one text relationship at a time.

# Bibliography


[1] O. Levy, T. Zesch, I. Dagan, and I. Gurevych, "Recognizing partial textual entailment," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 451–455. [Online]. Available: https://aclanthology.org/P13-2080.

[2] A. Gupta, M. Kaur, S. Mirkin, A. Singh, and A. Goyal, "Text summarization through entailment-based minimum vertex cover," in *International Workshop on Semantic Evaluation*, 2014.

[3] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015. DOI: 10.18653/v1/d15-1075.

[4] L. Sha, S. Li, B. Chang, Z. Sui, and T. Jiang, "Recognizing textual entailment using probabilistic inference," Jan. 2015, pp. 1620–1625. DOI: 10.18653/v1/D15-1185.

[5] N. Sharma, R. Sharma, and K. K. Biswas, "Recognizing textual entailment using dependency analysis and machine learning," in *North American Chapter of the Association for Computational Linguistics*, 2015.

[6] Q. Liu, H. Jiang, A. Evdokimov, *et al.*, *Probabilistic reasoning via deep learning: Neural association models*, 2016. arXiv: 1603.07704 [cs.AI].

[7] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom, *Reasoning about entailment with neural attention*, 2016. arXiv: 1509.06664 [cs.CL].

[8] O. Cocarascu and F. Toni, "Identifying attack and support argumentative relations using deep learning," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1374–1379. DOI: 10.18653/v1/D17-1144. [Online]. Available: https://aclanthology.org/D17-1144.

[9] M.-Y. Kim and R. Goebel, "Two-step cascaded textual entailment for legal bar exam question answering," in *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ser. ICAIL '17, London, United Kingdom: Association for Computing Machinery, 2017, pp. 283–290, ISBN: 9781450348911. DOI: 10.1145/3086512.3086550. [Online]. Available: https://doi.org/10.1145/3086512.3086550.

[10] T. Khot, A. Sabharwal, and P. Clark, "Scitail: A textual entailment dataset from science question answering," in *AAAI Conference on Artificial Intelligence*, 2018.

[11] S. Mudgal, H. Li, T. Rekatsinas, *et al.*, "Deep learning for entity matching: A design space exploration," May 2018, pp. 19–34, ISBN: 978-1-4503-4703-7. DOI: 10.1145/3183713.3196926.

[12] A. Naserasadi, H. Khosravi, and F. Sadeghi, "Extractive multi-document summarization based on textual entailment and sentence compression via knapsack problem," *Natural Language Engineering*, vol. 25, pp. 121–146, 2018.

[13] D. Chen, Y. Li, M. Yang, H. Zheng, and Y. Shen, "Knowledge-aware textual entailment with graph attention network," *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[14] A. Hanselowski, H. Zhang, Z. Li, *et al.*, *Ukp-athene: Multi-sentence textual entailment for claim verification*, 2019. arXiv: 1809.01479 [cs.IR].

[15] J. Rabelo, M.-Y. Kim, and R. Goebel, "Combining similarity and transformer methods for case law entailment," *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, 2019.

[16] N. Tawfik and M. Spruit, "Towards recognition of textual entailment in the biomedical domain," in Jun. 2019, pp. 368–375, ISBN: 978-3-030-23280-1. DOI: 10.1007/978-3-030-23281-8_32.

[17] A. Poliak, *A survey on recognizing textual entailment as an nlp evaluation*, 2020. arXiv: 2010.03061 [cs.CL].

[18] T. Saikh, A. De, A. Ekbal, and P. Bhattacharyya, *A deep learning approach for automatic detection of fake news*, 2020. arXiv: 2005.04938 [cs.CL].

[19] A. Bhattacharjee, T. Hasan, W. U. Ahmad, *et al.*, *Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla*, 2022. arXiv: 2101.00204 [cs.CL].

[20] M. Shajalal, M. Atabuzzaman, M. B. Baby, M. R. Karim, and A. Boden, *Textual entailment recognition with semantic features from empirical text representation*, 2022. arXiv: 2210.09723 [cs.CL].