Master of Science in Computer Science and Engineering

Developing an Automated Framework for Detecting Check-worthiness of Bangla Claims

by

Md. Rashadur Rahman

ID: 18MCSE008P

This thesis is submitted in partial fulfillment of the requirement for the degree of
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chattogram-4349, Bangladesh.
January, 2025

CERTIFICATION

The thesis titled **"Developing an Automated Framework for Detecting Check-worthiness of Bangla Claims"** submitted by **Md. Rashadur Rahman**, Roll No. **18MCSE008P**, Session **2018-2019** has been accepted as satisfactory in partial fulfillment of the requirement for the degree of **Master of Science in Computer Science and Engineering** on **23/01/2025**.

BOARD OF EXAMINERS

1. _____

Dr. Mohammad Shamsul Arefin — Chairman (Supervisor)
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

2. _____

Dr. Abu Hasnat Mohammad Ashfak Habib — Member (Ex-officio)
Professor and Head
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

3. _____

Dr. Muhammad Ibrahim Khan — Member
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

4. _____

Dr. Mohammed Moshiul Hoque — Member
Professor
Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

5. _____

Dr. K. M. Azharul Hasan — Member (External)
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology (KUET))

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

Md. Rashadur Rahman
ID: 18MCSE008P

Dedicated to,

My beloved parents,

Md. Saidur Rahman *and* ***Mst. Rabeya Akter***

My lovely and supporting wife,

Moni Aktar

and My dearest son,

Zubair Rahman Sadiq

for their love, endless support, encouragement and sacrifices throughout this journey.

Acknowledgment

I am overwhelmed to get this opportunity to express my gratitude to those whose constant support and encouragement led me to the completion of this thesis. The satisfaction that accompanies the successful completion of this thesis would be incomplete without the mention of the people whose ceaseless cooperation made it possible. First and foremost, I feel immense proud in expressing my heartfelt respect and deepest sense of gratitude to my supervisor, Dr. Mohammad Shamsul Arefin for his continuous guidelines, constructive criticism, inspiration, and overall supervision of this research work. I am ever grateful to the honorable board members for their valuable suggestions and helpful feedback. I am also incredibly thankful to all the faculty members and the staff of the department of CSE, CUET for their support. Many of my colleagues and friends deserve special thanks for their assistance and motivation.

Finally and always, I am grateful to the Almighty Allah for giving me the strength and patience to complete this thesis work.

সারসংক্ষেপ

প্রযুক্তির বিবর্তন দ্রুত গতিতে মিডিয়া ট্রেন্ডগুলিকে রূপান্তরিত করছে, বিশেষ করে প্রধান স্রোতের মিডিয়া অনলাইন প্ল্যাটফর্মে স্থানান্তরিত হওয়ার সাথে সাথে। ডিজিটাল প্ল্যাটফর্ম এবং সোশ্যাল মিডিয়া মাধ্যমে ভুল তথ্যের দ্রুত ছড়িয়ে পড়া আজকের তথ্য সমৃদ্ধ সমাজে একটি গুরুতর চ্যালেঞ্জ হয়ে দাঁড়িয়েছে। ভুল তথ্যের বিরুদ্ধে লড়াইয়ে ফ্যাক্ট-চেকিংয়ের গুরুত্বপূর্ণ ভূমিকা থাকা সত্ত্বেও, ঐতিহ্যগত ম্যানুয়াল পদ্ধতি সময় সাপেক্ষ এবং বিপুল পরিমাণে কনটেন্ট পর্যালোচনা করতে অপর্যাপ্ত। স্বয়ংক্রিয় দাবি সনাক্তকরণ সিস্টেমগুলি টেক্সটের মধ্যে সম্ভাব্য মিথ্যা প্রমাণযোগ্য দাবিগুলি চিহ্নিত করে ফ্যাক্ট-চেকিংয়ে একটি গুরুত্বপূর্ণ প্রথম ধাপ হিসাবে কাজ করে, দ্রুত প্রতিক্রিয়া সময় এবং মানব ফ্যাক্ট-চেকার্সের কার্যভার হ্রাস করে। ইংরেজি ভাষার মতো সমৃদ্ধ ভাষার জন্য দাবি সনাক্তকরণ সম্পর্কিত গবেষণা পরিচালিত হলেও, বাংলা ভাষার মতো সম্পদ সীমিত ভাষার জন্য এই ডোমেন অনাবিষ্কৃত রয়েছে। এই গবেষণায়, আমরা বাংলার জন্য প্রথমবারের মতো একটি বহু-শ্রেণীর চেক-যোগ্য দাবি সনাক্তকরণ ডেটাসেট তৈরি করেছি যার নাম CheckBanC, যা তিনটি শ্রেনীতে এনোট করা ১০,০২৩ টি বাক্য নিয়ে গঠিত। বাংলা দাবি সনাক্তকরণের অনন্য চ্যালেঞ্জগুলি মোকাবেলা করার জন্য, আমরা একটি নতুন হাইব্রিড ফিচার ফিউশন মডেল প্রস্তাব করছি যা DistilBERT থেকে প্রাসঙ্গিক এম্বেডিংগুলিকে পরিসংখ্যানগত বাইগ্রাম ফিচারগুলোর সাথে একত্রিত করে। এই পদ্ধতিটি স্থানীয় সিনট্যাকটিক প্যাটার্ন এবং গ্লোবাল সিম্যান্টিক সম্পর্ক উভয়কেই বুঝতে পারে, মডেলের চেক-যোগ্য দাবিগুলি চিহ্নিত করার ক্ষমতা উল্লেখযোগ্যভাবে বাড়িয়ে তোলে। পরীক্ষামূলক ফলাফলগুলি আমাদের পদ্ধতির শ্রেষ্ঠত্ব প্রমাণ করে, CheckBanC ডেটাসেটে একটি নতুন স্টেট-অফ-দ্য-আর্ট পারফরম্যান্স অর্জন করে। এটি বিভিন্ন মেশিন লার্নিং, ডিপ লার্নিং এবং ট্রান্সফরমার-ভিত্তিক বেসলাইনগুলিকে ছাড়িয়ে যায়, ০.৮৪ ওয়েটেড F1-স্কোর অর্জন করে, দাবি সনাক্তকরণে অন্যান্য বিদ্যমান স্টেট-অফ-দ্য-আর্ট পদ্ধতিগুলিকে ছাড়িয়ে যায়। আমাদের ফলাফলগুলি বাংলা ভাষার জন্য স্বয়ংক্রিয় দাবি সনাক্তকরণে ভবিষ্যত গবেষণার জন্য একটি দৃঢ় ভিত্তি প্রতিষ্ঠা করে, ফ্যাক্ট-চেকিং প্রচেষ্টাকে সহজতর করে এবং সম্পদ সীমিত ভাষায় ভুল তথ্যকে দমন করে।

Keywords: ফ্যাক্ট চেকিং, দাবি সনাক্তকরণ, Distil-BERT, বাংলা দাবি ডাটাসেট, কম্পিউটেশনাল সাংবাদিকতা, টেক্সট ক্লাসিফিকেশন

Abstract

The evolution of technology is swiftly reshaping media trends, particularly with the mainstream media's migration to online platforms. The rapid proliferation of misinformation through digital platforms and social media has become a critical challenge in today's information-rich society. Despite the crucial role of fact-checking in combating misinformation, traditional manual approaches are time-intensive and insufficient to address the sheer volume of content. Automated claim detection systems serve as a critical first step in fact-checking by identifying potentially falsifiable claims within the text, enabling quicker response times and reducing the workload for human fact-checkers. While research on claim detection has been conducted for resource-rich languages like English, this domain remains underexplored for resource-constrained languages such as Bangla. In this study, we introduce the first-ever multiclass check-worthy claim detection dataset for Bangla named *CheckBanC*, comprising 10,023 sentences annotated into three categories. To address the unique challenges of Bangla claim detection, we propose a novel hybrid feature fusion model that combines contextual embeddings from DistilBERT with statistical bigram features. This approach captures both local syntactic patterns and global semantic relationships, significantly enhancing the model's ability to identify check-worthy claims. Experimental results demonstrate the superiority of our method, achieving a new state-of-the-art performance on the CheckBanC dataset. It outperforms various machine learning, deep learning, and transformer-based baselines, achieving a weighted F1-score of 0.84, surpassing other existing state-of-the-art approaches in claim detection. Our findings establish a strong foundation for future research in automated claim detection for Bangla, facilitating fact-checking efforts and curbing misinformation in resource-constrained languages.

Keywords: *Fact-checking; Claim detection; Distil-BERT, Bangla claim dataset; Computational journalism; Text classification*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Misinformation is one of the major threats to our modern society. Due to the rapid growth of information and social media, a vast amount of information is generated daily. Now it is easier than ever to propagate any misleading information to a large audience through modern technology and contemporary media. Around the world, misinformation has become more prevalent in recent times [1, 2]. Because of casual mistakes and even deliberate manipulation of information, there are frequently inaccurate, exaggerated, and misleading claims on critical issues [3]. The abundance of information online necessitates a new layer of verification to ensure the accuracy and reliability of what people encounter. Unfortunately, manually verifying all online content is incredibly time-consuming and can only cover a small fraction of the available information. Moreover, misinformation disseminates approximately eight times faster than valid information [4]. Therefore, automated or semi-automated fact-checking systems have become an imperative.

1.1 Fact Checking

Fact-checking is considered one of the major approaches for combating misinformation [5]. Fact-checking is the process of producing an informed assessment of the veracity of a claim [6, 7, 8]. Fact-checking aims to determine whether claims expressed in written or spoken language are true [6]. This is an important work in journalism and is often performed manually by dedicated organizations such as PolitiFact [1], EU FactCheck [2], BoomBD[3]. Fact-checking typically involves a series of steps: (i) Identifying claims to be checked (ii) Prioritizing the most

[1] https://www.politifact.com
[2] https://eufactcheck.eu
[3] https://www.boombd.com

important claims to address (iii) Gathering evidence related to the identified claims (iv) Reaching a final verdict by evaluating the claims against the gathered evidence. The process starts by identifying claims that require verification. It then determines whether these claims are supported or refuted by reliable evidence. Journalists and fact-checkers play a crucial role in identifying and rectifying misinformation and disseminating their findings to the public. However, the velocity with which information propagates through digital channels presents a significant challenge. Given the finite resources available for both human and automated fact-checking processes, it is impractical to verify all online content. Consequently, fact-checkers are compelled to prioritize their investigative efforts. Furthermore, it is essential to acknowledge the potential influence of cognitive biases on manual fact-checking procedures [9].

Due to the rapid dissemination of information on the internet, there is less or no time for verifying claims, leading to the rapid proliferation of misinformation before undergoing fact-checking [10]. An automatic claim detection system could speed up the fact-checking process which leads to a quicker response time when addressing claims. Additionally, it can preserve the valuable time of human fact-checkers, allowing them to focus on more intricate checks that require careful human judgment [11].

Claim detection constitutes the primary and essential step in automated fact-checking, as all subsequent stages rely on its results. The fact-checking pipeline has several steps [6, 12, 13, 14] which is shown in the Fig. 1.2.

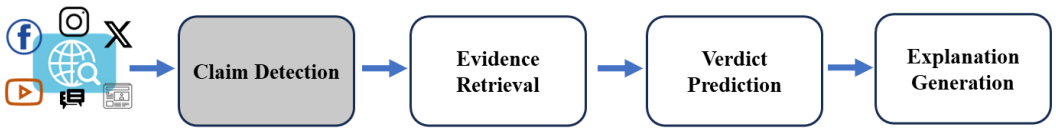


Figure 1.1: Fact-checking pipeline.

The fact-checking pipeline is a multi-step process that automates the verification of claims. It begins by identifying potential claims within text or speech. Then, it retrieves relevant evidence from various sources to support or contradict the claim. Next, the pipeline predicts the claim's veracity using machine learning or rule-based systems. Finally, it generates an explanation for the predicted verdict, providing insights into the reasoning behind the assessment.

1.2 Claim Detection

Claim detection is the first and one of the major steps of automated fact-checking, and all the other steps depend on the output of this stage. It attempts to relieve fact-checkers of the load of finding claims and to assist them by reducing the volume of internet information they must deal with[12]. Claim detection refers to the process of identifying claims within a given text or document. A claim is a statement or assertion made by a person or source that expresses a viewpoint or stance on a specific topic. The objective is to single out sentences in a long text that can be subjected to fact-checking [11]. Claim detection entails keeping an eye out for potentially false content on the internet or news reports so that it may be fact-checked [13]. A possibly falsifiable claim must be identified to stop the flow of misinformation.

A number of approaches have been made for detecting the check-worthiness of claims in some resource-rich languages like English. However, research in resource-constrained languages, particularly Bangla, remains significantly under-explored. To mitigate this gap, we propose the first approach for detecting check-worthiness of Bangla claims in multiclass categories. We also developed and published the first-ever multiclass claim detection dataset named *CheckBanC*. We proposed a hybrid feature fusion model incorporating both contextual and statistical features for effective check-worthiness on Bangla textual claims. Unlike traditional methods that rely solely on hand-crafted features, the use of DistilBERT allows the model to automatically learn rich, domain-specific representations from large-scale pre-trained language models. Our proposed model effectively integrates bigram features with contextual embeddings extracted from DistilBERT. This hybrid approach allows the model to capture both local syntactic patterns and global semantic relationships within the text. By leveraging this combination of sources of information, the model learns richer representations, surpassing previous methods in checkworthiness detection and establishing a new state-of-the-art on CheckBanC dataset for the Bangla language.

1.3 Motivation

In today's interconnected world, the rapid spread of misinformation poses a significant threat to society. Social media and digital platforms have amplified the dissemination of misleading information, which often spreads faster than verified facts. This has grave implications for public opinion, decision-making, and societal harmony. While numerous automated claim detection systems exist for

resource-rich languages like English, Bangla—a widely spoken language—remains vastly underrepresented in this critical research area.

A nationwide survey revealed that news literacy in Bangladesh is alarmingly low, with 76% of the population exhibiting poor news literacy and only 24% demonstrating high news literacy [15]. This lack of awareness and ability to critically evaluate news content further exacerbates the problem of misinformation. Without effective tools to assist in identifying misleading claims, the Bangla-speaking population remains vulnerable to the adverse impacts of false information.

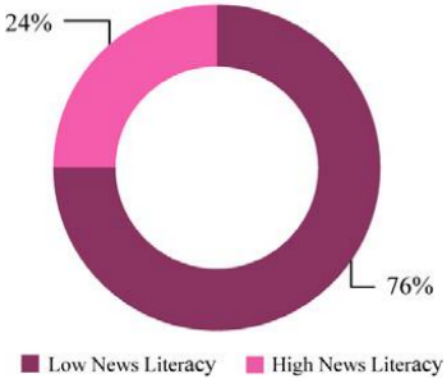


Figure 1.2: News literacy rate in Bangladesh [15].

The lack of computational tools to identify check-worthy claims in Bangla texts has created a gap in combating misinformation effectively for Bangla-speaking populations. Fact-checking organizations and journalists working in Bangla face immense challenges in manually sifting through vast volumes of information. Without automated tools to identify potential claims for verification, the process remains slow, inefficient, and error-prone. This research is motivated by the urgent need to address these gaps and empower Bangla-speaking communities with advanced tools for misinformation detection. Our key motivations for this work are:

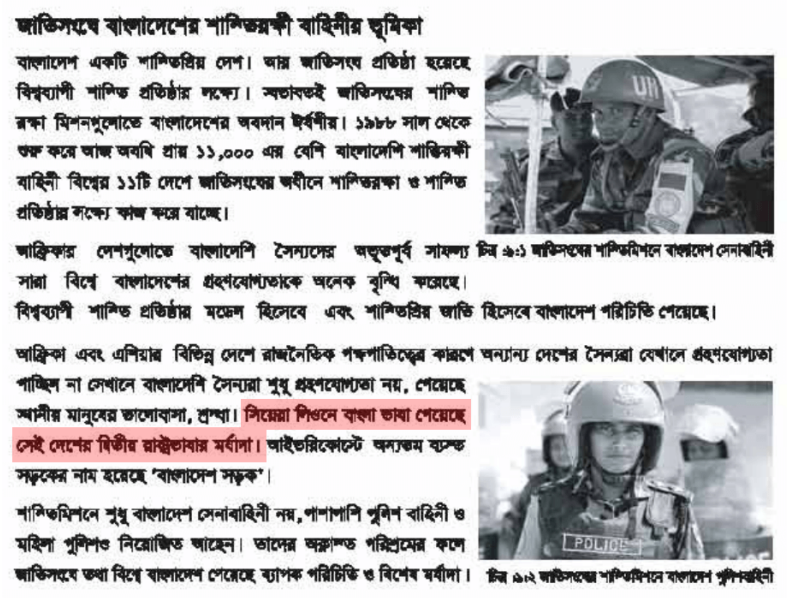
- To address the lack of a check-worthy claim detection system in Bangla.
- Develop computational resources for Bangla textual claim detection and fact-checking to assist fact-checkers and journalists by reducing their workload in identifying check-worthy claims.
- Contribute to the control of misinformation propagation by promoting access to verified information and improving public awareness.
- Set forth a benchmark methodology for further Bangla claim detection approaches.

1.4 Importance

In an era dominated by social media and digital platforms, the rapid dissemination of false information can mislead people, influence opinions, and even incite unrest. Misinformation is a growing concern in today's digitally connected world, particularly in languages like Bangla, where automated tools for fact-checking are virtually nonexistent. The rapid spread of false information on social media, official documents, and educational resources has significant implications for public opinion, societal harmony, and informed decision-making. For instance, during a recent critical political event in Bangladesh, a widely circulated social media post (Figure 1.3) falsely claimed that there was a huge possibility of international sanctions on the military, which led to widespread public unrest and protests. Such misinformation not only destabilizes social harmony but also exacerbates political tensions. An automated system for identifying potentially false or check-worthy claims in Bangla could serve as the first line of defense in addressing such challenges, enabling quicker fact-checking and preventing mass panic.

Figure 1.3: Deliberate misinformation on a critical issue in a viral social media post.

The propagation of false information is not limited to social media; it can also infiltrate trusted resources such as educational materials. A notable example is the incorrect statement published in the National Curriculum "Bangladesh and World Studies" book for classes 9-10 (Figure 1.4), claiming that "Bangla is the second state language of Sierra Leone." This misinformation not only misguides students but also undermines the credibility of educational institutions. Detecting and rectifying such false claims is crucial for ensuring the integrity of academic content and promoting accurate knowledge dissemination.



জাতিসংঘে বাংলাদেশের শান্তিরক্ষী বাহিনীর ভূমিকা

বাংলাদেশ একটি শান্তিপ্রিয় দেশ। আর জাতিসংঘ প্রতিষ্ঠা হয়েছে বিশ্বব্যাপী শান্তি প্রতিষ্ঠার লক্ষ্যে। স্বভাবতই জাতিসংঘের শান্তি রক্ষা মিশনগুলোতে বাংলাদেশের অবদান ঈর্ষণীয়। ১৯৮৮ সাল থেকে শুরু করে আজ অবধি প্রায় ১১,০০০ এর বেশি বাংলাদেশি শান্তিরক্ষী বাহিনী বিশ্বের ১১টি দেশে জাতিসংঘের অধীনে শান্তিরক্ষা ও শান্তি প্রতিষ্ঠার লক্ষ্যে কাজ করে যাচ্ছে।

চিত্র ৯:১ জাতিসংঘের শান্তিমিশনে বাংলাদেশ সেনাবাহিনী

আফ্রিকার দেশগুলোতে বাংলাদেশি সৈন্যদের অভূতপূর্ব সাফল্য সারা বিশ্বে বাংলাদেশের গ্রহণযোগ্যতাকে অনেক বৃদ্ধি করেছে। বিশ্বব্যাপী শান্তি প্রতিষ্ঠায় মডেল হিসেবে এবং শান্তিপ্রিয় জাতি হিসেবে বাংলাদেশ পরিচিতি পেয়েছে।

আফ্রিকা এবং এশিয়ার বিভিন্ন দেশে রাজনৈতিক পক্ষপাতিত্বের কারণে অন্যান্য দেশের সৈন্যরা যেখানে গ্রহণযোগ্যতা পাচ্ছিল না সেখানে বাংলাদেশি সৈন্যরা শুধু গ্রহণযোগ্যতা নয়, পেয়েছে স্থানীয় মানুষের ভালোবাসা, শ্রদ্ধা। সিয়েরা লিওনে বাংলা ভাষা পেয়েছে সেই দেশের দ্বিতীয় রাষ্ট্রভাষার মর্যাদা। আইভরিকোস্টে অন্যতম ব্যস্ত সড়কের নাম হয়েছে 'বাংলাদেশ সড়ক'।

শান্তিমিশনে শুধু বাংলাদেশ সেনাবাহিনী নয়, পাশাপাশি পুলিশ বাহিনী ও মহিলা পুলিশও নিয়োজিত আছেন। তাদের অক্লান্ত পরিশ্রমের ফলে জাতিসংঘে তথা বিশ্বে বাংলাদেশ পেয়েছে ব্যাপক পরিচিতি ও বিশেষ মর্যাদা।

চিত্র ৯:২ জাতিসংঘের শান্তিমিশনে বাংলাদেশ পুলিশবাহিনী

Figure 1.4: False information published in a national curriculum textbook.

Our system can play a crucial role by automatically analyzing such claims and flagging them as check-worthy based on their potential to create controversy or panic. This early detection allows fact-checkers to verify the claim and disseminate accurate information promptly, mitigating the spread of misinformation and restoring public confidence. By prioritizing politically sensitive and high-impact claims, the system ensures that the most critical misinformation is addressed swiftly. Some important implications of our work are listed below:

- This system can play a crucial role in preventing the spread of misinformation by detecting claims that warrant verification early, the system helps mitigate the rapid dissemination of false information in both digital and traditional media.
- By incorporating the system into the content creation process, the reliability of educational and informational materials in journalism and academia can be significantly improved.
- The automated approach reduces the workload of fact-checkers by prioritizing claims that require immediate attention, allowing for efficient resource allocation.
- Finally this work addresses a critical gap by developing the first-ever approach to detecting check-worthy Bangla claims, paving the way for further advancements in Bangla automated claim detection and fact-checking research.

1.5 Challenges

The necessity for an automated claim detection technique in Bangla is undeniable, especially given the lack of any existing approaches. While several advanced techniques for claim detection exist in resource-rich languages like English, Bangla remains an underexplored domain. This gap is particularly concerning given the increasing spread of misinformation in digital spaces. Our study represents the first-ever attempt to address this issue by introducing a Bangla claim detection system. However, implementing this pioneering system posed several challenges, as described below:

- **Absense of Dataset:** The absence of a Bangla-specific claim detection dataset was one of the most fundamental obstacles. A properly annotated dataset is very crucial to capture language-specific features, such as morphology and sentence structure, which are crucial for accurately identifying claims [8].
- **Designing Suitable Labels and Annotation Guidelines:** Determining the appropriate labels and designing comprehensive annotation guidelines was another significant challenge. The labels had to capture varying degrees of check-worthiness while remaining clear and intuitive for annotators. Additionally, creating guidelines that reduced subjectivity and ensured consistent annotations was a complex task, given the nuanced nature of claim detection.
- **Overlapping Characteristics Across Categories:** Many claims exhibited characteristics that blurred the boundaries between these categories, making it difficult to classify them accurately [3]. This required advanced feature extraction methods to distinguish between such closely related categories effectively.
- **Lack of Benchmark for Performance Comparison:** The absence of any benchmark for Bangla claim detection, which made it difficult to evaluate our model's performance against pre-existing systems.

By addressing these challenges, our study introduces the first automated claim detection approach for Bangla, providing a much-needed framework for combating misinformation in this language. These efforts lay the groundwork for future advancements, enabling more sophisticated and scalable solutions for automated fact-checking in Bangla.

1.6 Contributions

The rapid proliferation of misinformation, particularly in resource-constrained languages like Bangla, poses a significant challenge to fact-checking efforts. Despite considerable advancements in claim detection for resource-rich languages, research in Bangla remains underexplored. This work bridges this gap by introducing a comprehensive solution that combines dataset development and methodological innovation tailored to the unique challenges of Bangla claim detection. Our contributions are multifaceted, addressing both foundational and applied aspects of the problem. Below, we outline the key elements of our work:

- We have developed the first-ever Bangla multiclass claim check-worthiness detection dataset named *CheckBanC*, which Contains 10,023 sentences collected and annotated from different fact-checking sites, interviews, and speeches. Labeled as one of the three categories Check-worthy claim (3476), Weak Factual claim (2957), and Non-Claim (3590).
- Proposed a novel deep neural network leveraging a hybrid feature fusion technique that combines bi-gram and DistilBERT-based features. Our approach integrates both contextual and frequency-based features to enhance the detection of check-worthy claims.
- Presented a comparative analysis of the performance of our proposed model with various machine learning, deep learning, and transformer baseline models, as well as existing state-of-the-art techniques, thereby establishing a benchmark for future research.

To the best of our knowledge, an approach has yet to be made to automatically detect check-worthy claims for the Bengali text. We are proposing the first-ever computational method for detecting the check-worthiness of Bengali claims to facilitate automated fact-checking for the Bangla language. These contributions collectively advance the field of claim detection and fact-checking, particularly in resource-constrained settings, providing a scalable solution to mitigate misinformation in digital and social media platforms.

1.7 Organization of the Thesis

The remaining of thesis is organized as follows: Chapter 2 presents a detailed review of prior research on claim detection techniques. It discusses prior work in

resource-rich languages like English and highlights the lack of research in resource-constrained languages, particularly Bangla, setting the stage for the contributions of this thesis. Chapter 3 focuses on the creation and annotation of the CheckBanC dataset, the first-ever Bangla multiclass claim detection dataset. It discusses the dataset collection, preprocessing, annotation schema, and descriptive statistics, ensuring transparency in the dataset's construction. Chapter 4 describes the baseline models, including traditional machine learning, deep learning, and transformer-based approaches, and introduces the proposed hybrid feature fusion model. Chapter 5 presents the experimental setup, evaluation metrics, and detailed performance analysis of all models, including comparisons with baseline methods. It includes discussions on the misclassification analysis, computational complexity, and the strengths and limitations of the proposed approach. Chapter 6 summarizes the key contributions of the thesis, discusses its limitations, and provides directions for future research in claim detection

Chapter 2

Literature Review

Claim detection is the initial and most critical step of the automated fact-checking process. A vast portion of the literature is focused on determining the correctness of claims that were already extracted [16, 17]. There are also some approaches for spotting previously verified claims [18]. Research on automated claim detection is still in the emerging phase.

2.1 Claim Detection Approaches in English

ClaimBuster [3, 19, 20] is one of the pioneering approaches for check-worthy claim detection. They developed a check-worthy factual claim dataset from U.S. general election presidential debates. They experimented with several multiclass machine learning classifiers with different combinations of term frequency-inverse document frequency (TF-IDF), part-of-speech (POS) tags, and entity type (ET) feature sets. Their proposed Support Vector machine(SVM) model utilizing TF-IDF and POS features achieves the highest performance of a weighted average f_1 score (WF1) score of 0.818.

A model named "CNC" (Claim/No Claim) was proposed by Konstantinovskiy et al. [11]. They developed a dataset containing 5,571 sentences from subtitles of UK political television shows and FullFact database of claims. They experimented with four machine learning classifiers for binary and multiclass classification of claims. The model utilized TF-IDF, named entity recognition (NER), POS features along with InferSent [21], Word2Vec, and GloVe embeddings for feature set generation. Rather than check-worthiness, they annotated claims based on their checkability. Their proposed logistic regression-based model achieved an F1 score of 0.83 in binary classification and a macro average F1 score (MF1) of 0.48 in multiclass classification. The dataset they developed was class imbalanced, furthermore, the performance on multiclassification was very low. Meng

et al. [22] incorporated adversarial training on transformer models for detecting check-worthy factual claims. They achieved better F1 scores on the ClaimBuster and CLEF2019 datasets. A multi-classifier model named TATHYA was developed in the study [23]. Multiple SVMs were trained on different segments of the dataset then taking the output from the most confident one. They utilized TF-IDF weighted bag-of-unigrams, POS tuples, sentence topics, and entity counts as feature set. Bai et al. [24] introduced a query-based claim detection system. They developed a dataset named Tweet-MythQA, on which the transformed-based DeBERTa-large model achieved the highest score.

There are some recent approaches for detecting domain-specific check-worthy claims. Gollapalli et al. [25] attempted to detect check-worthy claims on medical condition claims named 'CURE Claims'. Developed a dataset (CW-CURE) containing tweets on medical conditions and utilized a crowd annotation platform to annotate the dataset. The study proposed a fine-tuned text-to-text transformer T5 [26] and pre-trained with CCT-22 [27] dataset. Wei et al. [28] utilized supervised constructive learning for detecting scientific claims. Their proposed model named ClaimDistiller is based on a word and character embedding Bidirectional Long Short-Term Memory (WC-BiLSTM) model trained in two stages. They augmented the data using the Wordnet synonym replacement technique before training. Alhindi et al. [29] developed a dataset from 95 news articles on climate change and employed a multi-layered annotation methodology. This approach incorporated the argumentative discourse surrounding the target sentence, providing a more comprehensive context. They fine-tuned the BERT model on their developed dataset. Another previous attempt named 'SciClops' at detecting scientific claims was proposed in this study [30].

Woloszyn et al. [31] proposed a 'green claim' detection technique which are claims related to environmental issues. They experimented with three models RoBERTa [32], BERTweet [33], and Flair [34]. The RoBERTa model achieved the highest performance in detecting 'green claim'. A recent approach incorporating stylistic vectors with BERT based model for detecting environmental claims was proposed by Saha et al. [35].

CLEF Check That! Lab is renowned for its shared task competition arrangement for detecting check-worthy claims since 2018 [36]. The CLEF-2018 CheckThat Lab [37] established a crucial foundation for the automated detection of verifiable claims in political discourse. The dataset was constructed from U.S. political debates and speeches where statements that had been fact-checked by FactCheck.org were labeled as check-worthy. For the Arabic check-worthy detection task, the organizers relied solely on translations of the English data.

In 2019, the organizers expanded the dataset introduced in the previous year [38]. Subsequently, in 2020, they augmented the dataset further by incorporating tweets related to COVID-19, alongside those from the political domain [39]. The dataset of the 2022 competition [27] consists of Twitter data in Arabic, Bulgarian, Dutch, and English language. Eyuboglu et al. [40] proposed a four-layer feedforward network with Manifold Mixup regularization with BERT-based embeddings. Their proposed model performed best in the competition in detecting verifiable claims in Arabic. For data augmentation, they applied both translation and back-translation techniques. In the competition of CheckThat! 2024 [41] task-1, team OpenFact [42] proposed DeBERTa [43] and mDeBERTa [44] transformer models fine-tuned with CT-CWT-24 dataset. They utilized different under-sampling methods to tackle the imbalance in the dataset. The binary classification dataset contains four different languages Arabic, English, Dutch, and Spanish. The Spanish set was only for training purposes. The team achieved outstanding performance in the Arabic and English leaderboard.

2.2 Claim Detection Approaches in other Languages

While all of the above research focused on claims of English, there have been a few efforts in other languages as well. The very first approach for check-worthy claim detection in the Turkish language was proposed by Kartal et al [45]. They developed a containing 2,287 tweets on major events and labeled them. The study utilized two BERT-based models ('m-BERT', 'BERTTurk') and two machine learning models (Logistic regression and SVM). The m-BERT model achieved the highest average precision (AP) of 0.383 in binary classification on their dataset. FactRank [46] is the first approach for automated claim detection in the Dutch language which classifers sentences into three classes. They constructed a dataset of 7,7037 sentences and a codebook to annot claims in the Dutch language. The study proposed a convolutional neural network (CNN) architecture with 320-dimensional COW-big word vectors as input. The model shows the CNN model achieves an accuracy of 74.7% on the test set and the BERTje model achieves an accuracy of 74.8% on the validation set. The sample size in the test and validation set is very insufficient for reliable evaluation.

Ranking claims based on their check-worthiness is another way of detecting check-worthiness. A system named ClaimRank for ranking claims was proposed by Jaradat et al. [47]. For developing the dataset, they used seven English

political debates and Arabic translation of two of the English debates. They used the class probability score of the positive class predicted by a feed-forward neural network (FNN) as the ranking score of a claim. They experimented with different cross-language embeddings such as VecMap, MUSE, and Attract-Repel as the input features for the neural network. The model was incorporated from their previous context-aware approach [48] for claim detection.

A summary of the related work in check-worthy claim detection highlighting their technique, dataset, performance, and shortcomings are presented in Table 2.1.

Table 2.1: Summary of the literature on claim detection.

Research	Proposed technique	Dataset	Language	Performance	Limitation
ClaimBuster Hassan et al. [19]	SVM with TF-IDF and POS tag features	8,231 sentences (Developed from U.S. presidential election debates)	English	WF1: 0.818	Class imbalance, low performance on the UFS class
CNC Konstantinovskiy et al. [11]	Binary: logistic regression; Multiclass: multinomial logistic regression; with POS tag and NER features	5,571 sentences (Developed from subtitles of UK political TV shows + FullFact database)	English	F1: 0.83 (Binary) MF1: 0.48 (Multi)	Class imbalance, poor performance on multi-classification
FactRank Berendt et al.[46]	CNN with COW-big word vectors	7,037 sentences (Constructed from Flemish, Dutch political and news sites	Dutch	Accuracy: 74.7%	Inconclusive evaluation due to insufficient test set Class imbalance
TrClaim Kartal et al. [45]	m-BERT	2,287 tweets (collected tweets on major events)	Turkish	AP: 0.383	Low inter-annotator agreement No hyperparameter optimization
ClaimRank Jaradat et al.[47]	Neural network with two hidden layers	Political debates	English, Arabic	MAP: 0.34 (English) MAP:0.30 (Arabic)	Ranking performed without initial ranking score
CURE Claim Gollapalli et al. [25]	T5 transformer pre-trained with CCT-22 dataset	3,266 tweets (Tweets related to medical condition)	English	F1: 0.826	Low generalization
ClaimDistiller Wei et al. [28]	Supervised constructive learning based on WC-BiLSTM with WordNet augmentation	SciCE [49], SciARK [50]	English	F1: 0.874 (SciCE) F1: 0.894 (SciARK)	Over augmentation of data
OpenFact Sawiński et al.[42]	mDeBERTa and DeBERTa with undersampling	CT-CWT-24 [41]	English, Arabic, Dutch	F1: 0.796 (English) F1: 0.557 (Arabic) F1: 0.590 (Dutch)	No hyperparameter optimization

2.3 Research Gap

Despite the increasing prevalence of misinformation in Bangla-speaking communities, there has been no prior attempt to develop an automated system for

detecting check-worthy claims in the Bangla language. A critical prerequisite for developing robust claim detection systems is the availability of well-annotated datasets. However, until now, there has been no dedicated dataset for Bangla claim detection. This lack of resources has hindered the development and benchmarking of models tailored to the language, leaving a substantial gap in the foundational infrastructure required for research in this domain. While previous studies in other languages have primarily focused on binary classification (check-worthy or not), multiclass claim detection—categorizing claims into nuanced classes is rare. This gap prevents more comprehensive evaluations of claim characteristics and their potential impact. Existing claim detection models in other languages primarily focus on either statistical features (e.g., n-grams) or contextual embeddings (e.g., derived from transformer models). However, no model to date has explored a hybrid feature approach that combines these two complementary sources of information. Statistical features capture local syntactic patterns, while contextual embeddings provide global semantic understanding, both of which are crucial for efficient claim detection.

This thesis aims to address these gaps by introducing the first-ever Bangla claim detection dataset, proposing a hybrid model that leverages both statistical and contextual features, and establishing benchmark methodologies. The key research questions that this work aims to address are:

- **RQ-1:** How can we develop an annotated dataset of check-worthy Bangla claims?
- **RQ-2:** How can we effectively detect check-worthiness of Bangla claims and categorize them into appropriate categories?

Chapter 3

Dataset Development

A critical prerequisite for developing robust claim detection systems is the availability of well-annotated datasets. To the best of our knowledge, there is no publicly available corpus for check-worthiness detection of claims in Bangla. This Chapter provides the overall procedure of developing the first ever Bangla multi-class textual claim detection dataset named *"CheckBanC"*. Figure 3.1 shows the steps of developing our Bangla check-worthy claim detection dataset.

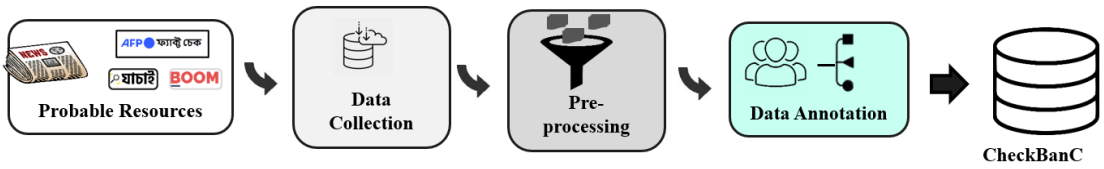


Figure 3.1: Bangla check-worthy claim detection dataset CheckBanC development procedure.

3.1 Primary Data Collection

Meticulous selection of initial sources is essential for the creation of a quality textual dataset. Fact-checking websites are ideal sources for textual claim data but they are limited in number for Bangla language [5, 51]. Speeches are often used as a source for check-worthiness detection of claims in previous studies [6, 11, 52, 37]. Interviews published in the newspapers are another vital source of claims. The candidate sentences were collected from fact-checking sites, newspaper fact-checking, published interviews, and speeches. Data were collected within the duration from August 2022 to April 2024. We only considered the textual claims from the fact-checking and newspaper fact-checking sites. Additionally, the related descriptions or evidence used to verify the accuracy of the claims

were included as primary data. We meticulously collected the primary sentences containing factual claims from the selected sources. We manually accumulated 17,745 sentences as primary data from the selected sources.

3.2 Initial Data Processing

To ensure the consistency and quality of the dataset, the primary accumulated data undergoes several pre-processing steps.

- We filtered the sentences containing non-Bengali words and emojis.
- The irrelevant and duplicate sentences were also removed.
- We excluded claims that span multiple sentences or contain fewer than three words.

A total of 5,340 sentences were discarded in this stage and the remaining 12,405 sentences were selected for annotation.

3.3 Data Annotation

The quality of the dataset is directly contingent upon the precision of the annotation process, which presents the most significant challenge in the development of any textual dataset [53]. Accurate and consistent labeling ensures that the data can be effectively utilized for training and evaluating machine learning models. The criteria used to assess a claim's check-worthiness will vary depending on its topic or area and the user groups (i.e., audience) that are interested in the claim's validity [13, 54]. Owing to the subjective nature of this work, it is challenging to determine if a verifiable claim is check-worthy, and this depends on the topic covered in the claim as well as the user group that is interested in it [52]. To mitigate these challenges, we carefully selected annotators and developed a comprehensive annotation guideline with the assistance of domain experts.

3.3.1 Selection of Annotators

The selection of appropriate annotators is crucial for ensuring the accuracy and reliability of labeled text data, particularly in tasks like claim detection [47]. Annotators must have a deep understanding of the language, cultural nuances, and domain-specific knowledge to accurately interpret the context of claims and assign appropriate labels. Ten annotators were initially selected for the annotation

process. After undergoing training with annotation guidelines and a codebook containing representative examples of each category, a test was conducted to assess their annotation quality. Each annotator was given 100 unseen samples to label, and their results were compared to expert annotations. The five best-performing annotators were then chosen for the final annotation process.

The selected five annotators were all native Bengali speakers, comprising two females and three males aged between 22-30 years. Three of them are undergraduate students, while two are pursuing graduate studies. Their diverse backgrounds include computer science, business administration, and journalism. Our expert, an academic with extensive experience in natural language processing, oversaw the training and guidance of the annotators. This diversity of backgrounds and experiences ensured a comprehensive and nuanced understanding of the subject matter.

3.3.2 Annotation Guidelines

Each of the sentences was independently annotated by three annotators using our custom annotation web portal[1]. The annotation guideline, depicted in Figure 3.2, outlines the decision-making process. The interface of our annotation portal is

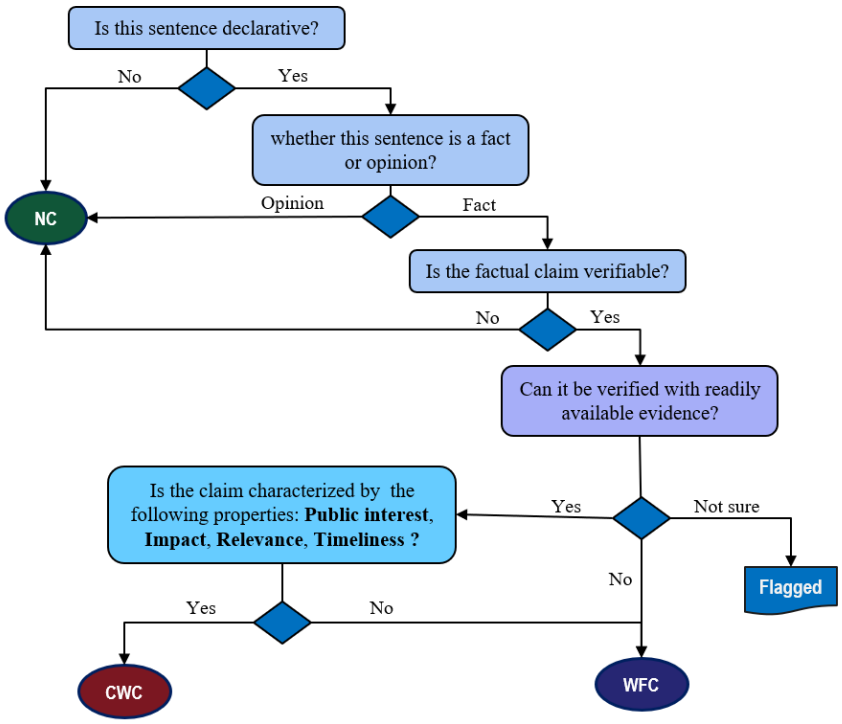


Figure 3.2: Abstract decision flowchart of our annotation guidelines.

[1] https://annot.vercel.app/

shown in Figure 3.3 where each sentence appears and the annotator can assign each of the three labels to that sentence. Additionally, the annotators had the option to skip the sentence for later or mark it as "Fagged" if he or she is not sure about the appropriate label.

After collecting the primary labels from three annotators, we followed Algorithm 1 to assign the final label for each sample sentence. For each sentence s_i, we first check the primary label of the three annotators for that sentence. If any sentence was labeled as *Flagged* by more than one annotator, the sentence was discarded. This step is crucial for maintaining the quality and consistency of the dataset, as it removes sentences that are ambiguous. The final label of any sentence was selected on majority voting. If there is a tie, the sentence is marked for expert review. Based on the expert's discussion and review, the marked sentences were either discarded or re-annotated.

3.3.3 Annotation Quality

To assess the inter-annotator reliability of our dataset, we employed the Fleiss' Kappa coefficient [55]. This statistical measure, commonly used in the field of natural language processing, quantifies the level of agreement among multiple annotators. The agreement is calculated by the Equation 3.1. Where P_o is the observed agreement among annotators P_e is the expected agreement by chance.

$$K = \frac{P_o - P_e}{1 - P_e} \tag{3.1}$$

A Kappa score of 0.78 was achieved, indicating a substantial consensus among our three annotators. This result is a testament to the effectiveness of our annotation guidelines, the training provided to the annotators, and the overall quality of the dataset. The annotation also achieved inter-coder reliability of 81% which also indicates significant agreement among the annotators. Table 3.1 presents the Jaccard similarity index between different classes in the dataset, measuring the overlap of the 300 most frequent words among them.

The values indicate how much lexical similarity exists between classes, with a higher score representing greater overlap. The self-similarity values are 1.00, as expected. The similarity between CWC (Check-worthy claim) and WFC (Weak factual claim) is 0.37, suggesting a moderate overlap in word usage, while CWC and NC (Non claim) have a lower similarity of 0.26, indicating more distinct vocabulary. The highest inter-class similarity is observed between WFC and NC at 0.47, suggesting that these two categories share more linguistic features compared to their relationship with CWC. This overlap in vocabulary highlights the

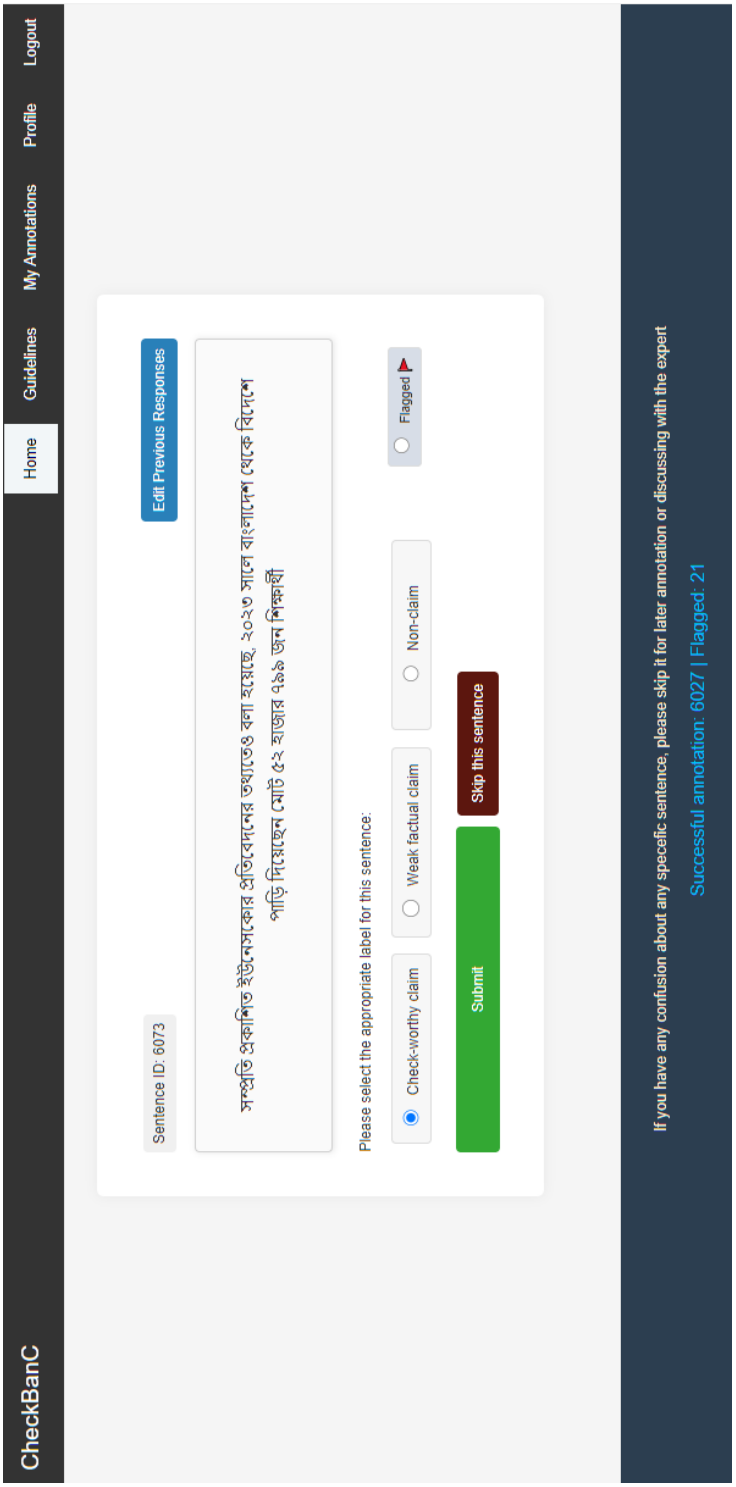


Figure 3.3: Data annotation interface.

Algorithm 1: Procedure for assigning final label

Input: Set of texts with initial labels from 3 annotators
Output: Final label for each text or flagged for expert review
$S \leftarrow \{s_1, s_2, ..., s_m\}$ {set of texts}
$FD \leftarrow []$ {final labelled texts}
$FL \leftarrow []$ {final class labels}
$E \leftarrow []$ {texts marked for expert review}
for $s_i \in S$ **do**
    $L \leftarrow [IL[i][1], IL[i][2], IL[i][3]]$ {labels from annotators}
    $flagged\_count \leftarrow$ count of $Flag$ in L
    if $flagged\_count \geq 2$ **then**
        continue {Discard the text}
    end if
    if $flagged\_count == 1$ **then**
        $L \leftarrow$ non-flagged labels in L {remove the flagged label, keep valid class labels}
    end if
    $label\_freq \leftarrow (label, frequency\_count)$ for each $label$ in L
    $max\_label, max\_frequency \leftarrow (label, frequency\_count)$ of maximum $frequency\_count$ in $abel\_freq$
    if $max\_frequency \geq 2$ **then**
        FD.append(s_i)
        FL.append(max_label)
    else
        E.append(s_i) {annotators disagree, mark for expert review}
    end if
end for
for $d_j \in E$ **do**
    discussion with the expert
    discard or re-annotate d_j
end for

Table 3.1: Jaccard similarity index between classes.

	CWC	**WFC**	**NC**
CWC	1.00	0.37	0.26
WFC		1.00	0.47

challenge of distinguishing between certain claim types, underscoring the need for effective feature selection and model robustness to ensure accurate classification.

3.4 *CheckBanC*: Bangla Claim Dataset

Our final annotated dataset named as *CheckBanC* contains a total of 10,023 sentences. The sentences are labeled as one of the following classes:

- **Check-worthy claim (CWC):** Statements containing factual claims with a higher degree of interest in knowing the veracity of the claim. These statements have a significant impact, relevance, timeliness, and public interest.
- **Weak Factual claim (WFC):** Texts containing factual claims but these are not worth checking the veracity or can not be verified the truthfulness of the claim with readily available evidence.
- **Non-Claim (NC):** There are no assertions or facts in these sentences. Non-declarative, subjective phrases (opinions, views, pronouncements), and unverifiable sentences come into this class.

Table 3.2 shows samples from our dataset along with English translation. The CheckBanC dataset includes samples gathered from a diverse set of sources, offering a comprehensive representation of claim types pertinent to Bangla fact-checking. Figure 3.4 presents the source distribution of the dataset, showing the number of samples and their respective percentages.

Table 3.3 provides a detailed breakdown of major data sources, including URLs and sample counts. A significant portion of the data was sourced from dedicated Bangla fact-checking sites, such as AFP Factcheck, BOOM Bangladesh, and Fact Watch, each of which contributed substantial sample numbers. In addition to fact-checking websites, the dataset incorporates samples from the fact-checking sections of popular Bangla news portals, including Ajker Patrika, News Bangla 24, and Somoy News. The dataset also includes interview content from established news outlets such as Prothom Alo, The Daily Star, and Jugantor. These interview sections offer a unique source of claims, typically in the form

Table 3.2: Some samples of our CheckBanC dataset from different classes.

Sentence	Label	Remarks
দেশের সবক'টি উপজেলাকে ইন্টারনেটের আওতায় আনা হয়েছে (All the upazilas of the country have been brought under the internet)	**CWC**	There is strong public interest in this statement
যে কোনও পরিস্থিতি থেকে সেরাটা বার করার চেষ্টা করেন তাঁরা (They try to make the best out of any situation)	**WFC**	A claim which can not be verified with readily available information
সদ্য সমাপ্ত অর্থ বছরে সরকার ৯৮ হাজার ৮২৬ কোটি টাকা ঋণ নেওয়ার পর বাংলাদেশ ব্যাংক সরকারের জন্য অর্থ ছাপানো বন্ধ করে দেয় (Bangladesh Bank stopped printing money for the government after the government took a loan of Tk 98,826 crore in the last financial year)	**CWC**	Verifiable statement that demonstrates relevance, impact, and timeliness
আমরা আসলে বর্তমান পরিস্থিতির জন্য মোটেই প্রস্তুত ছিলাম না (We were actually not prepared at all for the current situation)	**NC**	This statement is a personal anecdote
এ পর্যন্ত কুকিচিনের পুতে রাখা আইইডি বিস্ফোরণ ও হামলায় বান্দরবানে পাঁচ সেনা সদস্য নিহত হয়েছেন (So far, five army personnel have been killed in Bandarban in an IED blast and attack in Kukichin)	**CWC**	Statement containing significant impact
তবে দেশের সমস্যাগুলোর সমাধান হলে বেশির ভাগই বিদেশ থেকে ফিরে আসতে চান বলে জানিয়েছেন (However, if the problems of the country are solved, most of them said that they want to return from abroad)	**WFC**	Statement lacking relevance, impact and interest
নারীরা যেহেতু পিছিয়ে আছেন, তাই তাঁদের দিকে বিশেষভাবে নজর দিতে চাই (Since women are lagging behind, we want to pay special attention to them)	**NC**	This is an opinion

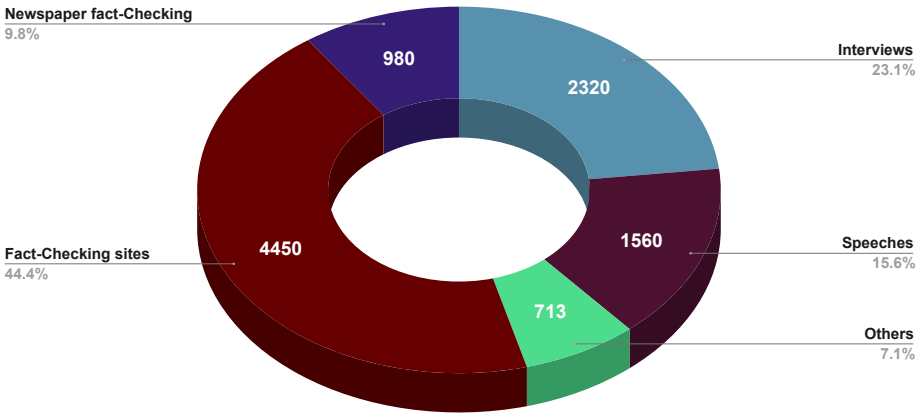


Figure 3.4: Source distribution of our developed CheckBanC claim dataset.

of quotes or assertions by public figures, which can contain both factual information and subjective opinions. For speech sources, we collected the speeches of important political persons published in different newspapers mentioned in Table 3.3.

Table 3.3: Breakdown of our data collection sources with respective sample count.

	Name	URL	No. of Samples
Fact-Checking sites	AFP Factcheck	factcheckbangla.afp.com	1088
	BOOM Bangladesh	www.boombd.com	1042
	Fact Watch	www.fact-watch.org/web	253
	newschecker	bangladesh.newschecker.co	996
	Reumor Scanner	rumorscanner.com	986
	Jachai	www.jachai.org	185
Newspaper fact-checking	Ajker Patrika	www.ajkerpatrika.com/fact-check	445
	News Bangla 24	www.newsbangla24.com/fact-check	267
	Daily Bangladesh	www.daily-bangladesh.com/fact-check	158
	Somoy News	www.somoynews.tv	110
Interviews	Prothom Alo	www.prothomalo.com/opinion/interview	983
	The Daily Star	bangla.thedailystar.net/opinion/views/interview	485
	Jugantor	www.jugantor.com/interview	395
	Jago News 24	www.jagonews24.com/interview	292
	Desh Rupantor	www.deshrupantor.com/interview	165

3.4.1 Dataset Statistics

The CheckBanC dataset is the first of its kind for Bangla claim detection, specifically designed to address the task of identifying the check-worthiness of textual claims. Table 3.4 provides a statistical overview of the CheckBanC dataset, highlighting notable distinctions across three classes. It contains a total of 10,023 samples, divided into three categories: Check-Worthy Claims (CWC) with 3,476 samples, Weak Factual claims (WFC) with 2,957 samples, and Non-Claims (NC) with 3,590 samples. This balanced distribution ensures adequate representation of each category, enabling the development of robust and generalizable models. The dataset includes a total of 119,008 words, with each category contributing a substantial amount: CWC includes 45,512 words, WFC contributes 33,940 words, and NC accounts for 39,556 words. Additionally, the vocabulary size of 16,621 unique words highlights the linguistic richness and diversity of the dataset, with CWC, WFC, and NC contributing 8,905, 7,816, and 8,165 unique words, respectively.

Table 3.4: Statistics of our CheckBanC dataset

	CWC	**WFC**	**NC**	**Total**
Total samples	3476	2957	3590	10023
Total words	45512	33940	39556	119008
Unique words	8905	7816	8165	16621
Max. length(words)	43	47	49	49
Avg. length(words)	13	12	11	12

The sentence length within the dataset varies, adding further complexity to the task. The longest sentence spans 49 words, found in the NC category, while the average sentence length across all categories is 12 words. The CWC category has slightly longer sentences on average, with 13 words, compared to WFC and NC, which average 12 and 11 words, respectively. This variability in sentence structure and length ensures that the dataset encompasses both concise and verbose expressions, reflecting the diverse nature of real-world claims.

Figure 3.5 presents a box plot depicting the distribution of word counts across three classes in our dataset. Each box shows the interquartile range (IQR), with the median line within each box indicating the median word count for each class. The spread of word counts is similar across the three classes, with CWC generally having a slightly higher median and a larger upper range. Outliers, shown as individual red points above the main distribution, indicate instances where samples exceed typical word counts. These are particularly noticeable for the CWC class. The box plot pattern suggests that the CWC samples often

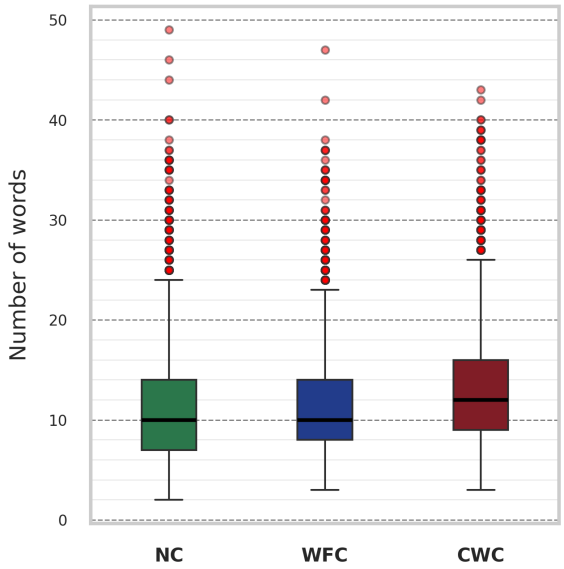


Figure 3.5: Box-plot of word count distribution of different classes of CheckBanC.

contain more extensive information, as opposed to the generally shorter lengths observed in NC and WFC samples.

Our CheckBanC dataset's extensive size, balanced distribution, and linguistic diversity make it a valuable resource for Bangla claim detection research. The dataset provides a balanced and diverse collection of claims and includes various linguistic structures and semantic contexts, making it a comprehensive benchmark for Bangla claim detection research. Its detailed annotation and carefully curated samples ensure that models trained on this dataset can achieve both accuracy and generalizability.

Chapter 4

Methodology

The primary objective of our work is to classify the texts into one of the three classes. Several machine learning, deep learning, and transformer-based models were trained and fine-tuned with our developed dataset for baseline comparison. Figure 4.1 presents an overview of the baseline models for check-worthy claim detection using the CheckBanC dataset. The process begins with pre-processing the raw text to prepare it for analysis. The pre-processed data is then utilized in three different modeling approaches. Machine learning models extract features such as n-grams and TF-IDF, which are processed by classifiers like Logistic Regression and Random Forest. Deep learning models employ embeddings like bi-grams and FastText, feeding them into architectures such as DNN and LSTM. Lastly, transformer-based models, including Bangla-BERT and DistilmBERT, utilize tokenized text for advanced contextual understanding.

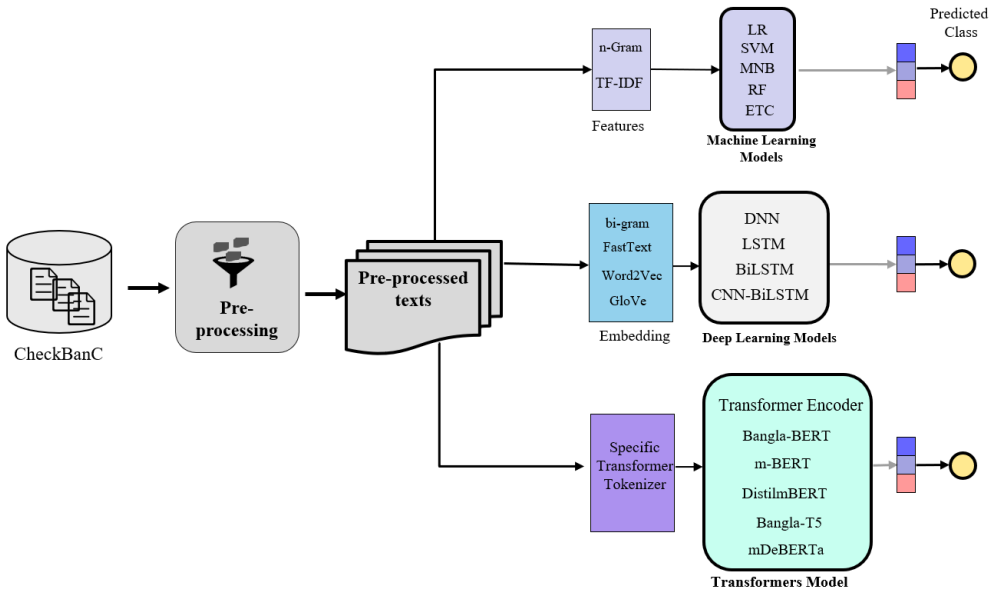


Figure 4.1: Overview of the baseline methods.

4.1 Preprocessing and Feature Extraction

The text data was preprocessed to eliminate special characters, punctuation, and extraneous whitespace. To prepare the feature vectors for machine learning models, Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer was employed to convert the text into numerical representations. For deep learning models, pre-trained Bengali FastText word embeddings were utilized to capture semantic and syntactic information. Finally, for pre-trained transformer models, their respective tokenizers were employed to transform the text into appropriate sequences.

- **TF-IDF:** The TF-IDF [56] vectorizer captures the importance of words within a document relative to the entire corpus, effectively distinguishing frequent yet informative terms. We used an n-gram range of (1,2), capturing both unigrams and bigrams features. To balance computational efficiency and representation power, the most frequent 15000 features are considered. This approach allowed us to capture meaningful co-occurrences of words and phrases.

- **FastText:**For the deep learning-based approach, we utilized a pre-trained FastText word embedding specifically trained on Bengali text [57] . FastText offers robust semantic representations by capturing morphological characteristics of words [58]. The pre-trained embeddings had a dimensionality of 300, enabling us to leverage rich contextual information encoded in the vector representations. This word vector model was employed to initialize the embedding layer in our deep learning architecture, allowing the network to benefit from pre-learned semantic knowledge.

4.2 Machine Learning Models

We employed five baseline machine learning classifiers: Logistic Regression (LR), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), Random Forest (RF), and Extra Trees Classifier (ETC). These models were chosen for their diverse approaches to handling text data, complexity, and adaptability. All the baseline machine learning models were trained on the TF-IDF features extracted from the training data. Table 4.1 summarizes the hyperparameters configured for each model. The LR model was chosen for its simplicity and effectiveness in linearly separable tasks, employing an 'l1' penalty to add sparsity, which helps in feature selection. The 'liblinear' solver and a maximum of 100 iterations were specified to ensure efficient convergence for this text classification task.

The SVM model, using a 'sigmoid' kernel and gamma set to 1.0, was selected for its ability to handle non-linear relationships in high-dimensional spaces, making it suitable for nuanced sentence classification in the claim detection task. The degree parameter was set to 3 to further refine decision boundaries, allowing for more flexible classification.

MNB was employed for its probabilistic approach, as it performs well with text data by modeling word frequency distributions, making it suitable for distinguishing claims classes. The model's smoothing parameter alpha was set to 1.0 to manage zero-frequency issues.

The RF and ETC models were both selected to leverage ensemble methods for a higher generalization capability. By combining multiple decision trees, RF and ETC can capture intricate patterns in claim classification, especially when individual claims may have diverse and subtle features. Both models were configured with the 'gini' criterion for consistency in measuring split quality, ensuring that nodes are split based on maximum information gain.

Table 4.1: Summary of the hyperparameters for machine learning models.

Classifier	Parameters
LR	solver = 'liblinear', penalty = 'l1', class_weight = none, max_iter = 100
SVM	kernel = 'sigmoid', gamma = 1.0, cache_size = 200, class_weight = none, degree = 3
MNB	alpha = 1.0, force_alpha = 'warn', fit_prior = True, class_prior = None
RF	n_estimators = 100, criterion = 'gini', min_samples_split = 2, min_samples_leaf = 1, random_state = 2
ETC	n_estimators = 100, criterion = 'gini', min_samples_split = 2, min_samples_leaf = 1, random_state = 2

4.3 Deep Learning Models

Various deep learning models were trained to leverage their ability to capture contextual relationships and sequential dependencies in textual data. These architectures included simple Deep Neural Networks (DNNs), Long Short-Term Memory network (LSTM), Bidirectional Long Short-Term Memory network (BiLSTM), and a hybrid Convolutional Neural Network with BiLSTM (CNN-BiLSTM). The architectural decisions and hyperparameters for each model were optimized for effectively learning semantic patterns in textual data, as summarized in Table 4.2.

4.3.1 DNN

The DNN models served as baseline deep learning classifiers, utilizing both bi-gram and pre-trained Fasttext word embedding features. The DNN(bi-gram) model was designed to leverage combination of uni-gram and bi-gram features, capturing short-term dependencies within text. The DNN model utilizing bi-gram feature employed 4 layers, whereas the DNN trained on pre-trained FastText word vectors included 6 layers for deeper feature learning. Both used the ReLU activation function for non-linear transformations, with a dropout rate of 0.5 for regularization. Training was optimized using the RMSprop optimizer with a learning rate of 0.001 across 20 and 13 epochs, respectively.

4.3.2 LSTM

The LSTM model was chosen for its proficiency in modeling sequential dependencies, crucial for understanding context in claim detection. The network was implemented with 4 layers and an LSTM cell size of 128 to balance model complexity and capture temporal dependencies. The 'tanh' activation function was selected for its ability to represent positive and negative states in the sequence. A sequence length of 50 was used, ensuring long-term dependencies were captured. The model was trained using the RMSprop optimizer at a learning rate of 0.005 and batch size of 32 over 18 epochs, providing a balance between learning and convergence speed.

4.3.3 BiLSTM

BiLSTM extended the LSTM's capabilities by incorporating bidirectional context, enabling the model to analyze sentences from both forward and backward perspectives. Configured similarly to the LSTM in terms of vocabulary size and sequence length, the BiLSTM used 6 layers and 128 LSTM cells. A 'tanh' activation function was used, with a dropout rate of 0.3 to mitigate overfitting. Training used the RMSprop optimizer at a learning rate of 0.001 and batch size of 32 over 17 epochs.

4.3.4 CNN-BiLSTM

To combine the feature extraction strengths of CNNs with the sequential learning capabilities of BiLSTM, a hybrid CNN-BiLSTM model was employed. The CNN architecture extracted local patterns within sequences, while the BiLSTM layers captured temporal dependencies. The CNN component used a kernel size

of 3, filter size of 200, and max pooling. Outputs from the CNN were fed into an 8-layer BiLSTM with 128 LSTM cells to model sequential dependencies. ReLU activation functions enabled efficient feature mapping, while dropout (0.5) provided regularization. The model was trained using RMSprop with a learning rate of 0.001 and a smaller batch size of 16 for 12 epochs.

4.4 Transformer Models

The introduction of attention-based Transformer [59] opens a new era for nlp tasks. Different variations of models from the transformer architecture developed for faster and efficient performance [60]. Recent studies showed that transformer-based models provide better performance for various state-of-the-art claim detection techniques [54]. The present studies experimented with a basic transformer encoder and various pre-trained transformer models from the HuggingFace[1] library. The models are fine-tuned on our developed CheckBanC dataset for optimal performance. The fine-tuned hyperparameters of the transformer models are shown in Table 4.3.

4.4.1 Transformer Encoder

We implemented a transformer encoder model following basic transformer architecture [59]. The transformer encoder model utilizes a sequence length of 50 and an embedding dimension of 512 to represent text inputs. The architecture includes 8 attention heads within the encoder to capture complex relationships in the data. The model is trained using a learning rate of 0.0001, a batch size of 16. We utilized positional embeddings to preserve sequence order.

4.4.2 Bangla-BERT

BERT (Bidirectional encoder representations from transformers) [61] is a transformer-based pre-trained model. Bangla-BERT [62] is a pre-trained language model of Bengali language using mask language modeling trained on two large Bangla corpus OSCAR[2] and Bengali Wikipedia Dump Dataset[3]. We used *"sagorsarker/bangla-bert-base"* model which follows the bert-base-uncased model structure of multi-head attention containing 12 heads. The architecture of the model has 12 layers, 768 hidden, and 110M parameters.

[1] https://huggingface.co/models
[2] https://oscar-corpus.com/
[3] https://dumps.wikimedia.org/bnwiki/latest/

Table 4.2: Summary of the hyperparameters for deep learning models.

Hyperparameter	**Hyperparameter space**	**DNN(bi-gram)**	**DNN**	**LSTM**	**BiLSTM**	**CNN-BiLSTM**
Max tokens	[10K, 15K, 20K]	10K	10K	10K	15K	15K
Sequence length	[25, 30, 40, 50]	max tokens	50	50	50	50
Embedding dimension	[100, 200, 300]	_	300	300	300	300
No of layer	[3, 4, 6, 8, 10]	4	6	4	6	8
Kernel size	[3, 5]	_	_	_	_	3
LSTM cell	[32, 64, 128]	_	_	128	128	128
Activation	[*relu*, *tanh*]	*relu*	*relu*	*tanh*	*tanh*	*relu*
Filter size	[100, 200]	_	_	_	_	200
Pooling	[max, average]	_	max	_	_	max
Dropout rate	[0.2, 0.3, 0.5]	_	_	0.5	0.3	0.5
Optimizer	[*rmsprop*, *adam*]	*rmsprop*	*rmsprop*	*rmsprop*	*rmsprop*	*rmsprop*
Learning rate	[0.01, 0.001, 0.005 0.0001]	0.001	0.001	0.005	0.001	0.001
Batch size	[16, 32, 64, 128]	32	32	32	32	16
No of epoch	[_]	20	13	18	17	12

4.4.3 m-BERT

This is the multilingual version of BERT [61] which is pre-trained on the largest Wikipedia data of the top 104 languages utilizing masked language modeling (MLM). We used the *"bert-base-multilingual-cased"* model which contains 12 layers, 768-hidden, 12 heads, and 110 M parameters.

4.4.4 DistilmBERT

DistilBERT [63] is a distilled version of the BERT model. The general architecture of this model is similar to the BERT-base with the number of layers reduced by a factor of 2. Knowledge distillation and triple loss combination were incorporated during the pre-training phase of the model. It is a lighter (40%) and faster (60%) version of the BERT and performs almost similarly (97%) in natural language understanding. In this work, we used *"distilbert-base-multilingual-cased"* model which is a multilingual model trained on the Wikipedia concatenation data in 104 different language including Bengali. The model includes 134M parameters, 6 layers, 768 dimension, and 12 head attention mechanism. On average DistilmBERT performs twice as faster than m-BERT.

4.4.5 Bangla-T5

T5 [26] (Text-to-Text Transfer Transformer) is an encoder-decoder transformer model. It is pre-trained on a multi-task mixture of unsupervised and supervised tasks. The multilingual version of T5 is known as mT5 [64] which is pre-trained of Common Crawl based dataset from 101 languages. Bangla-T5 [65] is a text-to-text transformer pre-trained with "Span corruption objective on 5.25 million Bangla text documents from web sources. Bangla-T5 outperformed the multilingual version of T5 which is mT5 in various nlp tasks for the Bengali language. We used the *"csebuetnlp/banglat5"* model which has 12 layers, 12 attention heads, a hidden size of 768, and feed-forward size of 2048 with GeGLU activation.

4.4.6 mDeBERTa

DeBERTa (Decoding-enhanced BERT with disentangled attention) [43] is a transformer based model utilizing disentangled attention and enhanced mask decoder to improve the performance of BERT and RoBERTa models. The multilingual version of the model mDeBERTa [44] was trained with CC100 [4] multilingual data. We used the *"microsoft/mdeberta-v3-base"* model and fine-tuned on our

[4] https://huggingface.co/datasets/statmt/cc100

dataset. The model has 12 layers, 12 attention heads, 768 hidden size, and 86M backbone parameters.

Table 4.3: Fine-tuned hyperparameters of the transformer models.

Transformer Model	Maximum sequence	Learning rate	Batch size	Epochs
Transformer encoder	50	$1e^{-4}$	16	16
Bangla-BERT	100	$5e^{-5}$	16	10
m-BERT	80	$1e^{-4}$	16	8
DistilmBERT	80	$5e^{-6}$	16	8
Bangla-T5	100	$5e^{-5}$	16	12
mDeBERTa	80	$5e^{-5}$	16	10

4.5 Proposed Hybrid Feature based Model

Our proposed hybrid feature-based model introduces a novel approach to detecting the check-worthiness of claims by combining traditional feature engineering techniques with advanced transformer-based deep learning architectures. The architecture of our proposed model is shown in Figure 4.2. The core contribution of this model lies in its ability to capture both local syntactic patterns and global semantic context, essential for detecting check-worthiness of claims.

The first feature component captures local syntactic relationships within the text using bigram features. The second component leverages the DistilBERT model, a lightweight transformer architecture pre-trained on a massive corpus, to capture deep contextual information. DistilBERT transforms the input text into contextualized embeddings, capturing the deeper semantic and contextual meaning of words and sentences. This is particularly important for understanding the nuances of claim texts, where the meaning of a sentence often relies on broader context rather than individual words.

4.5.1 Bigram Feature Generation

For detecting syntactic relationships and local dependencies between adjacent words in a claim, we employed bigram features. The TextVectorization layer is employed to extract combinations of unigram and bigram features from the text, providing a representation of word co-occurrences and local dependencies. The vocabulary size is set to 20,000 for considering the most frequent 20,000 features, and the features are encoded using a multi-hot encoding format, which captures the presence of word pairs in the text.

The bigrams help the model capture essential text patterns, which can be particularly useful in understanding simple but important claim structures, such

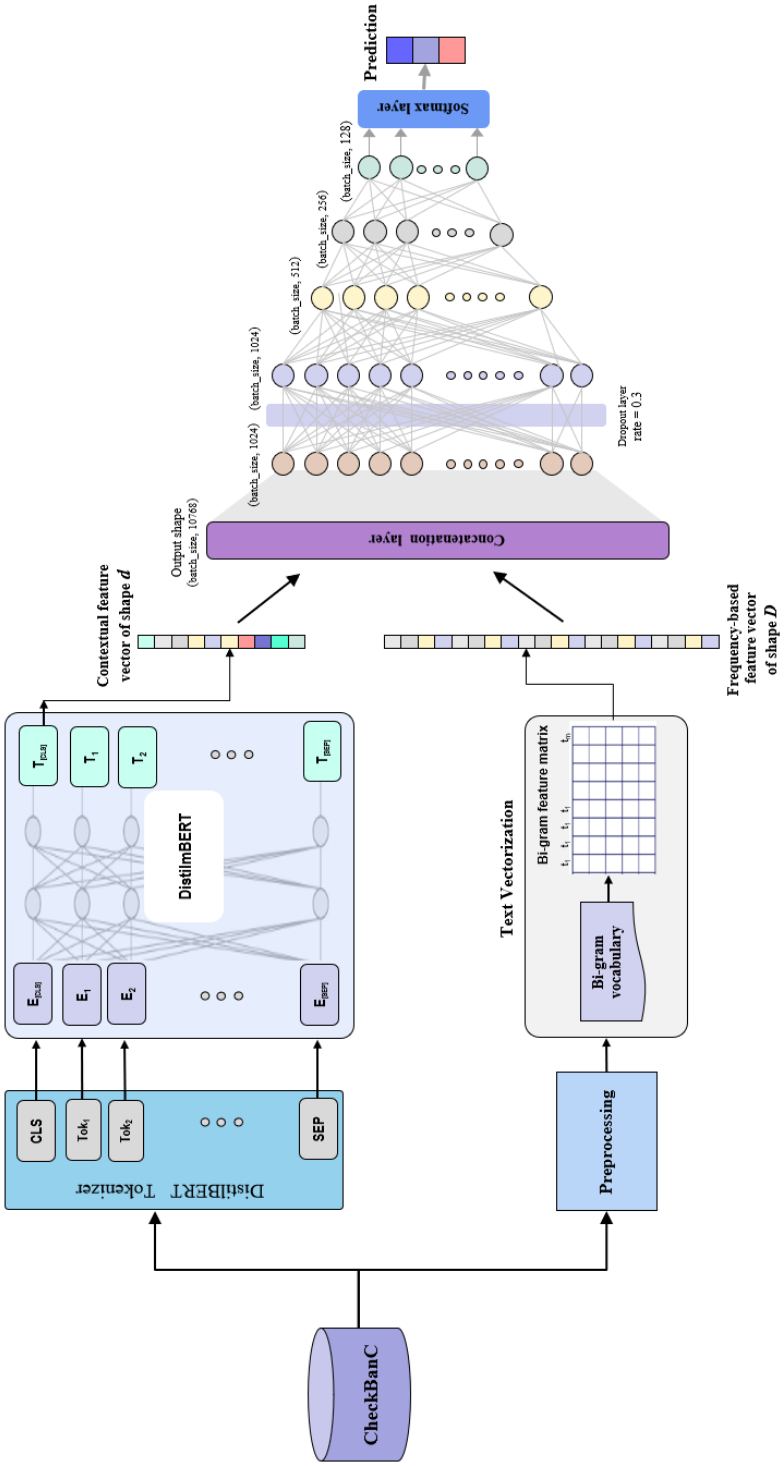


Figure 4.2: Proposed DistilBERT and bi-gram feature fusion neural network.

as recurring phrases or common collocations. By focusing on bigrams, the model effectively handles word-level interactions that are critical for detecting checkworthiness in claims. The generation process of the combination of bigram feature is shown in Algorithm 2. At first the bigram feature matrix F_{bigram} is initialized to zeros. A bigram tokenizer is used to tokenize the input texts and extract a vocabulary limited to $V_{\max}$ tokens. For each input text t_i, the tokenizer generates a sequence of bigrams. The bigram feature vector for each text is initialized with zeros, and each bigram is checked against the dictionary. If the bigram already exists in the dictionary, the corresponding index in the feature vector is set to 1. If it does not exist and the dictionary size is less than $V_{\max}$, the bigram is added to the dictionary, and its corresponding feature vector position is set to 1. The feature vector for each text is updated in the feature matrix F_{bigram}.

Algorithm 2: Generating Bigram features

Input: $X \in \mathbb{R}^n$: Set of input texts, where n is the number of texts
        $V_{\max}$: Maximum number of tokens
Output: $F_{\text{bigram}} \in \mathbb{R}^{n \times V_{\max}}$: Bigram feature matrix

Dict $\leftarrow$ {} {Initialize dictionary for bigrams }
Initialize bigram feature matrix: $F_{\text{bigram}} \leftarrow \mathbf{0}_{n \times V_{\max}}$
Tokenizer $\leftarrow$ BiGramTokenizer(max_tokens $= V_{\max}$)
Tokenizer.fit_on_texts(X)
$V \leftarrow$ Tokenizer.get_vocabulary() {Extract vocabulary (limited to $V_{\max}$ tokens)}
for $i = 1$ to n **do**
    $W \leftarrow$ Tokenizer.tokenize(t_i)
    $B_i \leftarrow$ generate_bigrams(W)
    $f^i_{\text{bigram}} \leftarrow \mathbf{0}_{V_{\max}}$ {Extract vocabulary (limited to $V_{\max}$ tokens)}
    for each bigram $b \in B_i$ **do**
        if $b \in$ Dict **then**
            index $\leftarrow$ Dict[b]
            f^i_{bigram}[index] $\leftarrow 1$
        else
            if |Dict| $< V_{\max}$ **then**
                Dict[b] $\leftarrow$ |Dict|
                f^i_{bigram}[Dict[b]] $\leftarrow 1$
            end if
        end if
    end for
    Update the feature matrix: $F_{\text{bigram}}[i, :] \leftarrow f^i_{\text{bigram}}$
end for
return F_{bigram}

4.5.2 Contextualized Embeddings with DistilBERT

Pretrained transformer-based contextual embeddings have significantly advanced the field of natural language processing [66]. It has the ability to capture the nuances of language, understanding how a word's meaning can change based on its surrounding context. The second part of our feature extraction pipeline utilizes DistilBERT, a pre-trained transformer model, to obtain contextualized word embeddings. DistilBERT captures the semantic meaning of words and their relationships in context, making it exceptionally well-suited to handle complex claim-related texts where word meanings can change depending on the surrounding context. The model uses DistilBERT's embeddings to represent each input text as a fixed-size vector, encapsulating deep contextual information that is crucial for understanding the nuanced meaning of claims. This transformer-based approach allows the model to perform well on tasks involving ambiguous, domain-specific language. The procedure of generating contextual embedding is shown in Algorithm 3.

The input sequence S is augmented with special tokens: $[CLS]$ (classification token) and $[SEP]$ (separator token). The augmented sequence is passed through an embedding layer to generate initial word embeddings. For each attention layer, the sequence embeddings are split into multiple attention heads. Each attention head computes queries, keys, and values using the projection matrices. The attention scores are calculated by applying the softmax function on the dot product of queries and keys, scaled by $\sqrt{d_k}$. The attention scores are used to weight the values, and the results are concatenated and linearly transformed. After each attention layer, residual connections are applied to retain the previous layer's information, followed by layer normalization to stabilize training. The output corresponding to the $[CLS]$ token is extracted after all attention layers, representing the contextual embedding of the entire sentence. This embedding encapsulates the global semantic context of the claim, providing insights into its deeper meaning and the relationships between words beyond simple word co-occurrence.

4.5.2.1 DistilBERT Architecture

The DistilBERT architecture, as illustrated in Figure 4.3, is a compressed version of the BERT base model designed to achieve comparable performance with significantly fewer computational resources. The embedding layer in DistilBERT is identical to that of BERT. It maps input tokens into dense vector representations that capture semantic information and serve as the input to the subsequent

Transformer layers. The embedding layer in DistilBERT is identical to that of BERT. It maps input tokens into dense vector representations that capture semantic information and serve as the input to the subsequent Transformer layers.

One of the key modifications in DistilBERT is the reduction in the number of Transformer layers. While the BERT base consists of 12 Transformer layers, DistilBERT reduces this to 6 layers, effectively halving the depth of the network

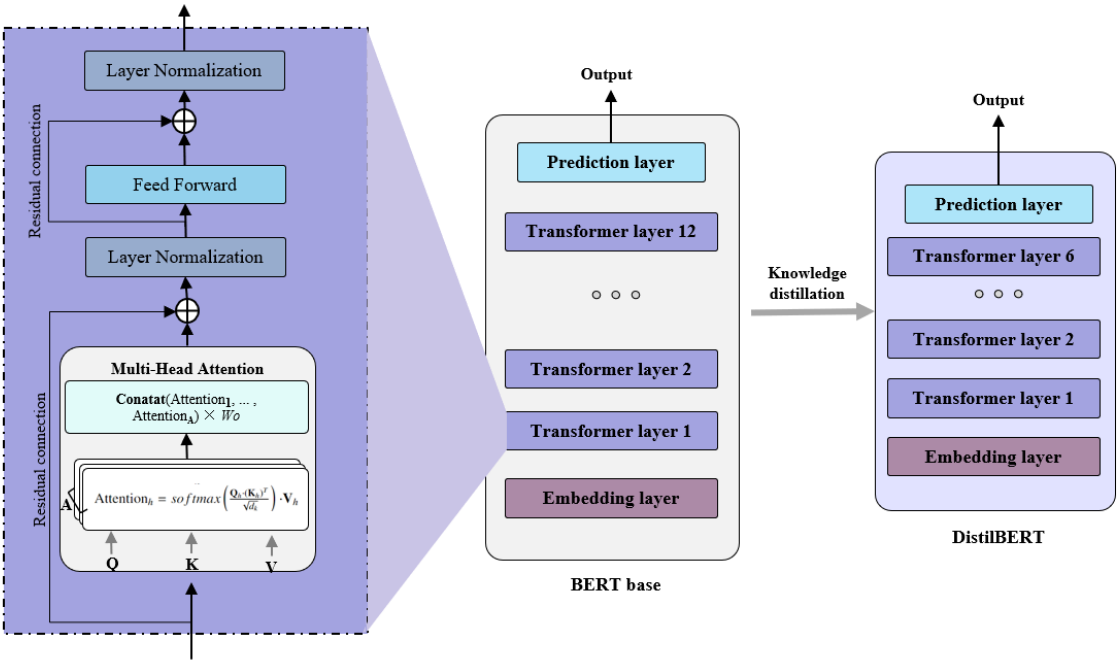


Figure 4.3: DistilBERT architecture with attention mechanism.

To train DistilBERT, knowledge from the BERT base model (teacher) is transferred using knowledge distillation. This process ensures that the smaller model learns to mimic the behavior of the larger model while maintaining efficiency. Knowledge distillation involves transferring the knowledge of the teacher model (BERT base) to the student model (DistilBERT). The teacher's logits, hidden states, and output probabilities serve as training signals for the student. The knowledge distillation process involves calculating **distillation loss**, **cosine embedding loss**, and **supervised loss**. The primary training objective for the student is the distillation loss, which is the cross-entropy between the teacher's soft target probabilities and the student's predicted probabilities. The distillation loss is given by the following equation:

$$L_{ce} = -\sum_{i} t_i \cdot \log(s_i) \tag{4.1}$$

Where, L_{ce} is the cross-entropy distillation loss. t_i is the probability predicted

by the teacher model for class i. s_i is the probability predicted by the student model for class i. The loss is computed over all classes i, and it encourages the student to match the teacher's output distribution.

The softmax function with a temperature parameter T is used to control the smoothness of the output distribution:

$$p_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)} \tag{4.2}$$

Where, p_i is the softmax probability for class i. z_i is the logit or score predicted by the model for class i. T is the temperature parameter that controls the smoothness of the output distribution. A higher T produces a softer probability distribution, while $T = 1$ gives the standard softmax output.

The cosine embedding loss aligns the directions of the hidden state vectors of the student and teacher models. Given two vectors (from the teacher) and (from the student), the cosine embedding loss is defined as: by minimizing this loss, the student's hidden representations become more similar to the teacher's, enabling it to replicate the teacher's learned features more effectively. The cosine embedding loss is given by the following equation:

$$L_{cos} = 1 - \frac{\mathbf{h}_{\text{student}} \cdot \mathbf{h}_{\text{teacher}}}{\|\mathbf{h}_{\text{student}}\| \|\mathbf{h}_{\text{teacher}}\|} \tag{4.3}$$

Where, L_{cos} is the **cosine embedding loss**. $\mathbf{h}_{\text{student}}$ is the hidden state vector from the student model. $\mathbf{h}_{\text{teacher}}$ is the hidden state vector from the teacher model. The dot$(\cdot)$ denotes the dot product and $\| \cdot \|$ denotes the Euclidean norm.

In addition to the distillation and cosine embedding losses, DistilBERT is trained with a standard supervised loss, such as the Masked Language Modeling (MLM) loss. Supervised loss measures how well the student predicts the correct labels based on the ground truth in the dataset. The total loss L_{total} is a weighted combination of the distillation loss, the supervised loss, and the cosine embedding loss:

$$L_{total} = \alpha \cdot L_{ce} + \beta \cdot L_{mlm} + \gamma \cdot L_{cos} \tag{4.4}$$

Where, L_{total} is the total loss. L_{ce} is the distillation loss. L_{mlm} is the supervised loss (e.g., masked language modeling loss). L_{cos} is the cosine embedding loss. α, β, and γ are the weights for each loss component, controlling their contribution to the total loss. By carefully tuning the weights, the total loss ensures that the student retains the teacher's efficiency while remaining lightweight, computationally efficient, and effective across multiple tasks. This synergy enables

the student to closely match the teacher's performance despite having fewer parameters and reduced complexity.

Algorithm 3: Generating contextual embedding of the sentence

Input:
  $\mathbf{S} = \{w_1, w_2, \ldots, w_n\}$
  $\mathbf{W}_Q$: Query projection matrix
  $\mathbf{W}_K$: Key projection matrix
  $\mathbf{W}_V$: Value projection matrix
  $\mathbf{W}_O$: Output projection matrix
  $\mathbf{B}_E$: Embedding weights
  Number of attention heads A
  Number of attention layers L_{att}
Output:
  $\mathbf{T}_{[CLS]} \in \mathbb{R}^d$: Final state corresponding to [CLS] (contextual sequence embedding)

$S' \leftarrow \{I_{[CLS]}, I_1, I_2, \ldots, I_{[SEP]}\}$
Initialize $\mathbf{H}_0 = \text{Embedding}(\mathbf{S}'; \mathbf{B}_E)$
for layer = 1 **to** L_{att} **do**
  for h = 1 **to** A **do**
    Compute Queries, Keys, Values:
      $\mathbf{Q}_h = \mathbf{H}_{layer-1} \cdot \mathbf{W}_h^Q$
      $\mathbf{K}_h = \mathbf{H}_{layer-1} \cdot \mathbf{W}_h^K$
      $\mathbf{V}_h = \mathbf{H}_{layer-1} \cdot \mathbf{W}_h^V$
      $\text{Attention}_h = softmax\left(\frac{\mathbf{Q}_h \cdot (\mathbf{K}_h)^T}{\sqrt{d_k}}\right) \cdot \mathbf{V}_h$
  end for
  Concatenate and Linearly Transform:
    $\mathbf{H}_{layer} = \text{Concat}(\text{Attention}_1, \text{Attention}_2, \ldots, \text{Attention}_A) \cdot \mathbf{W}_O$
  Apply Layer Normalization and Residual Connection:
    $\mathbf{H}_{layer} = \text{LayerNorm}(\mathbf{H}_{layer-1} + \mathbf{H}_{layer})$ {Residual connection}
end for
Extract [CLS] Token Embedding:
  $\mathbf{T}_{[CLS]} = \mathbf{H}_{L_{att}}[:, 0, :]$
Return $\mathbf{T}_{[CLS]}$

4.5.3 Concatenation of Features

The bigram features and the contextual embeddings from DistilBERT are concatenated to create a unified feature vector. This concatenation enables the model to learn from both the local and global features simultaneously, effectively combining syntactic and semantic information. The concatenated feature vector is then passed through multiple fully connected layers (with ReLU activation

functions) to learn complex interactions between these features. The dense layers help refine the representations further, allowing the model to capture intricate patterns between the bigram features and contextual embeddings. This concatenation ensures that both the syntactic and semantic aspects of the claim are captured in the final representation, allowing the model to make more informed decisions about the check-worthiness of a claim.

4.5.4 Deep Neural Network Architecture

The hybrid feature vector is processed by a deep neural network comprising several dense layers. The dense layers reduce the high-dimensional feature space progressively from 1024 to 128 units, allowing the model to distill important information while maintaining flexibility in decision-making. The final layer is a softmax classifier, which outputs the probability distribution for each claim class CWC, WFC, or NC. The use of deep layers allows the model to capture increasingly abstract representations of the input features, enhancing its ability to learn complex relationships in the data. Additionally, a dropout layer with a rate of 0.5 is used for regularization, reducing the risk of overfitting during training.

4.5.5 Model Training and Optimization

The proposed model is trained using the 'RMSprop' optimizer with a learning rate of 0.0001. This choice of optimizer helps the model converge smoothly and avoids issues such as exploding or vanishing gradients, which are common in deep learning models. The model is trained over 50 epochs with a batch size of 32, allowing it to learn from a large number of samples while maintaining a balance between training speed and model accuracy. To prevent overfitting and ensure generalization to unseen data, we incorporate early stopping and model checkpointing. Early stopping monitors validation accuracy and halts training when no improvement is observed for a predefined number of epochs (20 in this case). Model checkpointing ensures that the best-performing model, in terms of validation accuracy, is saved and used for testing.

Chapter 5

Experimental Results and Evaluation

This section presents a comprehensive analysis of the performance of our proposed approach for check-worthy claim detection. It outlines the experimental setup, including hardware configurations and software tools, to establish the foundation for reproducibility. The section further discusses the metrics used to evaluate the models, ensuring a fair and comprehensive assessment of their performance. Additionally, it includes an in-depth analysis of the results, highlighting the strengths of the models and examining cases of misclassification to uncover potential limitations. Finally, it compares our approach with existing state-of-the-art techniques, demonstrating its superiority and contributions to advancing the field of claim detection.

5.1 Experimental Settings

The experiments were conducted on the Google Colaboratory cloud platform, leveraging a computational environment equipped with a T4 GPU featuring 15 GB of GPU memory. The system provided 12.7 GB of RAM and 112.6 GB of disk space. Python 3.10 was used as the programming environment. NumPy (1.26.4) and Pandas (2.1.4) were employed for data preprocessing. Machine learning models were developed and trained using Scikit-learn (2.1.4), while deep learning architectures were implemented with Keras (3.4.1) and TensorFlow (2.17.0). Transformer-based models were implemented using PyTorch (2.4.0+cu121).

5.2 Evaluation Metrics

The performance of the models was evaluated using several standard classification metrics, which include Precision, Recall, F1-score, Macro-average F1, Weighted-average F1, and Geometric Mean Score.

- **Precision (P)**: Precision measures the proportion of correctly predicted positive observations to the total predicted positives.

$$P = \frac{True\,positive}{True\,positive + False\,positive} \tag{5.1}$$

- **Recall (R)**: Recall, or sensitivity, measures the proportion of correctly predicted positive observations to all actual positives.

$$R = \frac{True\,positive}{True\,positive + False\,negative} \tag{5.2}$$

- **F1-Score**: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of performance.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{5.3}$$

- **Macro-average F1 (MF1)**: This metric computes the F1-score independently for each class and then takes the average. It treats all classes equally, regardless of their frequency

$$MF1 = \frac{1}{c}\sum_{i=1}^{c} F1_i \tag{5.4}$$

  where c is the number of classes.

- **Weighted-average F1 (WF1)**: The weighted-average F1 score computes the F1 score for each class and weights them by the number of samples in each class. This ensures that the performance of large classes does not completely dominate the evaluation.

$$WF1 = \frac{1}{N}\sum_{i=1}^{c}(n_i \times F1_i) \tag{5.5}$$

  where N is the total number of samples, n_i is the number of samples in class i.

- **Geometric Mean Score or G-Score (G)**: The G-Score (also called the Geometric Mean Score or G-Mean) is a metric used to evaluate the performance of a classification model, especially in imbalanced datasets [67]. It measures the balance between sensitivity across all classes. Its significance lies in its ability to avoid favoring majority classes over minority classes by focusing on per-class recall. It is calculated by equation 5.6.

$$G = \sqrt{\text{sensitivity} \times \text{specificity}} \tag{5.6}$$

For multi-class problems, it is a higher root of the product of sensitivity for each class. We calculated the G score following equation 5.7.

$$G = \sqrt[c]{\text{sensitivity}_1 \times \text{sensitivity}_2 \times \cdots \cdot \text{sensitivity}_c} \tag{5.7}$$

where c is the number of classes.

The weighted-average F1 score is considered the primary metric for determining model superiority as it ensures that the performance of the model is not skewed by the size of each class and accurately reflects its ability to handle all classes proportionally. As our dataset is slightly imbalanced, we also calculated the geometric mean (G) score to ensure that poor performance in one class has a significant impact on the overall score.

5.3 Performance Analysis

The dataset was split into three sets: train (80%), validation (10%), and test (10%). Train, validation and test sets contain 8018, 1002, and 1003 samples respectively. The stratified sampling method [68] was employed to maintain original class proportions when partitioning the dataset into train, test, and validation sets. Figure 5.1 shows the class distribution of the data samples in three splits. The models were trained with our train set, while the optimal hyper-parameters were selected based on the performance on the validation set. The performance of the trained models was evaluated using the unseen instance of the test set. Table 5.1 summarizes the evaluation results in terms of precision (P), recall (R), F1 score (F1), macro-average F1 (MF1), weighted average F1 (WF1), and accuracy.

Among traditional machine learning models, Support Vector Machine (SVM) demonstrates the best performance with an accuracy of 77% and WF1 of 77%. SVM's ability to effectively model linear decision boundaries explains its superior

Table 5.1: Performance comparison of different models with our proposed technique on the test set.

Model	CWC			WFC			NC						
	P	R	F1	P	R	F1	P	R	F1	**Accuracy**	**MF1**	**WF1**	**G**
LR	0.83	0.75	0.79	0.64	0.60	0.62	0.72	0.83	0.77	0.74	0.73	0.73	0.72
SVM	0.85	0.80	0.82	0.69	0.65	0.67	0.75	0.84	0.80	0.77	0.76	0.77	0.76
MNB	0.82	0.85	0.83	0.75	0.45	0.56	0.68	0.91	0.78	0.75	0.73	0.73	0.70
RF	0.85	0.79	0.82	0.71	0.56	0.63	0.69	0.86	0.77	0.75	0.74	0.75	0.72
ETC	0.85	0.82	0.83	0.75	0.54	0.63	0.70	0.90	0.78	0.76	0.75	0.75	0.74
DNN (bi-gram)	0.89	0.85	0.87	0.70	0.75	0.73	0.83	0.81	0.82	0.81	0.80	0.81	0.80
DNN (FT)	0.84	0.83	0.83	0.66	0.68	0.67	0.81	0.80	0.80	0.77	0.77	0.77	0.77
LSTM (FT)	0.85	0.83	084	0.69	0.64	0.66	0.76	0.83	0.79	0.77	0.76	0.77	0.76
BiLSTM(FT)	0.83	0.77	0.80	0.62	0.71	0.66	0.80	0.76	0.78	0.75	0.75	0.75	0.75
CNN-BiLSTM (FT)	0.84	0.77	0.80	0.62	0.71	0.66	0.80	0.76	0.78	0.75	0.75	0.76	0.75
Transformer Encoder	0.88	0.75	0.81	0.66	0.60	0.63	0.71	0.87	0.78	0.75	0.74	0.76	0.73
Bangla-BERT	0.83	0.86	0.84	0.78	0.56	0.65	0.75	0.90	0.82	0.78	0.77	0.78	0.76
m-BERT	0.87	0.86	0.86	0.77	0.67	0.71	0.78	0.87	0.82	0.80	0.80	0.80	0.79
DistilmBERT	0.84	0.87	0.85	0.67	0.72	0.69	0.86	0.75	0.80	0.78	0.78	0.79	0.78
Bangla-T5	0.82	0.86	0.84	0.72	0.56	0.63	0.76	0.87	0.82	0.78	0.76	0.77	0.75
mDeBERTa	0.89	0.82	0.85	0.69	0.79	0.74	0.85	0.81	0.83	0.81	0.80	0.80	0.81
Proposed	0.86	0.90	0.90	0.76	0.83	0.79	0.89	0.81	0.85	**0.84**	0.84	**0.84**	**0.84**

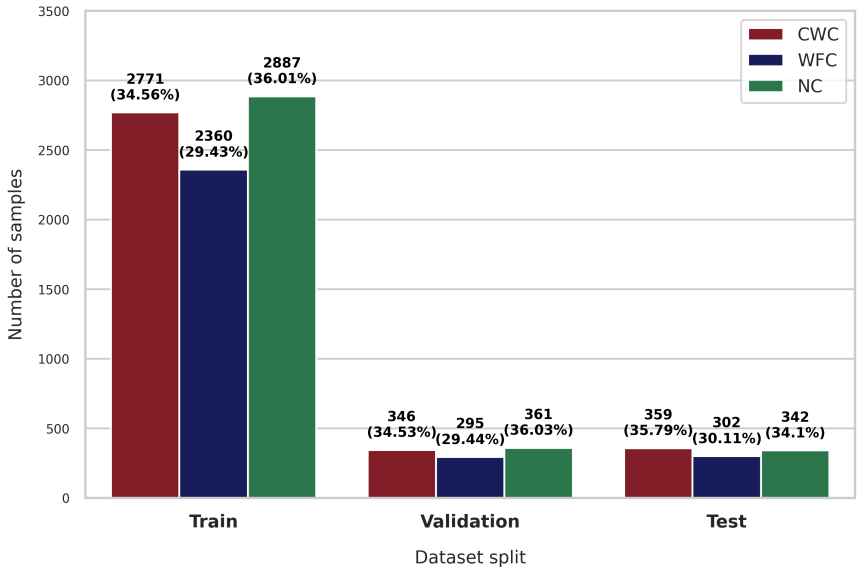


Figure 5.1: Class-wise distribution of data samples across train, validation, and test sets.

performance when handling bigram-based features. In contrast, Multinomial Naïve Bayes (MNB) achieves higher precision in identifying WFC but suffers from a significant drop in recall. This suggests that MNB's assumption of feature independence may not hold well for certain textual patterns, especially in weak claims, which often involve more contextual dependencies.

Random Forest (RF) and Extra Trees Classifier (ETC) perform moderately well but fail to generalize as effectively as SVM. While tree-based models can capture feature interactions, they tend to overfit specific n-gram combinations, which might explain their performance plateau. Additionally, these models may not fully exploit the sequential nature of the text data, as they lack mechanisms for capturing word-order dependencies. These models perform better with TF-IDF features because they emphasize word frequency and short-term word combinations. However, their inability to capture semantic and syntactic relationships limits their performance, especially in distinguishing WFC and NC classes.]

For deep learning models, the DNN model utilizing bi-gram features exhibits a substantial improvement, achieving an accuracy of 81% and WF1 of 81%. This demonstrates that bigram features, when fed into a fully connected network, allow the model to learn higher-level feature abstractions. The bigram features allow the model to capture word co-occurrences, which are particularly effective for the CWC class where specific phrases or contextual clues often distinguish claims. However, methods like LSTM, BiLSTM, and CNN-BiLSTM, which rely on sequential learning, show varied results. While these models perform well for

CWC, their slightly lower performance in the NC and WFC classes suggests that certain contextual relationships in the features may not be fully captured due to the limited sequential complexity of the data. Additionally, the WFC class, being inherently ambiguous, requires a more nuanced understanding of linguistic features that simpler deep learning models fail to capture.

In contrast, Transformer-based models exhibit significant improvements due to their superior capability to model long-range dependencies and contextual relationships within text. The m-BERT and mDeBERTa outperform other models, achieving a WF1 of 0.80. These models leverage pre-trained contextual embeddings, allowing them to understand nuanced differences between the three classes, particularly in WFC, where precision and recall improve substantially. mDeBERTa performs slightly better than other transformers due to its optimized architecture, which effectively integrates relative position embeddings and deeper contextual learning. Similarly, DistilBERT and Bangla-BERT perform competitively. DistilBERT shows an edge in WFC identification due to its efficient distillation process, which preserves key linguistic features. Generative model Bangla-T5, though effective in CWC and NC classes, struggles with WFC due to its generative nature, which may result in the loss of fine-grained discrimination during classification tasks. This highlights the complexity of the WFC class, where subtle linguistic cues make it challenging to distinguish the CWC and NC classes.

The Proposed Hybrid Feature-Based Model achieves the best overall performance, with an accuracy of 85% and WF1 of 85%. The success of this model lies in its ability to integrate bigram features with contextual embeddings. The bigram features provide lexical-level information, capturing local patterns within the text, while the contextual embeddings ensure that global semantic relationships are effectively modeled. This hybrid approach enables the model to overcome limitations faced by individual architectures, such as the inability to generalize for ambiguous classes like WFC. While bigram features allow the model to capture specific word co-occurrences that are critical for distinguishing CWC and NC, the contextual embeddings provide a deeper understanding of semantic relationships within the text. This hybrid approach ensures that the model overcomes the challenges posed by ambiguous features in the WFC class, as evidenced by its substantial recall improvement (0.83) for WFC and strong precision for NC (0.89).

5.3.1 Effect of N-gram Features in Hybrid Feature Fusion

In order to justify the use of bi-grams and evaluate the individual effects of uni-grams, bi-grams, tri-grams, and higher-order n-grams. A detailed performance analysis was conducted on our proposed hybrid feature fusion model, where n-gram features were combined with DistilBERT contextual features to classify claims into one of three categories: CWC, WFC, and NC. The results of this analysis are presented in Table 5.2.

Table 5.2: Performance analysis of different n-grams feature sets on our proposed model.

n-gram	Max tokens	Weighted f1 score
Uni-gram	10000	0.80
	15000	0.81
	20000	0.81
	25000	0.77
Bi-gram	10000	0.82
	15000	0.82
	20000	0.84
	25000	0.79
Tri-gram	10000	0.78
	15000	0.80
	20000	0.79
	25000	0.77

Bi-grams achieve the highest weighted F1 score of 0.84 at a token limit of 20,000, outperforming both uni-grams (0.81) and tri-grams (0.80). Their superior performance can be attributed to their ability to balance the richness of contextual information and computational efficiency in our hybrid feature fusion model.

Uni-grams focus on individual words, which often fail to provide sufficient contextual information about the relationships between words in a sentence. For instance, in the context of claim detection, phrases such as "বেচে আছে" or "বেচে নেই" are better understood when two-word combinations are analyzed rather than isolated words like "বেচে", "আছে", "নেই". Bi-grams capture meaningful word pairs, provide richer semantic information and help the model better differentiate between classes (CWC, WFC, and NC). This improved combination leads to a significant performance boost, as seen in the

While tri-grams provide even longer context windows, they often introduce noise in multiclass classification tasks when used alongside deep contextual features like DistilBERT. Tri-grams may overfit to specific patterns that are less generalizable across classes. Additionally, with tri-grams, the dimensionality of

the feature space increases significantly, which can dilute the contribution of DistilBERT's embeddings in the hybrid model and lead to reduced performance. Bi-grams balance capturing sufficient contextual information (compared to unigrams) and avoiding the sparsity or overfitting issues associated with tri-grams. This makes bi-grams particularly effective in our hybrid model for identifying nuanced distinctions between claims in the CWC, WFC, and NC categories.

In summary, the varying performance across models underscores the importance of both the features used and the architecture's ability to exploit these features. Traditional machine learning and deep models rely heavily on feature engineering, while transformer-based models capitalize on data-driven contextual understanding. The Proposed Hybrid Feature-Based Model bridges these strengths, leveraging both local and global information to achieve state-of-the-art performance. This highlights the necessity of feature integration to address the nuanced nature of claim detection tasks.

5.4 Error Analysis

This section discusses the misclassification trends observed across the models and the limitations that may have contributed to these errors. Table 5.3 highlights the misclassification rates for all models on the training set. Traditional

Table 5.3: Misclassification rate of different models on the test set.

Model	**Misclassification rate(%)**
LR	26.52
SVM	23.13
MNB	25.12
RF	25.02
ETC	23.83
DNN (bi-gram)	19.14
DNN(FT)	22.53
LTSM (FT)	22.63
BiLSTM(FT)	24.82
CNN-BiLSTM	24.83
Transformer encoder	25.32
Bangla-BERT	21.24
m-BERT	19.42
DistilmBERT	21.63
Bangla-T5	22.43
Proposed	15.25

machine learning models such as Logistic Regression (26.52%) and Multinomial Naïve Bayes (25.12%) exhibit relatively higher misclassification rates. These models struggle primarily due to their inability to effectively capture contextual

and sequential relationships, particularly for ambiguous or Weak Factual Claims (WFC).

Deep learning-based models demonstrate improved performance, with misclassification rates generally falling between 19% and 25%. The Deep Neural Network (bi-gram) achieves a 19.14% error rate, showing that bigram-based features allow the model to learn relevant local patterns for class separation. However, sequential models like LSTM (FT) and BiLSTM (FT) have slightly higher misclassification rates (22.63% and 24.82%), which can be attributed to their sensitivity to noisy or ambiguous sequences in the text. These models tend to misclassify WFC instances, as this class often exhibits subtle linguistic nuances that are difficult to capture with sequential architectures alone.

Transformer-based models exhibit superior performance compared to traditional and deep learning counterparts. For instance, m-BERT and DistilBERT achieve lower misclassification rates of 19.42% and 21.63%, respectively, due to their ability to effectively model contextual relationships. However, models such as Bangla-T5 show a slightly higher misclassification rate (22.43%), which can be attributed to the model's generative nature, making it less precise for fine-grained classification tasks.

The Proposed Hybrid Feature-Based Model achieves the lowest misclassification rate of 15.25%, highlighting its robustness in handling ambiguous boundaries between classes. The integration of both bigram features and contextual embeddings enables the model to overcome limitations faced by other architectures, achieving better precision and recall across all categories.

The confusion matrix in Figure 5.2 provides further insights into the nature of the misclassifications made by the proposed model The model misclassifies 54

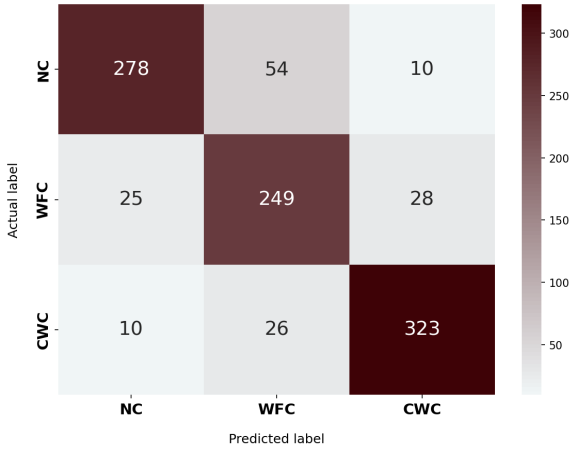


Figure 5.2: Confusion matrix.

instances of NC as WFC. This reflects the challenge of distinguishing between

these two classes, as weak factual claims often share structural and lexical similarities with statements that do not constitute claims. While the model correctly predicts 249 instances of WFC, it misclassifies 25 instances as NC and 28 as CWC. This highlights the nuanced nature of WFC, which often lies between check-worthy claims and no claims, making it challenging for the model to discriminate effectively. The model demonstrates strong performance in the CWC class, with only 10 instances misclassified as NC and 26 instances misclassified as WFC. This indicates that the model effectively identifies claims with clear context but may struggle when contextual boundaries are less distinct. The misclassification analysis reveals that the WFC class presents the most significant challenge, primarily due to its ambiguous nature and the subtle distinctions between CWC and NC. Transformer-based models perform better here because of their ability to analyze word dependencies across the sequence, identifying weak framing cues. Traditional models and LSTM-based approaches struggle with this class as they are less effective in discerning subtle patterns within the text.

Table 5.4 presents a few examples where the model incorrectly predicted the class labels. These errors offer valuable insights into potential challenges and areas for improvement. Many sentences exhibit features that overlap across the classes (CWC, WFC, NC), making it challenging for the model to make precise distinctions. For example, factual statements can simultaneously appear non-claim-like or weakly factual, leading to confusion. In several instances, sentences

Table 5.4: Some instances that our proposed model incorrectly classifies.

Sentence	Predicted	Actual
প্রত্যেকটা টেকনোলজির কিছু সুবিধা-অসুবিধা আছে (Each technology has some advantages and disadvantages)	**WFC**	**NC**
এমন হতে পারে যে অল্প কম্পনে কোনো কোনো অঞ্চলে বেশি ক্ষতি হয়ে যাচ্ছে, আবার বেশি কম্পনে কম ক্ষতি হয়েছে (It may be that low vibrations cause more damage in some areas, while high vibrations cause less damage)	**WFC**	**NC**
সরকার সংবাদমাধ্যম–সংশ্লিষ্ট যতগুলো আইন করেছে বা করার উদ্যোগ নিয়েছে, অন্য কোনো বিষয়ে এত আইন করেনি (As much as the government has enacted or is attempting to enact media-related laws, it has not enacted so many laws on any other issue)	**NC**	**WFC**
সম্প্রতি অল্প বয়সিদের মধ্যে কোলোরেক্টাল ক্যানসারের হার বাড়ছে (Recently, the rate of colorectal cancer among young people is increasing)	**CWC**	**WFC**
প্রায় একই সময়ে চাঁদের দক্ষিণ মেরুতে অবতরণের জন্য লুনা-২৫ নামের মহাকাশযান পাঠিয়েছিল রাশিয়া (Around the same time, Russia sent the Luna-25 spacecraft to land on the Moon's south pole)	**WFC**	**CWC**

that lack explicit factual content or claims (e.g., ”Each technology has some advantages and disadvantages”) were incorrectly classified as WFC instead of NC. This indicates the model’s difficulty in distinguishing between general statements

and weak factual claims. Sentences with complex structures and multiple clauses (e.g., "It may be that low vibrations cause more damage in some areas, while high vibrations cause less damage") were frequently misclassified. These errors suggest that the model struggles with interpreting conditional or nuanced information. The model sometimes fails to grasp the subtle contextual cues necessary to determine a statement's societal or scientific importance, which is essential for correctly identifying CWC. Hypothetical or generalized statements are frequently misclassified as WFC instead of NC, highlighting the need for improved handling of speculative language.

5.5 Execution time and Trainable Parameters

The deep learning and transformer-based models vary in computational intensity. The number of trainable parameters, memory footprint along with training time of the models in our experimental setup is presented in Table 5.5. Deep learning models are computationally lightweight, with significantly fewer parameters and smaller memory requirements, resulting in training times under four minutes. On the other hand, transformer models, while achieving higher performance, are computationally expensive. Models such as Bangla-T5 and mDeBERTa have the highest number of parameters (223M–278M), memory footprints exceeding 800 MB, and training times extending beyond 40 minutes, with Bangla-T5 taking over an hour to train. The proposed hybrid model achieves a balance between efficiency and performance, with a moderate number of parameters (12.7M), a memory footprint of 48.7 MB, and a training time of 18 minutes. This makes it a practical solution for resource-constrained environments while maintaining competitive performance.

5.6 Comparison with Existing Claim Detection Techniques

To the best of our knowledge, no study has been conducted for check-worthy claim detection with a focus on fine-grained claim categorization within the Bengali language. Existing claim detection techniques have primarily been developed and evaluated on datasets from other languages, limiting their applicability to the Bengali linguistic context. To bridge this gap, we have implemented several well-established methods [19, 11, 25, 42] that have demonstrated success in claim detection tasks across other languages. These techniques were systematically

Table 5.5: Computational complexity of deep neural networks and transformer models

Model	Trainable parameters	Memory footprint	Training time
DNN (Bi-gram)	640,611	2.44 MB	2 min 36s
DNN(FastText)	118,403	23.34 MB	3 min 15s
LSTM (FastText)	269,251	23.92 MB	2 min 5s
BiLSTM(FastText)	521,731	24.88 MB	2 min 43s
CNN-BiLSTM	426,549	22.23 MB	3 min 40s
Transformer Models			
Transformer encoder	18,799,747	71.72 MB	3 min 10s
Bangla-BERT	164,398,851	627.13 MB	33 min 18s
m-BERT	177,855,747	678.47 MB	23 min 12s
DistilmBERT	135,326,979	516.23 MB	14 min 4s
Bangla-T5	223,496,451	852.57 MB	1h 21 min 5s
mDeBERTa	278,811,651	1063.58 MB	46 min 52s
Proposed	12,766,467	48.70 MB	18 min 34s

evaluated on our newly developed CheckBanC dataset to ensure a consistent and fair comparison. The performance of these methods is summarized in Table 5.6, using the weighted F1 score as the evaluation metric.

Table 5.6: Comparison of our proposed method with existing techniques on CheckBanC.

Reference	Technique	weighted f_1 score
ClaimBuster [19]	TF-IDF + POS + SVM	0.76
CNC [11]	POS + NER + Logistic Regression	0.74
CURE Claim [25]	T5 Transformer	0.76
OpenFact [42]	mDeBERTa Transformer	0.80
Proposed	Bi-gram + DistilmBERT + DNN	**0.84**

Techniques like ClaimBuster [11], which combines TF-IDF, POS tagging, and an SVM classifier, and CNC [11], which integrates POS tagging, NER, and Logistic Regression, achieve weighted F1 scores of 0.76 and 0.74, respectively. These methods rely on hand-crafted features and simpler classifiers, which limits their ability to capture deeper semantic relationships within the text. The T5 transformer model used in CURE Claim [25] achieves a weighted F1 score of 0.76. While transformer-based architectures effectively capture contextual semantics, the performance is limited by the absence of additional feature representations that could enhance the model's ability to distinguish between claim categories. The approach proposed by Sawinski et al. [42] leverages the mDeBERTa Transformer for claim detection, achieving a higher weighted F1 score of 0.80. The advanced architecture of mDeBERTa enhances contextual understanding, particularly for claim detection. However, the model's reliance solely on contextual embeddings may miss key local patterns present in the data.

Our proposed hybrid feature-based model combines bi-gram features, DistilBERT embeddings, and a Deep Neural Network (DNN) which achieve a weighted F1 score of 0.85, outperforming all previous approaches. The comparative analysis demonstrates that our proposed hybrid feature-based model not only outperforms existing methods but also demonstrates the importance of integrating shallow and deep features for claim detection.

Chapter 6

Conclusion and Future Recommendations

In this study, we addressed the pressing need for automated check-worthy claim detection in resource-constrained Bangla language, to facilitate fact-checking and combat misinformation. Our key contributions include the development and publication of CheckBanC, the first multiclass check-worthy claim detection dataset for Bangla, and the proposal of a hybrid deep learning model that effectively combines bi-gram and DistilBERT-based contextual features. The combination of two types of features ensures that the model is not limited to simple word co-occurrences but also learns richer, contextual representations, making it highly suited for detecting check-worthiness, a task that involves understanding both linguistic structure and the nuanced meaning of the claims. Experimental results demonstrate that our proposed hybrid feature-based model successfully addresses the limitations of previous methods, establishing a new benchmark for claim detection in Bangla. Our method achieved a weighted f1 score of 0.85 outperforming other baselines and existing state-of-the-art methods on our CheckBanC dataset. This highlights the importance of combining diverse features and advanced neural architectures to achieve robust and accurate fine-grained claim categorization.

6.1 Limitations

This study addresses a critical gap in automated fact-checking for resource-constrained languages by introducing the first approach to multiclass claim detection for Bangla. While our study makes significant strides in advancing claim detection for Bangla, several limitations must be acknowledged to provide a balanced perspective on its contributions. These limitations stem from the challenges inherent in working with resource-constrained languages, the specific scope

of the dataset, and the computational methods employed. Recognizing these constraints not only helps contextualize the findings but also highlights opportunities for future research to refine and extend the work. Below, we outline the key limitations of our study: (i) The primary limitation lies in differentiating between WFC and other categories. Weakly framed claims exhibit overlapping characteristics with both NC and CWC. This ambiguity is a fundamental limitation of the dataset itself, as linguistic boundaries between these categories are inherently blurred. (ii) Although bigram and contextual embeddings improve performance, rare or unseen patterns in the data can still challenge model generalization. (iii)The integration of bigram features with contextual embeddings increases the computational complexity of the model. While effective, this may pose scalability challenges for larger datasets or real-time applications.

6.2 Future Recommendations

For future research direction, we aim to expand the corpus from other potential sources like public speeches, and social media. The check-worthiness of a claim is sometimes dominated by its surrounding context sentences. Features extracted from the contexts of the candidate claim sentence can be a very good way to provide a better understanding of the claim to model. This will allow the model to delve deeper into the nuances of a claim to determine its level of checkworthiness. The choice of feature also has scope to incorporate complex linguistic features like the subject entity of claims. This work also plans to implement a ranking mechanism of claims within a document based on their checkworthiness. The ranking of claims will expedite the fact-checking process. Leveraging the capabilities of large language models (LLMs) is a very potential research field yet to be explored in the claim detection domain. We also plan to investigate the performance of various embedding models on our dataset. Furthermore, we plan to conduct an ablation study as part of future research to refine feature selection and improve model effectiveness Another important aspect to address is integrating explainable AI (XAI) to improve the classification's interpretability. Finally, we plan to deploy our model on a live server to evaluate its performance in real-world scenarios.

6.3 List of Publications

1. **R. Rahman**, R. Karim, M.S. Arefin, P.K. Dhar, G. Hossain and T. Shimamura, "Facilitating automated fact-checking: a machine learning based weighted ensemble technique for claim detection", *Discover Applied Sciences*, Springer Nature, vol. 7, no. 73, 2025.

Bibliography

[1] T. Khan, A. Michalas, and A. Akhunzada, "Fake news outbreak 2021: Can we stop the viral spread?," *Journal of Network and Computer Applications*, p. 103112, 2021.

[2] S. Bradshaw and P. N. Howard, "The global disinformation order: 2019 global inventory of organised social media manipulation," 2019.

[3] N. Hassan, C. Li, and M. Tremayne, "Detecting check-worthy factual claims in presidential debates," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, (New York, NY, USA), p. 1835–1838, Association for Computing Machinery, 2015.

[4] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[5] M. M. Haque, M. Yousuf, A. S. Alam, P. Saha, S. I. Ahmed, and N. Hassan, "Combating misinformation in bangladesh: roles and responsibilities as perceived by journalists, fact-checkers, and users," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW2, pp. 1–32, 2020.

[6] Z. Guo, M. Schlichtkrull, and A. Vlachos, "A survey on automated fact-checking," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 178–206, 2022.

[7] D. Graves, "Understanding the promise and limits of automated fact-checking," *Reuters Institute for the Study of Journalism*, 2018.

[8] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "Fever: a large-scale dataset for fact extraction and verification," *arXiv preprint arXiv:1803.05355*, 2018.

[9] M. Soprano, K. Roitero, D. La Barbera, D. Ceolin, D. Spina, G. Demartini, and S. Mizzaro, "Cognitive biases in fact-checking and their countermeasures: A review," *Information Processing and Management*, vol. 61, no. 3, p. 103672, 2024.

[10] D. La Barbera, E. Maddalena, M. Soprano, K. Roitero, G. Demartini, D. Ceolin, D. Spina, and S. Mizzaro, "Crowdsourced fact-checking: Does it actually work?," *Information Processing and Management*, vol. 61, no. 5, p. 103792, 2024.

[11] L. Konstantinovskiy, O. Price, M. Babakar, and A. Zubiaga, "Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection," *Digital Threats*, vol. 2, apr 2021.

[12] X. Zeng, A. S. Abumansour, and A. Zubiaga, "Automated fact-checking: A survey," *Language and Linguistics Compass*, vol. 15, no. 10, p. e12438, 2021.

[13] A. Das, H. Liu, V. Kovatchev, and M. Lease, "The state of human-centered nlp technology for fact-checking," *Information Processing and Management*, vol. 60, no. 2, p. 103219, 2023.

[14] N. Micallef, V. Armacost, N. Memon, and S. Patil, "True or false: Studying the work practices of professional fact-checkers," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW1, pp. 1–44, 2022.

[15] M. Chowdhury, A. Hossain, and M. J. Rime, "News literacy in bangladesh," *Management and Resources Development Initiative (MRDI)*, 2023.

[16] Y. Cao, H. Li, P. Luo, and J. Yao, "Towards automatic numerical cross-checking: Extracting formulas from text," in *Proceedings of the 2018 World Wide Web Conference*, pp. 1795–1804, 2018.

[17] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or fiction: Verifying scientific claims," *arXiv preprint arXiv:2004.14974*, 2020.

[18] W. Mansour, T. Elsayed, and A. Al-Ali, "This is not new! spotting previously-verified claims over twitter," *Information Processing and Management*, vol. 60, no. 4, p. 103414, 2023.

[19] N. Hassan, F. Arslan, C. Li, and M. Tremayne, "Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster," KDD '17, (New York, NY, USA), p. 1803–1812, Association for Computing Machinery, 2017.

[20] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, *et al.*, "Claimbuster: The first-ever end-to-end fact-checking system," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1945–1948, 2017.

[21] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint arXiv:1705.02364*, 2017.

[22] K. Meng, D. Jimenez, J. D. Devasier, S. S. Naraparaju, F. Arslan, D. Obembe, and C. Li, "Gradient-based adversarial training on transformer networks for detecting check-worthy factual claims," *ACM Trans. Intell. Syst. Technol.*, Aug. 2024.

[23] A. Patwari, D. Goldwasser, and S. Bagchi, "Tathya: A multi-classifier system for detecting check-worthy statements in political debates," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 2259–2262, 2017.

[24] Y. Bai, A. Colas, and D. Z. Wang, "Mythqa: Query-based large-scale check-worthy claim detection through multi-answer open-domain question answering," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3017–3026, 2023.

[25] S. D. Gollapalli, M. Du, and S.-K. Ng, "Identifying checkworthy cure claims on twitter," in *Proceedings of the ACM Web Conference 2023*, WWW '23, (New York, NY, USA), p. 4015–4019, Association for Computing Machinery, 2023.

[26] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[27] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, R. Míguez, T. Caselli, M. Kutlu, W. Zaghouani, C. Li, S. Shaar, *et al.*, "Overview of the clef-2022 checkthat! lab task 1 on identifying relevant claims in tweets," in *CLEF 2022: Conference and Labs of the Evaluation Forum*, pp. 368–392, CEUR Workshop Proceedings, sep 2022.

[28] X. Wei, M. R. U. Hoque, J. Wu, and J. Li, "Claimdistiller: Scientific claim extraction with supervised contrastive learning," in *Proceedings of Joint Workshop of the 4th Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE2023) and the 3rd AI + Informetrics (All2023)*, pp. 65–77, CEUR Workshop Proceedings, jun 2023.

[29] T. Alhindi, B. McManus, and S. Muresan, "What to fact-check: Guiding checkworthy information detection in news articles through argumentative discourse structure," in *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 380–391, 2021.

[30] P. Smeros, C. Castillo, and K. Aberer, "Sciclops: Detecting and contextualizing scientific claims for assisting manual fact-checking," CIKM '21, (New York, NY, USA), p. 1692–1702, Association for Computing Machinery, 2021.

[31] V. Woloszyn, J. Kobti, and V. Schmitt, "Towards automatic green claim detection," FIRE '21, (New York, NY, USA), p. 28–34, Association for Computing Machinery, 2022.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[33] D. Q. Nguyen, T. Vu, and A. Tuan Nguyen, "BERTweet: A pre-trained language model for English tweets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Online), pp. 9–14, Association for Computational Linguistics, Oct. 2020.

[34] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proceedings of the 27th International Conference on Computational Linguistics*, (Santa Fe, New Mexico, USA), pp. 1638–1649, Association for Computational Linguistics, Aug. 2018.

[35] D. Saha, M. Sinha, and T. Dasgupta, "EnClaim: A style augmented transformer architecture for environmental claim detection," in *Proceedings of the 1st Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2024)*, (Bangkok, Thailand), pp. 123–132, Association for Computational Linguistics, Aug. 2024.

[36] P. Nakov, A. Barrón-Cedeño, T. Elsayed, R. Suwaileh, L. Màrquez, W. Zaghouani, P. Atanasova, S. Kyuchukov, and G. Da San Martino, "Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 9th*

International Conference of the CLEF Association, CLEF 2018, Proceedings 9, (Avignon, France), pp. 372–387, Springer International Publishing, 2018.

[37] P. Atanasova, A. Barron-Cedeno, T. Elsayed, R. Suwaileh, W. Zaghouani, S. Kyuchukov, G. D. S. Martino, and P. Nakov, "Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims. task 1: Check-worthiness," *arXiv preprint arXiv:1808.05542*, 2018.

[38] P. Atanasova, P. Nakov, G. Karadzhov, M. Mohtarami, and G. Da San Martino, "Overview of the clef-2019 checkthat! lab: Automatic identification and verification of claims. task 1: Check-worthiness.," *CLEF (Working Notes)*, vol. 2380, 2019.

[39] A. Barrón-Cedeno, T. Elsayed, P. Nakov, G. Da San Martino, M. Hasanain, R. Suwaileh, F. Haouari, N. Babulkov, B. Hamdan, A. Nikolov, *et al.*, "Overview of checkthat! 2020: Automatic identification and verification of claims in social media," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings 11*, pp. 215–236, Springer, 2020.

[40] A. B. Eyuboglu, B. Altun, M. B. Arslan, E. Sonmezer, and M. Kutlu, "Fight against misinformation on social media: Detecting attention-worthy and harmful tweets and verifiable and check-worthy claims," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 161–173, Springer Nature Switzerland Cham, 2023.

[41] M. Hasanain, R. Suwaileh, S. Weering, C. Li, T. Caselli, W. Zaghouani, A. Barrón-Cedeño, P. Nakov, and F. Alam, "Overview of the clef-2024 checkthat! lab task 1 on check-worthiness estimation of multigenre content," *Working Notes of CLEF*, 2024.

[42] M. Sawiński, K. Węcel, and E. Księżniak, "Openfact at checkthat! 2024: Cross-lingual transfer learning for check-worthiness detection," in *CLEF 2024: Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, sep 2024.

[43] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," in *International Conference on Learning Representations*, 2021.

[44] P. He, J. Gao, and W. Chen, "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing," 2021.

[45] Y. S. Kartal and M. Kutlu, "TrClaim-19: The first collection for Turkish check-worthy claim detection with annotator rationales," in *Proceedings of the 24th Conference on Computational Natural Language Learning*, (Online), pp. 386–395, Association for Computational Linguistics, Nov. 2020.

[46] B. Berendt, P. Burger, R. Hautekiet, J. Jagers, A. Pleijter, and P. Van Aelst, "Factrank: Developing automated claim detection for dutch-language fact-checkers," *Online Social Networks and Media*, vol. 22, p. 100113, 2021.

[47] I. Jaradat, P. Gencheva, A. Barrón-Cedeño, L. Màrquez, and P. Nakov, "ClaimRank: Detecting check-worthy claims in Arabic and English," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, (New Orleans, Louisiana), pp. 26–30, Association for Computational Linguistics (ACL), June 2018.

[48] P. Gencheva, P. Nakov, L. Màrquez, A. Barrón-Cedeño, and I. Koychev, "A context-aware approach for detecting worth-checking claims in political debates," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, (Varna, Bulgaria), pp. 267–276, Sept. 2017.

[49] T. Achakulvisut, C. Bhagavatula, D. Acuna, and K. Kording, "Claim extraction in biomedical publications using deep discourse model and transfer learning," *arXiv preprint arXiv:1907.00962*, 2019.

[50] A. Fergadis, D. Pappas, A. Karamolegkou, and H. Papageorgiou, "Argumentation mining in scientific literature for sustainable development," in *Proceedings of the 8th Workshop on Argument Mining*, (Punta Cana, Dominican Republic), Association for Computational Linguistics, Nov. 2021.

[51] M. M. Haque, M. Yousuf, Z. Arman, M. M. U. Rony, A. S. Alam, K. M. Hasan, M. K. Islam, and N. Hassan, "Fact-checking initiatives in bangladesh, india, and nepal: a study of user engagement and challenges," *arXiv preprint arXiv:1811.01806*, 2018.

[52] F. Arslan, N. Hassan, C. Li, and M. Tremayne, "A benchmark dataset of check-worthy factual claims," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, pp. 821–829, 2020.

[53] E. M. Bender and B. Friedman, "Data statements for natural language processing: Toward mitigating system bias and enabling better science," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 587–604, 2018.

[54] R. Panchendrarajan and A. Zubiaga, "Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research," *Natural Language Processing Journal*, vol. 7, p. 100066, 2024.

[55] J. L. Fleiss, "Measuring nominal scale agreement among many raters.," *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[56] T. Takenobu, "Text categorization based on weighted inverse document frequency," *Information Processing Society of Japan, SIGNL*, vol. 94, no. 100, pp. 33–40, 1994.

[57] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[58] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[60] Q. Fournier, G. M. Caron, and D. Aloise, "A practical survey on faster and lighter transformers," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–40, 2023.

[61] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[62] S. Sarker, "Banglabert: Bengali mask language model for bengali language understanding," 2020.

[63] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[64] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mT5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 483–498, Association for Computational Linguistics, June 2021.

[65] A. Bhattacharjee, T. Hasan, W. U. Ahmad, and R. Shahriyar, "BanglaNLG and BanglaT5: Benchmarks and resources for evaluating low-resource natural language generation in Bangla," in *Findings of the Association for Computational Linguistics: EACL 2023*, (Dubrovnik, Croatia), pp. 726–735, Association for Computational Linguistics, May 2023.

[66] J. K. Tripathy, S. C. Sethuraman, M. V. Cruz, A. Namburu, M. P., N. K. R., S. I. S, and V. Vijayakumar, "Comprehensive analysis of embeddings and pre-training in nlp," *Computer Science Review*, vol. 42, p. 100433, 2021.

[67] M. Kubat, S. Matwin, *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Icml*, vol. 97, p. 179, Citeseer, 1997.

[68] V. L. Parsons, "Stratified sampling," *Wiley StatsRef: Statistics Reference Online*, pp. 1–11, 2014.