Checkworthiness and Claim Span Identification from Tweets

A thesis submitted in partial fulfilment of the requirement for the degree of
Master of Engineering in Computer Science & Engineering
in the **Department of Computer Science & Engineering,**
Jadavpur University

By

Prantik Guha
Registration No: 154135 of 2020-2021
Examination Roll No.: M4CSE22011

Under the Guidance of

Dr. Dipankar Das
Department of Computer Science & Engineering
Jadavpur University, Kolkata-700032

2022

FACULTY OF ENGINEERING AND TECHNOLOGY JADAVPUR UNIVERSITY

To Whom It May Concern

I hereby recommend that the thesis titled "Checkworthiness and Claim Span Identification from Tweets" has been carried out by Prantik Guha (Reg. No.: 154135 of 2020-2021, Exam Roll: M4CSE22011), under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of Master of Engineering in Computer Science & Engineering in Department of Computer Science & Engineering, Jadavpur University.

Prof. Dipankar Das,
Assistant Professor,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Forwarded By:

Prof. Anupam Sinha,
Head,
Department of Computer Science and Engineering,
Jadavpur University.

Prof. Chandan Majumdar,
DEAN,
Faculty of Engineering and Technology,
Jadavpur University.

FACULTY OF ENGINEERING AND TECHNOLOGY JADAVPUR UNIVERSITY

Certificate of Approval

This is to certify that the thesis entitled "Checkworthiness and Claim Span Identification from Tweets" is a bonafide record of work carried out by Prantik Guha in fulfilment of the requirements for the award of the degree of Master of Engineering in Computer Science & Engineering in the Department of Computer Science & Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Signature of Examiner 1
Date:

Signature of Examiner 2
Date:

FACULTY OF ENGINEERING AND TECHNOLOGY JADAVPUR UNIVERSITY

Declaration of Originality Compliance of Academic Ethics

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as a part of his Master of Engineering in Computer Science & Engineering studies. All information in this document has been obtained and present in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name : Prantik Guha
Registration No : 154135 of 2020-2021
Exam Roll No. : M4CSE22011
Thesis Title : Checkworthiness and Claim Span Identification from Tweets

Signature with Date

Acknowledgement

The writing of the thesis as well as the related work has been a long journey with input from many individuals, right from the first day till the development of the final project. With my most sincere and gratitude, I would like to thank Dr. Dipankar Das, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required inputs and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestion and numerous discussions have always inspired new ways of thinking. I feel deeply honoured that I got this opportunity to work under him. I would like to thank all the faculty members of the Department of Computer Science and Engineering of Jadavpur University for their continuous support. I would also like to thank Mr. Subhabrata Dutta, Mr. Rudra Dhar, Mr. Atanu Mandal and Mr. Palash Nandi for their valuable suggestions and discussions to accomplish the work. Finally, I would like to thank all my family members and friends for their unconditional support.

Name : Prantik Guha
Registration No : 154135 of 2020-2021

Abstract

In Natural Language Processing (NLP), identification of check-worthiness and detection of span from any news article is a crucial yet difficult topic. Automatic fake news identification is a practical NLP problem that is helpful to all online content producers, given the vast amount of Web material. The rapid expansion of social networking platforms has resulted in a tremendous increase in information accessibility while also hastening the dissemination of false information. Identification of checkworthiness from any news article is very much required task. Not only that which part of the article is containing the main claim that is detection of claim span is a very challenging and interesting task. India has no such automatic or AI system to identify fake news, so it is critical to develop these systems in Indian languages. The first challenge is to find check-worthy words among the massive amounts of data created daily from news and other sources on the internet. Only completing this step completes half of the fact-checking task. Though a lot of training data is available for English, no such dataset is available in the Indian vernacular languages. So we have created our dataset from tweeter which contains Bengali, English, Hindi & Codemix data. Using this data we have built our systems to identify the checkworthiness and to detect the claim span from tweet.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

In Natural Language Processing(NLP), identification of checkworthiness and detection of span from any news article is a crucial yet difficult topic. The use of social media for news consumption has two sides. On the one hand, consumers seek out and consume news via social media because of its low cost, easy access, and rapid transmission of information. On the other side, it facilitates the widespread dissemination of "fake news" which is low-quality news that contains purposefully misleading material.

The widespread dissemination of fake news has the potential to have tremendously detrimental consequences for both individuals and society. As a result, detecting false news on social media has recently become an emerging study topic that is gaining a lot of interest. Fake news identification on social media has unique characteristics and obstacles that render classic news media detection algorithms ineffective or inapplicable. The increasing growth of false news, as well as the damage it causes to democracy, justice, and public trust, has boosted demand for fake news detection and intervention.

The fast rise of social networking platforms has increased the dissemination of bogus news while simultaneously vastly increasing information accessible. As a result, the impact of fake news has grown, sometimes spilling over into the offline world and endangering public safety. Automatic fake news identification is a practical NLP problem helpful to all online content producers, given the vast amount of Web material, in order to decrease human time and effort in detecting and preventing the spread of fake news.

1.1 NATURAL LANGUAGE PROCESSING

Natural Language Processing is a branch of linguistics, computer science, information engineering, and artificial intelligence concerned with how computers interact

with human languages, particularly how to teach computers to process and analyse enormous amounts of natural language data. Vectorization is a process that converts words (text information) into digits in order to extract text properties (features) and then utilise machine learning (NLP) techniques to retrieve those features.

Some applications of NLP are:

- Machine translation
- Grammar and spell checking
- Text classification
- Named-entity recognition (NER)
- Summarization
- Text generation

Some of the NLP techniques are:

- Bag of words
- N gram
- Term Frequency Inverse Document Frequency (TFIDF)
- Tokenization
- Part Of Speech (POS) Tag

Some of the NLP tools are:

- Natural Language Toolkit (NLTK)
- SentiWordNet
- TextBlob
- Stanford CoreNLP

1.1.1 TRADITIONAL PROBLEMS

In our daily lives, we see AI technology with NLP in the form of Alexa and Siri, email and text predictive text, and customer care chatbots. Using machine learning algorithms and natural language processing, they all comprehend, "understand," and react to human language, both spoken and written (NLP). While natural language processing (NLP) and its sister discipline, natural language understanding (NLU), are constantly improving their ability to calculate letters and phrases, human language is enormously complex, fluid, and inconsistent, offering significant challenges that NLP

has yet to conquer. The majority of the difficulties come from data complexity, as well as features like sparsity, variety, dimensionality, and the dynamic properties of the datasets. NLP is still a young technology, therefore there is a lot of room for engineers and businesses to tackle the numerous unsolved problems that come with deploying NLP systems. Lets take a look at some of those challenges in more detail below.

- **Contextual words and phrases and homonyms** : The same words and phrases can have distinct meanings depending on the context of a statement, and many words particularly in English have identical pronunciation but completely different meanings. For example:
  - I **ran** to the store because we **ran** out of milk.
  - Can I **run** something past you real quick?
  - The house is looking really **run** down.

  Because we read the context of the statement and grasp all of the numerous definitions, they are simple for humans to comprehend. Even if NLP language models have learnt all of the meanings, distinguishing between them in context can be difficult.
- **Synonyms** : Because we use many different words to communicate the same idea, synonyms might present challenges comparable to contextual understanding. Furthermore, some of these terms may have identical meanings, while others may have varying degrees of complexity (little, little, tiny, minute), and various persons employ synonyms to signify somewhat different meanings within their particular vocabulary. As a result, it's critical to include all of a word's alternative meanings and synonyms while developing NLP systems. Although text analysis algorithms are not perfect, the more relevant training data they receive, the better they will be able to understand synonyms.
- **Ambiguity** : Ambiguity in NLP refers to sentences and phrases that potentially have two or more possible interpretations.
  - **Lexical ambiguity**: a word that could be used as a verb, noun, or adjective.
  - **Semantic ambiguity**: the interpretation of a sentence in context. For example: I saw the boy on the beach with my binoculars. This could mean that I saw a boy through my binoculars or the boy had my binoculars with him
  - **Syntactic ambiguity**: In the sentence above, this is what creates the confusion of meaning. The phrase with my binoculars could modify the verb, "saw," or the noun, "boy."

Even for humans, interpreting this line without the context of the surrounding text is challenging. One NLP solution that can assist tackle the problem in some ways is POS (part of speech) tagging.

- **Errors in text and speech** : Text analysis might be hampered by misspelt or misused terms. Although autocorrect and grammar checkers can handle common errors, they don't always understand the writer's purpose. Mispronunciations, various accents, stutters, and other nuances in spoken language can be difficult for a machine to comprehend. These difficulties can be mitigated as language databases develop and smart assistants are taught by their unique users.
- **Language Differences** : Although English is spoken by the majority of people, if you want to reach a worldwide and/or diverse audience, you'll need to support multiple languages. Not only do different languages have vastly different vocabularies, but they also have a wide range of phrasing, inflections, and cultural standards. This challenge can be solved by employing "universal" models that allow you to transfer at least some of what you've learnt to other languages. However, each additional language will necessitate time spent updating your NLP system.

The rapid expansion of social networking platforms has resulted in a tremendous increase in information accessibility while also hastening the dissemination of false information. As a result, fake news's influence has expanded. So identification of checkworthiness from any news article is very much required task. Not only that which part of the article is containing the main claim that is detection of claim span is a very challenging and interesting task in today's world. Let's discuss about fake news and the challenges first.

1.1.2 FAKE NEWS

A well-functioning society need news. It keeps people informed about critical topics and allows them to create their own opinions on them. As a result, the information that people receive can have a huge impact. While this places a greater emphasis on media outlets to create trust and credibility, these qualities are only one part of the equation when it comes to distinguishing real news from false news. Fake news, unfortunately, has no antidote. There is no vaccination available. There's only one requirement: a commitment to seeing the news, or what passes for news, for what it is. Such a commitment is enormous, and only a few people have the time or means to devote to it.

In Natural Language Processing, detecting fake news is a crucial yet difficult subject (NLP). The rapid growth of social networking platforms has resulted in a massive boost

in information accessibility while simultaneously hastening the spread of bogus news. As a result, the impact of fake news has grown, sometimes spilling over into the offline world and endangering public safety. Automatic fake news identification is a practical NLP problem helpful to all online content producers, given the vast amount of Web material, in order to decrease human time and effort in detecting and preventing the spread of fake news.

1.1.3 CHALLENGES

The proliferation of fake news on the internet has far-reaching consequences for people's lives offline. The demand on content sharing platforms to act and reduce the spread of fake news is growing, but intervention is met with charges of unfair censorship. The conflict between fair moderation and censorship highlights two related issues that arise when flagging online content as fake or legitimate: first, what types of content should be flagged, and second, is it practical and theoretically possible to gather and label instances of such content in an unbiased manner? Below are few points about the fake news detection:

- The main challenge while detecting a fake news is to validate that news using some authentic source.
- Discriminating between truthful and fake information that are both machine-generated. Neural language models are increasingly used to produce any text. Applications such as auto-completion,text modication, question answering, text simplication, summarization, and others, are growing in popularity. Those might be used both for producing legitimate and malicious text, resulting in "fake" and "real" text sources that are almost identical.
- Detecting an attacker that uses text from a legitimate human source, but automatically corrupts it to alter its meaning by making only subtle, relatively small changes. Unfortunately, powerful language models designed for text generation, can also easily be used to corrupt existing text.

The first challenge is to find check-worthy words among the massive amounts of data created daily from news and other sources on the internet. Only completing this step completes half of the fact-checking task. Almost everyone in India, including the government, recognises the country's "fake news" problem. Though such systems, such as Claimbuster, are made in Western countries, India has no such automatic or AI system. It is critical to develop these systems in Indian languages. To make Machine Learning systems to identify check-worthy sentences, a lot of training data is required. Though a lot of training data is available for English, no such dataset is available in

the Indian vernacular languages. So, getting labelled training data is a big challenge for this job.

After finding the check-worthy sentences, next step is to find the span from it. That means the part of the sentence which is containing the claim.

1.1.4 OBJECTIVE

In our current research, we are mainly focusing on checkworthiness identification and span detection from tweet data. First lets discuss on the problem statement briefly.

- **Checkworthiness Identification** is the process of checking whether a tweet or any news article is having any claim in it or not. This kind of binary classification is called checkworthiness identification.

  For example, "Mamta Banerjee all set to contest the By-elections from her traditional Bhawanipore seat. Shobhandev Chatterjee to submit his resignation anytime soon . #BengalElection2021 #Bengal #MamataBanerjee", this tweet is containing claim in it.

  And "Read your thread @rohansmitra. Felt like reading about when the Ostrich amp; Kalidas got together. And the rest is....well history. 0 in #BengalElection2021! https://t.co/5rbJU7tC39", this tweet is not containing claim in it. So this kind of binary identification is called Checkworthiness Identification.

- **Claim Span Detection** is the process of detecting the original claim from the tweet or any news article. Suppose any tweet is having claim in it. So how to find the exact claim from the main tweet that is the problem defination for Claim Span detection.

  Example: for the tweet "Mamta Banerjee all set to contest the By-elections from her traditional Bhawanipore seat. Shobhandev Chatterjee to submit his resignation anytime soon . #BengalElection2021 #Bengal #MamataBanerjee", it is having multiple claims in it. Those are i) "Mamta Banerjee all set to contest the By-elections from her traditional Bhawanipore seat", ii) "Shobhandev Chatterjee to submit his resignation anytime soon".

  But for the tweet "Not The First Time Gujarat Fails To Win #Begal #KhelaHobe @MamataOfficial @MahuaMoitra @MahuaMoitraFans #ElectionResult #BengalElection2021 #KhelaHoyeGeyche #KhelaCholcheCholbe", it is not having any claim in it. So no claim will be detected for this.

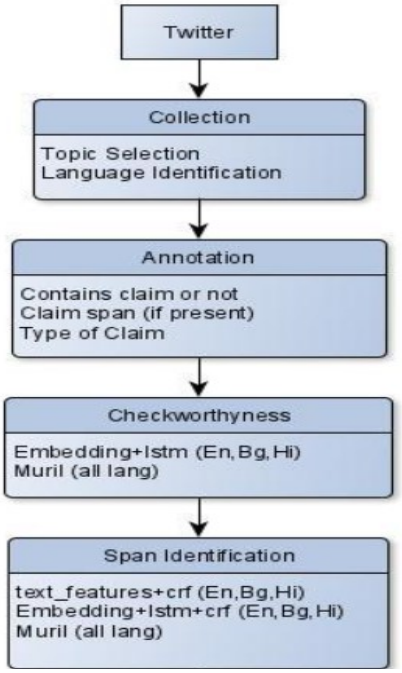

Figure 1.1: Proposed workflow

1.1.5 CONTRIBUTION

In this thesis, we will describe our work on data collection, preparation of annotation guidelines, data annotation, development of claim identification models, and development of span detection models from tweet. Following diagram describes the proposed workflow 1.1.

Our contributions to this thesis are as follows:

- **Data crawler from twitter** : We have used twitter's v2 api to crawl the twitter data for our entire research purpose. To develop a multilingual perspective, we have focused primarily on tweets written in English, Bengali, Hindi, and codemixed (within the three mentioned languages). We have developed one python script to crawl the data from twitter based on the hashtag search within a mentioned timeframe.
- **Language Identifier** : We prepared a language Identifier based on UTF encoding. This is specifically made for tweets, as it cleans a given tweet and then identifies the language. It classifies the language of the tweet as English, Bengali, Hindi, Codemixed, and others (containing languages other than the mentioned). Among the codemixed tweets, it can also classify between English-

Bengali, Bengali-Hindi, and English-Hindi. We have also done some comparative study between our developed language identifier and Meta's fasttext.

- **User Interface for data annotation** : We have created one user interface using python to do the data annotation faster and in better way.
- **Models for checkworthiness detection** : We have developed two models one based on LSTM and another one based on MURIL to detect the checkworthiness. We have also done comparative study and error analysis between two models.
- **Models for span detection from tweets** : We have developed three models one based on text features + crf and second one using Golve word embeddings + LSTM + crf (please find the system architecture 5.2)and another one based on MURIL to detect the span from the tweets. We have also done comparative study and error analysis between the models.

1.1.6 THESIS OUTLINE

In the first chapter, we have done a brief introduction about NLP and traditional problems in NLP. Then we have discussed about Fake News and why it is important to be distinguished. After we have discussed about the various challenges in checkworthiness identification and span detection. Then we have discussed about our proposed solution to overcome that problems.

In the second chapter, we have mentioned various already proposed methodology on checkworthiness identification and span detection.

In the third chapter, we have elaborately discussed about the data that we have used in our thesis. We have mentioned about how we have crawled the data from twitter and how we have created our data annotations required for our thesis.

In the fourth chapter, we have discussed on checkworthiness identification. Our approach to identify the checkworthy tweets, we have discussed in this chapter. Complete system description, model comparison, observations and error analysis, we have discussed.

In the fifth chapter, We discussed claim span detection from tweets. We covered our method for detecting the span from tweets in this chapter. We went over the entire system description, model comparison, observations, and error analysis.

Chapter 2

LITERATURE SURVEY

Nowadays, Machine Learning is Artificial Intelligence's workhorse. ML is used in a variety of applications, including stock market forecasting, object detection, computer vision, and the arduous task of Natural language generation. However, with the exponential growth of data and processing capacity, machine learning is no longer sufficient. Neural Networks' much more complicated functionalities are ideal for the job. Natural language processing is a branch of AI concerned with teaching machines to understand natural languages such as English, Bengali, and others and to make inferences from them.

Natural language processing (NLP) is a subject of computer sciencespecifically, a branch of artificial intelligence (AI)concerning the ability of computers to understand text and spoken words in the same manner that humans can. Computational linguisticsrule-based human language modelingis combined with statistical, machine learning, and deep learning models in NLP. These technologies, when used together, allow computers to process human language in the form of text or speech data and 'understand' its full meaning, including the speaker's or writer's intent and sentiment. NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidlyeven in real time. Theres a good chance youve interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

2.1 CHECKWORTHINESS IDENTIFICATION

In this era of fake news and misinformation, fake news detection is becoming more and more important. But to detect fake news, we need to identify check-worthy factual sentences. In this thesis we address this problem. We take up various approaches and make various systems for identifying check-worthiness of a sentence and claim span detection. In the next subsection we note research works attempted by other people in this domain, and some Techniques and Tools that are required in this task.

Authenticity and intent, are the two most important aspects of the definition of Fake News. For starters, fake news involves deceptive material that can be proven to be false. Second, fake news is manufactured with the purpose of deceiving consumers. In recent studies, this definition has been extensively used. Broader definitions of fake news focus on the news content's authenticity or intent. Even while satire is frequently entertainment-oriented and reveals its own deception to the consumers, some papers classify satire news as fake news because the contents are incorrect.

Language modelling is a method for calculating the probability of any word sequence. Language modelling is utilised in a wide range of applications, including Speech Recognition and Spam Filtering. In reality, the development of many state-of-the-art Natural Language Processing models is driven by language modelling. The contiguous sequence of n items from a given sample of text or speech is known as an N-gram. Depending on the application, the elements can be letters, words, or base pairs. N-grams are usually extracted from a text or speech corpus (A long text dataset). N-grams are utilized for a wide range of tasks. When creating a language model, for example, n-grams are utilized to create not only unigram models but also bigram and trigram models. Google and Microsoft have created web-scale n-gram models that can be used for things like spelling correction, word breaking, and text summarizing. Brown et al., 1992 discussed class-based n-gram models of natural language.

Fake news detection on social media presents unique characteristics and challenges that make existing detection algorithms from traditional news media ineffective or not applicable. Fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content. In this survey Shu et al., 2017, they present a comprehensive review of fake news characterizations on psychology and social theories, existing algorithms from a data mining perspective, evaluation metrics and datasets.

Digital media have led to the emergence of fake news that misinform. This phenomenon has reached huge proportions and is becoming a serious problem. Different approaches have been proposed to automatically detect fake news, based on analyzing their content, source or dispersion. The objective of this work Pérez-Santiago et al., 2022 is to explore whether the written style of news can be used for this task. The main objective of this

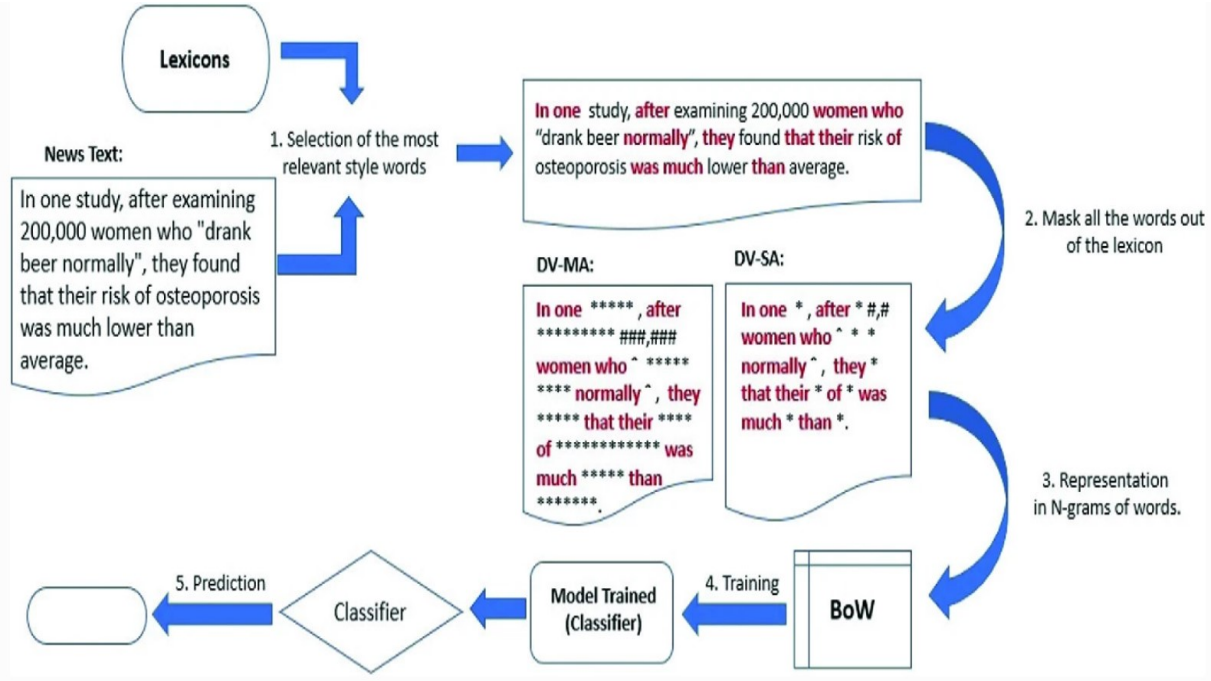

Figure 2.1: Proposed workflow from Pérez-Santiago et al., 2022

task is to use the written style of the news to conclude whether it is fake or not. Please refer to the below figure 2.1 for detail overview of the work.

Today's social networks create huge quantities of data on a regular basis, providing a useful starting point for detecting rumours as soon as they begin to spread. However, given the enormous volume of high-velocity streaming data released by social networks, rumour detection faces tight latency limitations that can't be satisfied by current algorithms. In this paper Nguyen et al., 2022, we argue for best-effort rumour detection that detects most rumours quickly rather than all rumours with a high delay. We combine techniques for efficient, graph-based matching of rumour patterns with effective load shedding to minimise the loss in accuracy.

The increasing growth of fake news, as well as the damage it causes to democracy, justice, and public trust, has boosted demand for false news detection and intervention. This study examines and assesses strategies for detecting fake news from four perspectives: (1) the inaccurate information it contains, (2) its writing style, (3) its dissemination patterns, and (4) the source's legitimacy. Based on the review, the survey also suggests some interesting research topics. To stimulate multidisciplinary study on false news, they identify and discuss related core theories across several fields. They anticipate that (Zhou & Zafarani, 2020) by conducting this poll, professionals in computer and information sciences, social sciences, political science, and journalism would be able to collaborate on fake news research, resulting in more efficient and,

more crucially, explainable fake news detection.

Fake news is a growing global concern as it damages democracy and the public's trust in the media. In this study Mishra et al., 2022, they have investigated the identification of false news using lexical features collected from three datasets and various machine learning models. They find that style-based fake news detection algorithms consistently outperform the other models by overcoming the disadvantage of recognizing fake news before it spreads. Logistic regression is more efficient for classification than the others, with an average performance of 90%.

2.2 SPAN DETECTION

The method of detecting fake news gives birth to span detection. Some of the earlier workers judged a news article's authenticity only on the basis of its source. As you can expect, this method is not scientific. With the growth of artificial intelligence and machine learning, span detection has recently piqued the interest of researchers, paving the way for it to become a stand-alone research topic.

Part-of-speech tagging (POS tagging), also known as grammatical tagging, is the technique of marking up a word in a text as relating to a specific part of speech based on its definition and context in computational linguistics and natural language processing. The wording of POS tags is quite important. In fact, there are a variety of pos labelling methods in any language. In English, the most commonly accepted POS tagging system is the one made in Taylor et al., 2003.

There are many tools made for POS tagging. The Stanford CoreNLP Toolkit made by Manning et al., 2014 is a very useful java based tool not only for POS tagging but for many other tasks, including syntactic parsing. This is the most reliable POS tagger in English. A POS Tagger comes with NLTK (Natural Language Toolkit). This is fast and very easy to use. It uses a default corpus for English, but other corpus can be imported to use it for other languages. The reliability depends a lot on the corpus used. So, for English, the NLTK POS Tagger is very much reliable. But it is not reliable enough for Indian languages like Bengali or Hindi.

Span detection is a rhetorical technique designed to serve a specific topic, which is often used purposefully in news article to achieve our intended purpose. It is significant to be clear where and what propaganda techniques are used in the news for people to understand its theme efficiently during our daily lives. Recently, some relevant researches are proposed for span detection. As a result, detection of propaganda techniques in news articles is badly in need of research. Li et al., 2021 described a process to detect span from sentences. Fig-2.2 implies the complete architecture of the system.

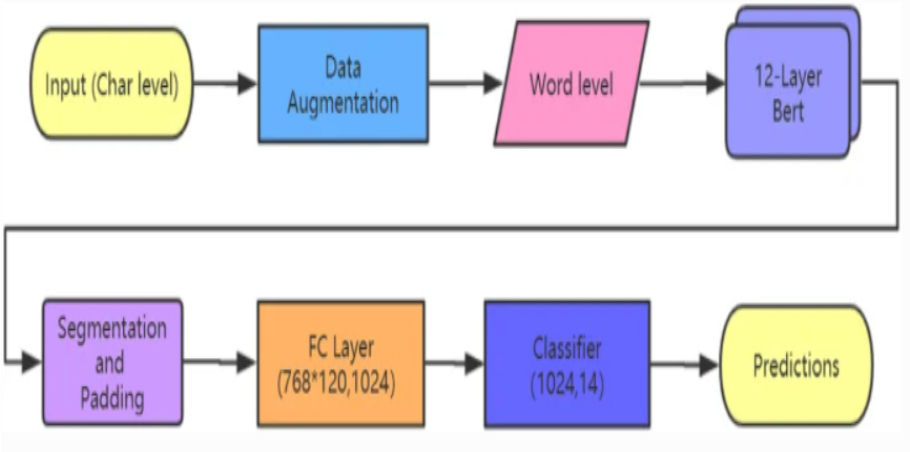

Figure 2.2: Proposed workflow from Li et al., 2021

Fake news is a real problem in today's world, and it has become more extensive and harder to identify. Raza and Ding, 2022 proposed a novel fake news detection framework that can address these challenges. The proposed model is based on a Transformer architecture, which has two parts: the encoder part to learn representations from the fake news data and the decoder part that predicts the future behaviour based on past observations.

In this study, Horne et al., 2019 examine the impact of time on state-of-the-art news veracity classifiers. We show that classification performance for both unreliable and hyper-partisan news classification slowly degrade over time. This degradation happens slower than expected, illustrating that hand-crafted, content-based features, such as style of writing, are fairly robust to changes in the news cycle.

In this study Birim et al., 2022, additional feature combinations to a sentiment analysis are searched to examine the critical problem of fake reviews made to influence the decision-making process using review from amazon.com. Results of the study points that behavior-related features play an important role in fake review classifications when jointly used with text-related factors. Verified purchase is the only behavior related feature used comparatively with other text-based features such as sentiment scores and bag of words.

Chapter 3

DATA PREPARATION & ANNOTATION

Large amounts of training data are required to create an AI or machine learning model that behaves like a human. A model must be trained to grasp specific information in order to make judgments and take action. The categorising and labelling of data for AI applications is known as data annotation. For a specific use case, training data must be correctly classified and annotated. Companies may establish and improve AI solutions by using high-quality, human-powered data annotation. Product recommendations, appropriate search engine results, computer vision, speech recognition, chatbots, and other features improve the consumer experience.

3.1 DATA CRAWLING

Social media platforms like Twitter are one of the great repositories for gathering datasets. Working on a new data science project necessitates a significant amount of data, and acquiring this data is not simple. And because Twitter is a compilation of tweets from people with various thoughts and attitudes, it presents a diverse genre of data. This type of bias-free dataset is essential for training a new machine learning model. I am using tweepy to crawl the twitter data. There are few constraints for tweepy like you can only scrape tweets that are not older than a week. And a limit in scraping, up to 18,000 tweets in a 15 minute time frame. This is the most important part for the startup of this project. Here I am using Twitter V2 api to crawl the data. I have used the academic research gateway api for this purpose which is more efficient way to get the data. Because I can get more number of data a time with multiple more number of features. This api enables to get i. 10 million Tweets per month, ii. Access to full-archive search and full-archive Tweet counts, iii. Access to

advanced search operators. I have used four keywords(**Firmbill, BengalElection, Vaccination, DelhiRiot**) to get the tweet data for this purpose. It is mostly a keyword based search and Hashtag search. From this whole data I am taking tweet id and tweet text(original tweet) for my research purpose. Some example of crawled tweets are given below on Table-3.1. I have collected tweets based on some timeframe of the tweets got published.

	tweet_id	tweet_text
Bengal Election	1467561135272450000	Bengal is so lucky to have defeated those who'd threatened to turn the state into UP before #BengalElection2021.
	1467560515236810000	@aitcsudip @myogiadityanath @BJP4UP Bengal is so lucky to have defeated those who'd threatened to turn the state into UP before #BengalElection2021
Vaccination	1470081797421776899	Dublin City, Ireland Going for your booster #vaccination
	1470038306943602700	Citizens' action to support global vaccination against Covid-19 #vaccination
Firm Bill	1470109950714160000	With the #FarmLawsRepealed, Ananyashree Gupta writes on how it is important to look at the non-farm sector and labour market to understand the failure of the otherwise reformatory laws. #farmbill https://t.co/r2CG4OqQx5
	1469903591104210000	Ananyashree Gupta writes on how the development of non-farm sectors alongside infrastructural changes in the #agricultural sector could bring reformation of the #Indianeconomy. #FarmBill
Delhi Riot	1468243991867110000	Dinesh Yadav is the first #OmicronVirus of #delhiriot declared by Karkarduma Court.
	1468149335514750000	The first conviction in the #delhiriot case: a man was found guilty of looting and torching a Muslim family's home.

Table 3.1: Some example of crawled tweets

3.2 LANGUAGE IDENTIFIER

We have primarily focused on tweets from Indian users as the source of claims. Four primary topics are selected for gathering tweets using Twitters official API: vaccination in India, Delhi riot 2020, West Bengal Election 2021, and, Farm bill/Farmers protest. To develop a multilingual perspective, we have focused primarily on tweets written in English, Bengali, Hindi, and codemixed (within the three mentioned languages). We prepared a language Identifier based on UTF encoding and one identifier using Meta's fasttext module Joulin et al., 2016. This is specifically made for tweets, as it cleans a given tweet and then identifies the language. It classifies the language of the tweet as English, Bengali, Hindi, Codemixed, and others (containing languages other than the mentioned). Among the codemixed tweets, it can also classify between English-Bengali, Bengali-Hindi, and English-Hindi. The overall language specific data statistics is given below on Table-3.2.

	English	**Bengali**	**Hindi**	**Codemix**	**Total**
No. of data	1255	381	542	750	2928

Table 3.2: Data statistics on different languages

3.3 USER INTERFACE FOR DATA ANNOTATION

We have created an user interface to annotate the data as per my requirement to proceed further with the research.

We have created an user interface to annotate the data as per my requirement to proceed further with the research. First field is "CLAIM", in this we are filling the claim from the tweet if the tweet is having claim, otherwise I have to skip the annotation for the tweet. Then we have to select the claim type and the entities from the tweet and have to add one entry for the tweet. Please refer to the Fig-3.2 for detail view.

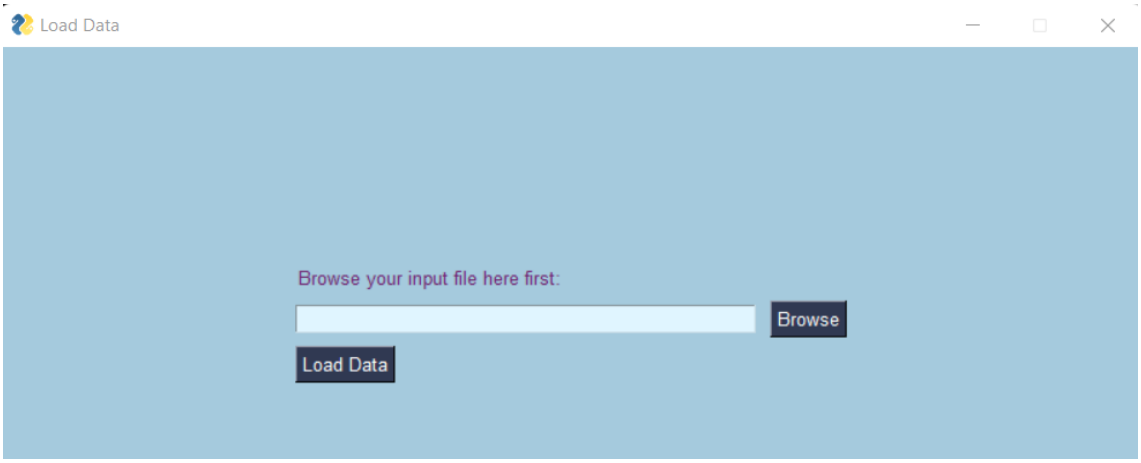

Figure 3.1: Load Excel file here

3.4 ANNOTATION PROCESS

3.4.1 SELECTING CLAIMS and ENTITIES

A claim expressed within a segment of social content (i.e, tweets, reddit submissions, blogs, etc.) should be a minimal span of meaningful text expressing one or more pieces of verifiable information that the author of the said content claims to be true. Additionally, one needs to point out the entities related to the piece of claim.

Minimally meaningful : The annotator should select only that part of the content text that is sufficient to express the information conveyed. For example, in the following tweet:

> #FarmBill
> All 3 farm laws have been repealed within a year of the bill being passed in the parliament.
> Read the farmers bill in detail: https://t.co/PjIsXDDJFf
> #farmbill , #FarmBills , #FarmBill2020, #FarmersBill , #Farmerslaw
> #FarmersBill2020 , #FarmersProtest https://t.co/iyIsp27Fzj

The highlighted segment expresses the piece of information provided by the author meaningfully and sufficiently. Rest of the tweet does not contribute any additional information. The span All 3 farm laws have been repealed stands as an independent claim in itself, but it does not cover the full information conveyed by the author.

Verifiability : A span selected as a claim must express information that the annotator

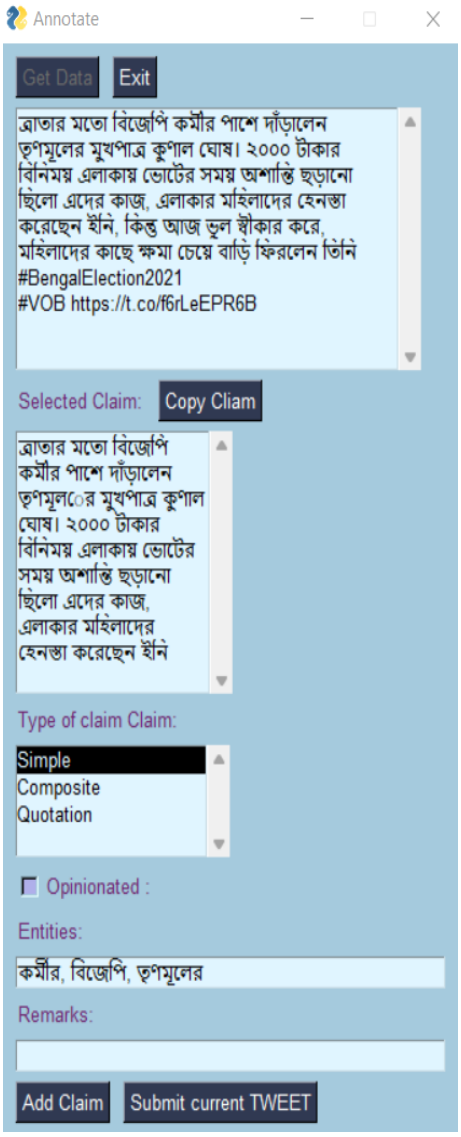

Figure 3.2: Annotate tweet here

thinks to be verifiable, i.e, the truthfulness of the information can be checked from arbitrary external information sources. For example, let us consider the following tweet:

> *sarkarswati*
> *But#Bengal#Mediahasalwaysbeenanti#Narendermodi.*
> *Notsomethingstartedpost*2021*electionresults.*
> *Infactifsomeoneinterestedtocheckhow Petmediaactuallyworks*
> *theycanjustcheckTelegraphandABPgroup.*
> *Forthem#TMCisbeyondanyquestion*

The information expressed in this tweet mostly contains the authors opinion regarding media. It is not possible to verify whether Bengal Media has always been against Narendra Modi (See 3 for more negative examples). Generally, spans like
1) X(some event) happened at Y(some place or time)
2) X(some person or organization) said Y(some statement),
3) X(some quantifiable entity) has a value Y(some numeric value)
4) X(some person or organization) has done Y(some act)
etc. can be readily verified and hence, should be tagged as a claim.

Authorship : A piece of text may contain nested information, like X said that Y happened. In such cases, we need to identify that whether Y happened or not is not the subject of the claim here; the claim is X said so. For example:

> #Delhi riot case: Court slams advocate Mehmood Pracha's claim
> that only particular community was targeted in the riot.

In this tweet, the author claims that the Court has slammed Mehmood Prachas claim and the whole underlined text should be annotated as the claim, not only particular community was targeted in the riot.

Entities : These can be objects, places, persons, organizations, timeframes, etc. that are involved in the information expressed within the claim. For example, in the first tweet, the entities are 3 farm laws, the bill, a year and the parliament. In the second tweet, the entities are Court, Mehemood Pracha, and the riot. Entities should be selected from the whole tweet context if referred but absent within the claim span. For example:

3.4.2 TYPES OF CLAIMS

A span of text that is marked as a claim can be further classified into 3 types:

Simple : the span contains a single factual claim stated directly by the author of the tweet Examples (claim span in yellow):

> Residence of Brindaban Sarkar BJP Candidate, Swarupnagar burned down by TMC Goons.
> Still BJP is in mood of dharna and reports. #SpinelessBJP #BengalElection2021 #BengalBurning #TMC #tmcgoons https://t.co/eaLsZyFKP8

Composite : the span contains multiple nested facts; it is noted that if each of these facts can be marked independently in a meaningful manner, then it is tagged separately. For example, in the following tweet (claim in yellow):

> BJP offices looted, Houses of office workers torched , mens are being brutally slaughtered , womens are raped, blood is all around amd it has just been 2 days since TMC won.
> #Bengal #BengalElection2021 #BengalViolence #BengalBurning #MamataBanarjee #TMCTerror

The selected span states 5 different facts that are expressed together. Any of these 5 facts cannot be marked independently since that would break the meaning of the span. Therefore, this is a composite claim.

Quotation : The author is claiming that some other person/entity is the source of some information. Example,

> In an interview with @AajTakBangla,
> Bengal BJP candidate Mumtaz Ali said: "Our party leadership did not even respond to calls. On the day of the scrutiny, I went alone. Even my election agent was absent."
> https://t.co/7ZuSBL0tjB

It is to be noted that the verifiability axiom need not be satisfied within the quoted statement. That is, X said Y is verifiable (and hence a claim) whether or not Y is verifiable.

3.4.3 MORE EXAMPLES ON CLAIM IDENTIFICATION

In this section, We have mentioned few type of tweets that anyone needs to keep in mind while annotating the data. Suppose Author is announcing an opinion regarding some topic, that kind of tweets are not having any claim into it. Some example of tweets are given below on Table-3.3. Let's pretend Author is presenting his or her opinion on some topic. There is no way to verify the accuracy of any of the statements made.

Tweet	**Claim**	**Remarks**
We all are ready plz through the green signal..... #StopPrivatization #SavePSBs	None	Author is announcing an opinion regarding something, no verifiable information presented.
@IndianExpress Any alliance without @asadowaisi @yadavakhilesh is infructuos in UP. Going without them is to help @BJP4India obliquely,as there is no space left for a 3rd front in the present day elections. #BengalElection2021 is a proof where strong left-Congress alliance doomed.	None	Author is expressing an opinion regarding election strategy. Any statements presented can not be verified for truthfulness.

Table 3.3: More examples of claims

3.4.4 MORE EXAMPLES ON TYPE OF CLAIM

In this part, We have mentioned about few tweets those are having multiple claims in it or having some claims which are composite in nature. Composite tweets are the tweets which are having multiple informations that are needed to be verified. Some example of tweets are given below on Table-3.4.

Claim	**Type**	**Remarks**
MamataBanerjee recently implemented LakhhiBhandar scheme that was part of her #BengalElection2021 manifesto whr women r given Rs 500 to Rs1000 in #Bengal.	Composite	Two pieces of information need verification: i) whether LakhhiBhandar was implemented by MamataBanerjee and ii) details of the scheme
During Left rule MLAs used to travel in buses, now TMC MLAs zip around in chauffeur driven flashy cars	Simple and Simple	Two spans express meaningful information independently. While the whole text is a comparison, that is not of focus. Our task is to identify chekworthy facts and verify later.
BJP4Bengal publishes again the list of star campaigners for the ensuing Bye- Election on October 30, 2021	Composite	Two verifiable facts presented: i) date of bye-election, ii) whether BJP published the list.

Table 3.4: More examples on various type of claims

3.5 DATA STATISTICS

As a source of claims, we largely looked at tweets from Indian users. Vaccination in India, Delhi riot 2020, West Bengal Election 2021, and Farm bill/Farmers' protest are the four key themes chosen for gathering tweets using Twitter's official API. We focused on tweets published in English, Bengali, Hindi, and codemixed to generate a multilingual perspective (within the three mentioned languages). Based on UTF encoding, we created a language identifier. This is designed specifically for tweets, as it cleans a tweet before identifying the language. It segregates tweets as English, Bengali, Hindi, Codemixed, and other languages (containing languages other than the mentioned). It can also distinguish between English-Bengali, Bengali-Hindi, and English-Hindi tweets among the codemixed ones. The overall annotation statistics is given below on Table-3.5.

Language	**English**	**Bengali**	**Hindi**	**Codemix**	**Total**
No. of tweets annotated	1255	381	542	750	2928
No. of tweets containing claim	323	183	188	258	952
Total no. of claim tokens	5909	2200	3303	4764	16176
Total no. of tokens in tweet	30554	5856	14041	20482	70933

Table 3.5: Annotation statistics

Chapter 4

CHECKWORTHINESS IDENTIFICATION

The rise of disinformation (sometimes known as "fake news") on social media has prompted a number of attempts to fact-check and confirm or refute statements of popular interest. Due to the time-consuming nature of manual fact-checking, automated systems have been offered as a faster alternative. Even with automated approaches, however, it is impossible to fact-check every single claim, and automatic fact-checking systems are much less accurate than human experts. Furthermore, prefiltering and prioritising what should be sent to human fact-checkers is required.

The work of estimating check-worthiness, which is viewed as a vital first step in general fact-checking, grew in popularity as the necessity to prioritise grew. We have developed a model for check worthy claim tweet detection in English, Bengali and Hindi. It uses pretrained GloVe word-embeddings and a single Bi-LSTM layer to generate distributed representation of the tweet. An embedding layer is also introduced so that embeddings can be retrained. A feed-forward layer then performs a binary classification (contains claim or not). We have experimented with a two-pronged approach to identify claims from tweets. Typically, the task requires one to first identify which tweets contain a check-worthy claim, and then identify the token-level boundaries of the claims in that tweet.

4.1 SYSTEM DESCRIPTION

We have developed two models for check worthy claim tweet detection in English, Bengali and Hindi. First one uses pretrained GloVe word-embeddings and a single Bi-LSTM layer to generate distributed representation of the tweet. An embedding

layer is also introduced so that embeddings can be retrained. A feed-forward layer then performs a binary classification (contains claim or not). The second one, instead of using a Bi-LSTM to encode the tweet representation, uses pretrained Muril and finetunes it over the task. The Table-4.1 summarizes their performances (P: Precision, R: Recall, F1: Macro F1 score) on different languages.

In English, Bengali, and Hindi, we constructed a model for detecting check worthy claim tweets. It generates a distributed representation of the tweet using pretrained GloVe word embeddings and a single Bi-LSTM layer. There is also an embedding layer that allows embeddings to be retrained. A binary classification is then performed using a feed-forward layer (contains claim or not).

4.1.1 WORD-EMBEDDING COMPONENT

Word embedding is a phrase used in natural language processing (NLP) to describe the representation of words for text analysis, which is often in the form of a real-valued vector that encodes the meaning of the word and predicts the meaning of words that are close in the vector space. Word embeddings are created by mapping words or phrases from a lexicon to real-number vectors using a suite of language modelling and feature learning algorithms. In terms of concept, it entails the mathematical embedding of a multi-dimensional space into a continuous vector space with a considerably smaller dimension.

Individual words are represented as real-valued vectors in a predetermined vector space in word embeddings, which is a class of approaches. Because each word is mapped to a single vector and the vector values are acquired in a manner that resembles that of a neural network, the technique is frequently grouped with deep learning. Some word relations are shown in the Fig-4.1 below from (Pennington et al., 2014).

GloVe is an unsupervised learning technique that generates word vector representations. The resulting representations highlight intriguing linear substructures of the word vector space, and training is based on aggregated global word-word co-occurrence statistics from a corpus. This data is made available under the Public Domain Dedication and License v1.0 whose full text can be found at Pennington et al., 2014. GloVe (Global Vectors) is a word representation approach for distributed systems. This is accomplished by mapping words into a meaningful space in which word distance is proportional to semantic similarity. On the word analogy task, it outperformed many conventional Word2vec models. GloVe has the advantage of explicitly modelling relationships rather than learning them as a side effect of training a language model. This dataset provides pre-trained English word vectors based on the Wikipedia 2014 + Gigaword 5th Edition corpora (6B tokens, 400K vocab). All of the tokens are written in lowercase. There are 300-dimensional word vectors in this dataset.

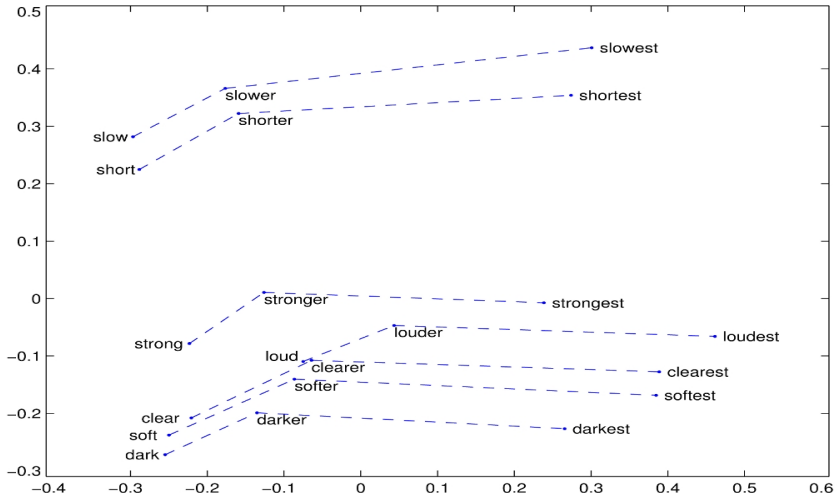

Figure 4.1: Word relations from GLOVE

4.1.2 Bi-LSTM MODULE

Bidirectional long-short term memory (bi-lstm) Fig-4.2 is the process of allowing any neural network to store sequence information in both backwards (future to past) and forwards (present to future) orientations (past to future). Our input runs in two directions in a bidirectional LSTM, which distinguishes it from a conventional LSTM. Bidirectional recurrent neural networks (RNN) are just two separate RNNs joined together. At each time step, this structure allows the networks to have both backward and forward knowledge about the sequence. Using bidirectional will run your inputs in two directions, one from past to future and the other from future to past. What distinguishes this approach from unidirectional is that in the LSTM that runs backward, information from the future is preserved, whereas using the two hidden states combined, you can preserve information from both past and future at any point in time.

We have used this model to identify check worthiness of tweet in English, Bengali and Hindi. It uses pre-trained GloVe word-embeddings and a single Bi-LSTM layer to generate distributed representation of the tweet. An embedding layer is also introduced so that embeddings can be retrained. A feed-forward layer then performs a binary classification (contains claim or not). We have used this model to check whether a tweet is containing claim or not. A detail result of check worthiness of tweets on different languages is shown below on Table-4.1.

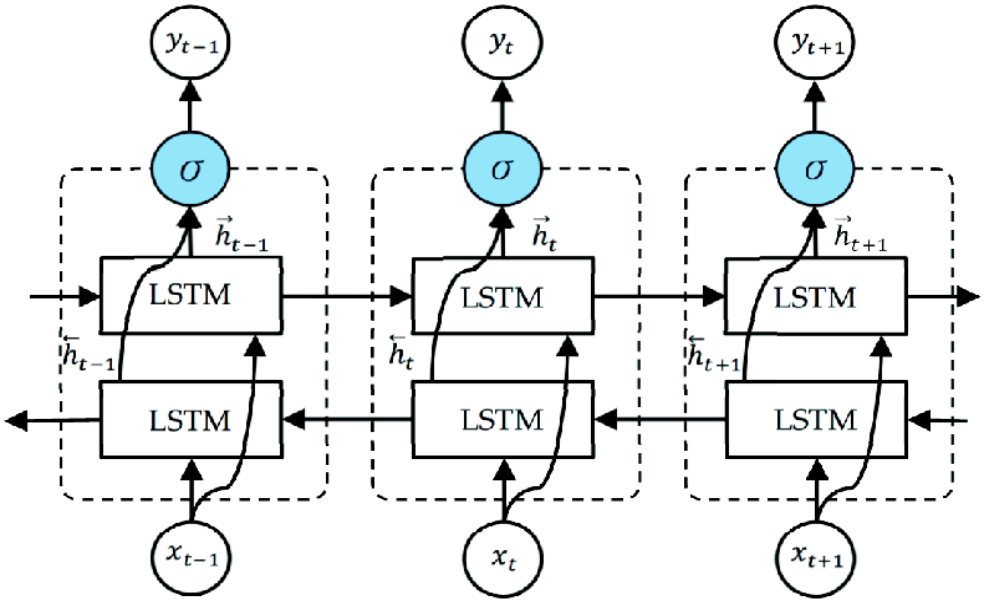

Figure 4.2: Bi-LSTM Architecture

4.1.3 Muril MODULE

MuRIL Khanuja et al., 2021 is a BERT Devlin et al., 2018 model that has been pre-trained on 17 Indian languages as well as their transliterated equivalents. This repository contains the pre-trained model (with the MLM layer intact, allowing for masked word predictions). We've also released the encoder on TFHub with a pre-processing module that converts raw text into the encoder's required input format. This study Khanuja et al., 2021 has more information on MuRIL. This model uses a BERT base architecture pretrained from scratch using the Wikipedia, Common Crawl, PMINDIA and Dakshina corpora for 17 Indian languages. This model is intended to be used for a variety of downstream NLP tasks for Indian languages. This model is trained on transliterated data as well, a phenomomenon commonly observed in the Indian context. This model is not expected to perform well on languages other than the ones used in pretraining, i.e. 17 Indian languages.

We have used this model to check whether a tweet is containing claim or not. We have evaluated this model on English, Bengali, Hindi and Codemix tweet data. A detail result of check worthiness of tweets on different languages is shown below on Table-4.1.

4.2 OBSERVATION

We have tested both model on test data. Following Table-4.1 summarizes the performance of the both models on test data of different languages. We have calculated three evaluation measures (P: Precision, R: Recall, F1: F1 score).

	EN			BN			HN			CM		
Models	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Muril	0.65	0.67	0.65	0.78	0.75	0.73	0.89	0.78	0.80	0.81	0.65	0.64
BLSTM	0.71	0.71	0.71	0.70	0.69	0.69	0.90	0.62	0.64	-	-	-

Table 4.1: Observation from Detection of check-worthy tweets

4.3 ERROR ANALYSIS

In the below Table-4.2, we have mentioned few tweets where both model is behaving differently. In the first tweet Muril model is working correctly and it is able to detect the claim. But Bi-LSTM model is not able to detect the claim. And in the second tweet both models are unable to detect the claim.

Now let's discuss about the statistics that how both models are behaving on test data. In this module, we will discuss all the models' results on test data on different languages one by one. We are using 20% of the total data as test data. The data count for test data is 188. Out of these 64 are English, 36 are Bengali, 37 are Hindi and 51 are codemix.

Lets start with the English data first. For this we are having 64 data as test data in our corpus. Out of 64 data Bi-LSTM model have classified 46 data correctly and 18 data are misclassified. Out of 64 data Muril model have classified 46 data correctly and 18 data are misclassified. Please refer to Table-4.3 for detail more understanding.

Models	English		
	Classified	Misclassified	Total
Bi-LSTM	46	18	64
Muril	57	7	64

Table 4.3: Error analysis statistics for checkworthiness identification on different models on English languages.

Let's start with Bengali data. We have 36 test data in our corpus to use for this. The Bi-LSTM model accurately identified 30 data out of 36, whereas 6 data were

tweet	Result from Muril	Result from Bi-LSTM
WATCH: Pollution levels in the national capital continue to rise; Delhi Government sets up Smog Tower to curb pollution levels https://t.co/RWswfrbOfH #DelhiAirPollution #NewDelhi #StubbleBurning #FarmBill #ArvindKejriwal	1	0
Amit Shah has stepped in to protect BJP workers! #BengalBurning #BengalViolence #BengalElection2021 #Bengal @MamataOfficial @AmitShah https://t.co/sAgKrxGfS0	0	0

Table 4.2: Some Observation from different model behaving on tweets, **1** : Having claim & **0** : Not having claim

misclassified. Muril model properly identified 31 data out of 36, whereas 5 data were misclassified. Please see Table-4.4 for a more detailed explanation.

Models	Bengali		
	Classified	Misclassified	Total
Bi-LSTM	30	6	36
Muril	31	5	36

Table 4.4: Error analysis statistics for checkworthiness identification on different models on Bengali languages.

Let's begin with Hindi data. To do this, we'll take 37 hindi test data from our corpus. Out of 37 data sets, the Bi-LSTM model correctly recognised 29 of them, while the remaining eight were misclassified. Out of 37 data sets, the Muril model correctly recognised 31 of them, while 6 were misclassified. A more complete explanation may be found in Table-4.5.

Models	Hindi		
	Classified	Misclassified	Total
Bi-LSTM	29	8	37
Muril	31	6	37

Table 4.5: Error analysis statistics for checkworthiness identification on different models on Hindi languages.

Our Bi-LSTM model can not handle codemix data as of now. So we are only considering Muril model for codemix data evaluation. Muril model properly identified 45 data out of 51, whereas 6 data were misclassified. Please see Table-4.6 for a more detailed explanation.

Models	Codemix		
	Classified	Misclassified	Total
Bi-LSTM	-	-	-
Muril	45	6	51

Table 4.6: Error analysis statistics for checkworthiness identification on different models on Codemix languages.

Chapter 5

CLAIM SPAN DETECTION

In the domain of Natural Language Processing, detection of Fake news is very challenging and interesting topic. Now a days, social media has come into the picture with a huge amount of data. Some data are generated by human and some data are generated by natural language generation systems, those are synthetically generated. Among these huge volume of data detection of whether it is fake or not is really an interesting problem as well as challenging too. Many researches have been published on this topic, but we are going to discuss on detection of the span form a text which is having the claim.

We formulate the task of identifying claim spans within a tweet as a sequence tagging problem. Existing literature suggests two well-known methods for span identification tasks:

- BIO-tag scheme where the starting token of the span is tagged as B, following tokens within the span are tagged I, and every other token is tagged as O; a CRF layer is used on top of some token encoder, where the CRF dictates the output tag-transition conditioned upon the outputs of the encoder.
- Start-End detection where the model classifies only the start and end tokens of the span; this method is commonly used in question-answering literature. Our initial experiments suggest that the second method performs poorly, majorly due to label imbalance created due to the sparsity of start/end tags. CRF allows us to define explicit transition rules as well (i.e, restricting inavlid transitions). We experiment with three different token encoders alongside the final CRF layer and design 3 separate models: Feature-CRF, BLSTM-CRF, and, Muril-CRF.

5.1 B-I-O TAGGING

In computer linguistics, the BIO format (beginning, short for inner, outside) is a typical tagging format for tagging tokens in a chunking assignment (ex. named-entity recognition). Ramshaw and Marcus presented it in their 1995 paper "Text Chunking with Transformation-Based Learning" Ramshaw and Marcus, 1995. The I- prefix in front of a tag denotes that the tag is contained within a chunk. A token with an O tag does not belong to any chunk. The B- prefix before a tag denotes that the tag is the start of a chunk that follows another chunk without any O tags in between. It's only used in that situation: when a chunk comes after an O tag, the chunk's first token gets the I- prefix.

In Natural Language Processing, a common tagging format for tagging tokens in a chunking task. In computational linguistics, the BIO format (short for inside, outside, beginning) is a typical tagging style for labelling tokens in a chunking assignment (ex. named-entity recognition). The B- prefix before a tag indicates that it is at the start of a chunk, while the I- prefix indicates that it is within a chunk. When a tag is followed by another of the same kind with no O tokens between them, the B- tag is applied. A token with an O tag does not belong to any entity or chunk. Words make up chunks, and the types of words are determined by part-of-speech tags. It's even possible to define a pattern or words that aren't part of chuck, and these are known as chunks. BIO is a format for chunks. These tags are similar to part-of-speech tags but can denote the inside, outside, and beginning of a chunk. Not just noun phrase but multiple different chunk phrase types are allowed here.

5.2 CONDITIONAL RANDOM FIELD

With the interest in Natural Language Processing, entity recognition has witnessed a recent spike in use (NLP). A section of text that is of interest to the data scientist or the company is referred to as an entity. Names of persons, addresses, account numbers, and localities are examples of frequently extracted entities. These are only samples; you may come up with your own entity to solve the problem. CRF (Conditional Random Fields) is a sequence modelling algorithm as well. This not only assumes that features are interdependent, but also that future observations be taken into account while learning a pattern. This algorithm combines the best features of both HMM and MEMM. It is thought to be the best method for entity recognition in terms of performance.

For numerous text classification problems, the bag of words (BoW) technique performs effectively. This method assumes that the existence or absence of a word(s) is more important than the order in which the words appear. However, there are issues like

entity recognition, which is a type of speech identification where word sequences are just as important, if not more important. CRF (Conditional Random Fields) comes to the rescue because it works with word sequences rather than just words. Let's have a look at how CRF is created. The formula for CRF is as follows in the diagram, y represents the hidden state (for example, a segment of speech) and x represents the observed variable (in our example this is the entity or other words around it). Broadly speaking, there are 2 components to the CRF formula:

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp \left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right\}$$

- **Normalization**: You may have observed that there are no probabilities on the right side of the equation where we have the weights and features. However, the output is expected to be a probability and hence there is a need for normalization. The normalization constant Z(x) is a sum of all possible state sequences such that the total becomes 1. You can find more details in the reference section of this article to understand how we arrived at this value.
- **Weights and Features**: This component can be thought of as the logistic regression formula with weights and the corresponding features. The weight estimation is performed by maximum likelihood estimation and the features are defined by us.

5.3 DATASET DESCRIPTION

We have to 2928 no of data set which is annotated in Table-3.2. But for detecting the span from the tweets, we are only using the tweets which are containing the claim. Please refer to the table below Table-5.1 for better understanding. We have used 815 tweets to train (test) these models.

Language	Train Data		Test Data	
	No. of Tweets	No. of Claims	No. of Tweets	No. of Claims
English (EN)	395	524	98	128
Bengali (BN)	240	248	60	60
Hindi (HI)	66	66	16	16
Code-mixed (CM)	114	131	28	34

Table 5.1: Dataset description for train and test data of span detection

5.4 FEATURE-CRF

5.4.1 SYSTEM DESCRIPTION

This model uses manually engineered features that are fed to the CRF model. Following are the text features we used:

- POS Tag
- Last 3 words
- Next 2 words
- Checking if it is a digit or not
- Checking isUpper and isLower(for English only)

Since the features are language dependent, we deploy three separate instances of the model for each of the three languages that work independent of each other and hence requires a language identifier to tag the language of the tweet beforehand.

5.4.2 OBSERVATION

We used exact span matches as correct predictions. Following Table-5.2 summarizes the performance of the three models on test data of different languages (P: Precision, R: Recall, F1: F1 score). Training loss vs Validation loss for all three languages(Bengali, Hindi, English) are shown below Fig-5.1.

Models	EN			BN			HI			CM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Feature + CRF	0.18	0.11	0.14	0.70	0.64	0.67	0.24	0.19	0.21	-	-	-

Table 5.2: Observation from Feature-CRF on test data

5.5 BLSTM-CRF

BiLSTM knows about the language, CRF knows the internal logic of the labeling. Each classification choice is conditionally independent when using a simple BiLSTM followed by a classifier. Linear-chain CRF expresses inter-label dependencies as a table

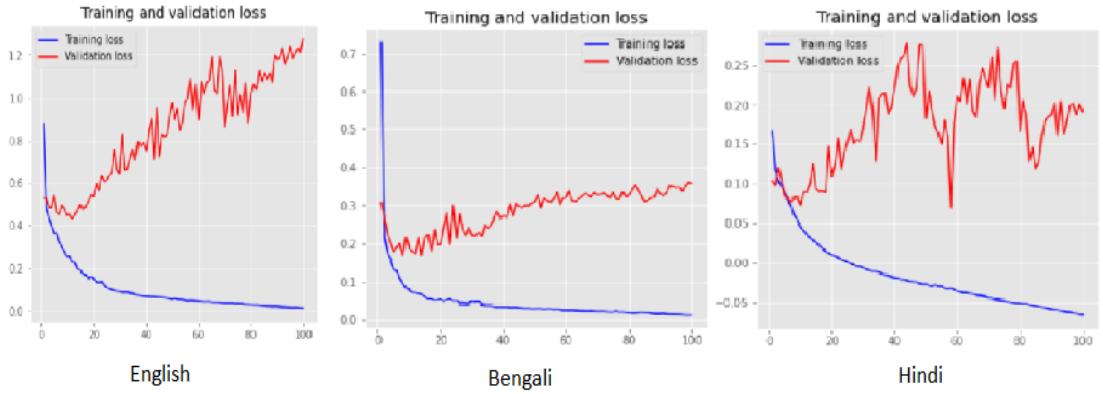

Figure 5.1: Training loss vs Validation loss of Feature CRF

with transition scores between all pairs of labels. If the labels adhere to a tight internal syntax, the CRF will have little trouble learning them. There are numerous ways to encode the output in NER, but they usually encode at least: Beginning, Inside, and Outside of an entity, which must be in a syntactically well-defined order. CRF will immediately see that it is impossible for I-LOC to follow O, and that it must always follow B-LOC. Fig-5.2 describes the architecture of Bi-LSTM followed by CRF.

5.5.1 SYSTEM DESCRIPTION

In this model we are using three separate word embeddings for three languages (English, Bengali, Hindi). For Bengali and English we are using GloVe Pennington et al., 2014 word embeddings, where each vectors length is 300. For Hindi we are using Fasttext Joulin et al., 2016 word embeddings of the same length. A Bidirectional LSTM layer then processes the tweet as a sequence of word vectors and encodes the token-level representations for the CRF layer. Similar to the Feature-CRF, this model is also language dependent.

5.5.2 OBSERVATION

As correct predictions, we chose exact span matches. The performance of the three models on test data from various languages is summarised in the table below (P: Precision, R: Recall, F1: F1 score) table-5.3. The following graphs demonstrate the training loss vs. validation loss for all three languages (Bengali, Hindi, and English) Fig-5.3.

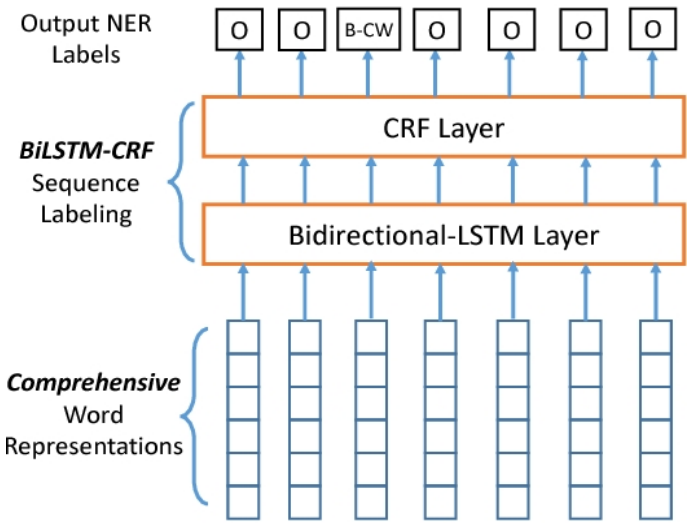

Figure 5.2: Bi-LSTM CRF

Models	EN			BN			HI			CM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Bi-LSTM + CRF	0.29	0.32	0.31	0.66	0.62	0.64	0.48	0.66	0.56	-	-	-

Table 5.3: Observation from Bi-LSTM-CRF on test data

5.6 Muril-CRF

5.6.1 SYSTEM DESCRIPTION

In this model we have used the pre-trained transformer-based language model Muril Khanuja et al., 2021 to encode the tokens. Muril is similar to BERT Devlin et al., 2018 but has been trained on 17 different Indian languages along with their code-mixed instances. We use a linear layer on top of Muril to compute the unary potentials for the CRF. Due to the multilingual pretraining of Muril, this model does not require any explicit language tags for the input tweet and is able to handle code-mixing as well.

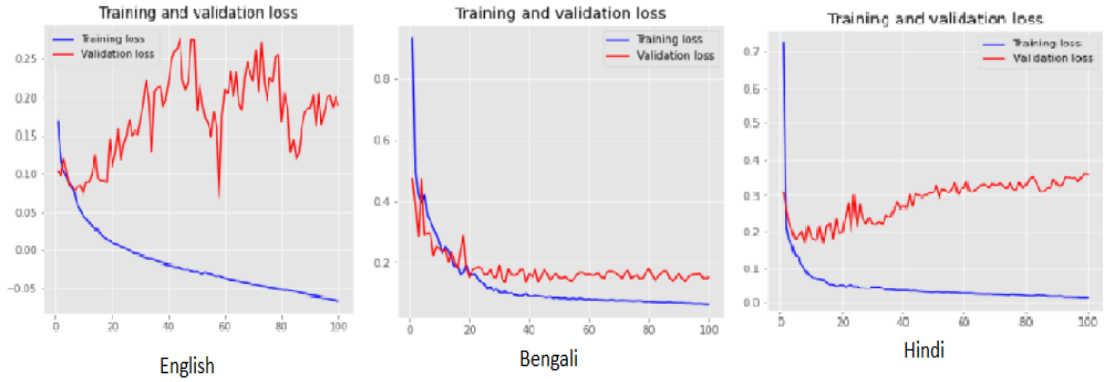

Figure 5.3: Training loss vs Validation loss of Bi-LSTM CRF

5.6.2 OBSERVATION

We used exact span matches as correct predictions. Following Table-5.4 summarizes the performance of the three models on test data of different languages (P: Precision, R: Recall, F1: F1 score).

Models	EN			BN			HI			CM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Muril + CRF	0.64	0.52	0.57	0.69	0.70	0.70	0.56	0.62	0.59	0.46	0.38	0.42

Table 5.4: Observation from Muril-CRF on test data

5.7 MODEL COMPARISON

The updated model with Bi-LSTM and word embeddings is performing better than the previous text features and CRF model. Model-1 (text features + CRF) does not effectively predict, while model-2 (Bi-LSTM + word embeddings) does, according to the annotation. Bi-LSTM is basically the combination of two independent RNNs. With the use of word embeddings, this structure allows the networks to have both backward and forward information about the sequence at each time step. Bidirectional inputs will run in two directions, one from past to future and the other from future to past. What distinguishes this approach from unidirectional is that the LSTM that runs backward preserves information from the future, whereas using the two hidden

states combined, you can preserve information from both past and future at any point in time. So for this our Model-2 is performing better than the Model-1.

We've also created a claim span tagger with Muril that has been pre-trained (a BERT-base model trained on 17 different Indian languages and their codemixed scripts). Muril is utilised to create deep distributed representations of tweet tokens, which is then followed by a feed-forward and a CRF layer to determine token boundaries. Muril's multilingual pretraining means that this model can handle any tweet within our scope without declaring its language attributes directly. Please refer to the table-5.5 for comparison of results on test data between three different models.

Models	EN			BN			HI			CM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Muril + CRF	0.64	0.52	0.57	0.69	0.70	0.70	0.56	0.62	0.59	0.46	0.38	0.42
BLSTM + CRF	0.29	0.32	0.31	0.66	0.62	0.64	0.48	0.66	0.56	-	-	-
Feature + CRF	0.18	0.11	0.14	0.70	0.64	0.67	0.24	0.19	0.21	-	-	-

Table 5.5: Comparison of results on test data between three different models.

5.8 ERROR ANALYSIS

In the below Table-5.6, we have mentioned few tweets where the three models are behaving differently. In the first tweet Muril+CRF model is working correctly and it is able to detect the span. But Bi-LSTM+CRF and Feature+CRF model is not able to detect the span. In the second tweet Muril+CRF and Bi-LSTM+CRF are able to detect the span but Feature+CRF is unable to detect. And in the third tweet which is having composite claim, Muril+CRF is able to detect it properly but Bi-LSTM+CRF and Feature+CRF are behaving differently.

tweet	Result from Muril+CRF	Result from Bi-LSTM+CRF	Result from Feature+CRF
Deaths,Rapes,loot all happening in west bengal and you guys talking about dharna? Shame #BJPFailsIndia #BengalViolence #Bengalisburning #BengalElection2021 #BJP4Bengal @AmitShah @amitmalviya @PMOIndia @JPNadda @BJP4India	['Deaths, Rapes, loot all happening in west bengal']	[]	[]
@AkhilAdityaa Sometime back Delhi also witnessed violence at the hands of Sore loosers. Remember #BJP har jagha violence karti hai. #BengalElection2021 have yet again shown their reality.	['Sometime back Delhi also witnessed violence at the hands of Sore loosers']	['Sometime back Delhi also witnessed violence at the hands of Sore loosers']	[]
@Twitter @TwitterIndia We strongly stand with you. One cheap raid can't shake the Pillar of Democracy. We trusted and vote them to power but now they are destroying India and Indian values. Already their downfall started with slap in #BengalElection2021 #shameonbjp #TwitterIndia	['We strongly stand with you. One cheap raid can't shake the Pillar of Democracy. We trusted and vote them to power but now they are destroying India and Indian values.']	['We trusted and vote them to power but now they are destroying India and Indian values.']	['We strongly stand with you. One cheap raid can't shake the Pillar of Democracy.']

Table 5.6: Some Observation from different model behaving on tweets.

Now let's discuss about the statistics that how all three models are behaving on test. In this module, we will discuss all the models' results on test data on different languages one by one. We are using 20% of the total data as test data. The data count for test data is 188. Out of these 64 are English, 36 are Bengali, 37 are Hindi and 51 are codemix.

Lets start with the English data first. For this we are having 64 data as test data in our corpus. On this test data different models are performing differently. Out of 64 data Feature+CRF model have classified 17 data correctly and 47 data are misclassified. Out of 64 data Bi-LSTM+CRF model have classified 37 data correctly and 27 data are misclassified. And Muril+CRF model have classified 55 data correctly out of 64 data. Please refer to Table-5.7 for detail more understanding.

Models	English		
	Classified	Misclassified	Total
Muril + CRF	55	9	64
Bi-LSTM + CRF	37	27	64
Feature + CRF	17	47	64

Table 5.7: Error analysis statistics for span detection on different models on English languages.

Let's get started with the Bengali data. We have 36 test data in our corpus to use for this. Different models perform differently on this test data. The Feature+CRF model properly identified 11 data out of 36, whereas 25 data were misclassified. The Bi-LSTM+CRF model accurately categorised 23 data out of 36, while misclassifying 13 data. Out of 36 data sets, the Muril+CRF model accurately identified 30 of them. Please see Table-5.8 for a more detailed explanation.

Models	Bengali		
	Classified	Misclassified	Total
Muril + CRF	30	6	36
Bi-LSTM + CRF	23	13	36
Feature + CRF	11	25	36

Table 5.8: Error analysis statistics for span detection on different models on Bengali languages.

Let's begin with the Hindi data. To do this, we'll take 37 test data from our corpus. On this test data, different models perform differently. Thirteen of the 37 data were correctly recognised using the Feature+CRF model, whereas 24 were incorrectly categorised. Out of 37 data sets, the Bi-LSTM+CRF model correctly classified 18 of them, while misclassifying 19 others. The Muril+CRF model correctly recognised 29 of the 37 data sets. A more complete explanation may be found in Table-5.9.

Models	Hindi		
	Classified	Misclassified	Total
Muril + CRF	29	8	37
Bi-LSTM + CRF	18	19	37
Feature + CRF	13	24	37

Table 5.9: Error analysis statistics for span detection on different models on Hindi languages.

Our Feature+CRF and Bi-LSTM+CRF can not handle codemix data as of now. So we are only considering Muril+CRF for codemix data evaluation. Muril+CRF model properly identified 40 data out of 51, whereas 11 data were misclassified. Please see Table-5.10 for a more detailed explanation.

Models	Codemix		
	Classified	Misclassified	Total
Muril + CRF	40	11	51
Bi-LSTM + CRF	-	-	-
Feature + CRF	-	-	-

Table 5.10: Error analysis statistics for span detection on different models on Codemix data.

Chapter 6

CONCLUSION

Identification of checkworthiness is a critical challenge in the internet age. It's getting more popular all across the world. Every year, more journalists and NLP researchers approach this field. As a result, developing automatic systems for checkworthiness detection has both an ethical and an economic perspective. This explains our efforts to develop automated claim verification systems. We have used twitter data for our research purpose. We have developed two models for identifying checkworthiness in twitter data. And three models for claim span detection.

We also have completed the comparison of those model on the test data and found that Muril model is performing better among the other models. We also did error analysis of the results from different models. We are using 4 languages in our data(English, Bengali, Hindi and Codemix(English-Bengali, English-Hindi, Bengali-Hindi)). Feature CRF and Bi-LSTM CRF models are unable to handle codemix type of data as word embeddings for codemix is difficult to built. But Muril model is able to handle the codemix data as well.

We will add more annotated data to our data corpus and train our models with that amount of data to increase the accuracy in future. We will also try to add the codemix sentence detection capability to our both Feature CRF and Bi-LSTM CRF models.

Manually tagging huge amounts of data is a challenge in NLP and in all of the AI world. We also face the same problem in this task of identifying checkworthiness sentences and claim span detection in low resource Indian languages. We anticipate that efficiency will improve in the future as more individuals participate in both activities and more complex models are created. The most crucial question is when considerably more labelled data will be accessible for training.

References

Alonso, M. A., Vilares, D., Gómez-Rodrguez, C., & Vilares, J. (2021). Sentiment analysis for fake news detection. *Electronics*, *10*(11), 1348.

Anand, S., Gupta, R., Shah, R. R., & Kumaraguru, P. (2018). Fully automatic approach to identify factual or fact-checkable tweets. *FIRE (Working Notes)*, 18–23.

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

Birim,. Ö., Kazancoglu, I., Kumar Mangla, S., Kahraman, A., Kumar, S., & Kazancoglu, Y. (2022). Detecting fake reviews through topic modelling. *Journal of Business Research*, *149*, 884–900. https://doi.org/https://doi.org/10.1016/j.jbusres.2022.05.081

Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational linguistics*, *18*(4), 467–480.

Conroy, N. K., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the association for information science and technology*, *52*(1), 1–4.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*. http://arxiv.org/abs/1810.04805

Dhar, R., Dutta, S., & Das, D. (2019). A hybrid model to rank sentences for check-worthiness. *CLEF (Working Notes)*.

Dutta, S., Caur, S., Chakrabarti, S., & Chakraborty, T. (2022). Semi-supervised stance detection of tweets via distant network supervision. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 241–251.

Dutta, S., & Das, D. (2017). Dialogue modelling in multi-party social media conversation. *International Conference on Text, Speech, and Dialogue*, 219–227.

Elsayed, T., & Nakov, P. (n.d.). Alberto barrón-cede no, maram hasanain, reem suwaileh, pepa atanasova, and giovanni da san martino. 2019. checkthat! at clef 2019: Automatic identification and verification of claims. *Proceedings of the 41st European Conference on Information Retrieval (ECIR'19). CEUR-WS. org, Cologne, Germany*.

Elsayed, T., Nakov, P., Barrón-Cedeno, A., Hasanain, M., Suwaileh, R., San Martino, G. D., & Atanasova, P. (2019). Overview of the clef-2019 checkthat! lab: Automatic identification and verification of claims. *International Conference of the Cross-Language Evaluation Forum for European Languages*, 301–321.

Granik, M., & Mesyura, V. (2017). Fake news detection using naive bayes classifier. *2017 IEEE first Ukraine conference on electrical and computer engineering (UKRCON)*, 900–903.

Guha, P., Dhar, R., & Das, D. (2022). Ju_nlp at hinglisheval: Quality evaluation of the low-resource code-mixed hinglish text. *arXiv e-prints*, arXiv–2206.

Gundapu, S., & Mamidi, R. (2021). Transformer based automatic covid-19 fake news detection system. *arXiv preprint arXiv:2101.00180*.

Hassan, N., Adair, B., Hamilton, J. T., Li, C., Tremayne, M., Yang, J., & Yu, C. (2015). The quest to automate fact-checking. *Proceedings of the 2015 computation+ journalism symposium*.

Horne, B. D., NØrregaard, J., & Adali, S. (2019). Robust fake news detection over time and attack. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *11*(1), 1–23.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Kaur, S., Kumar, P., & Kumaraguru, P. (2020). Automating fake news detection system using multi-level voting model. *Soft Computing*, *24*(12), 9049–9069.

Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., Margam, D. K., Aggarwal, P., Nagipogu, R. T., Dave, S., Gupta, S., Gali, S. C. B., Subramanian, V., & Talukdar, P. P. (2021). Muril: Multilingual representations for indian languages. *CoRR*, *abs/2103.10730*. https://arxiv.org/abs/2103.10730

Kim, K.-h., & Jeong, C.-s. (2019). Fake news detection system using article abstraction. *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 209–212.

Li, W., Li, S., Liu, C., Lu, L., Shi, Z., & Wen, S. (2021). Span identification and technique classification of propaganda in news articles. *Complex & Intelligent Systems*, 1–10.

Li, Y., Harfiya, L. N., Purwandari, K., & Lin, Y.-D. (2020). Real-time cuffless continuous blood pressure estimation using deep learning model. *Sensors*, *20*. https://doi.org/10.3390/s20195606

Lillie, A. E., & Middelboe, E. R. (2019). Fake news detection using stance classification: A survey. *arXiv preprint arXiv:1907.00181*.

Mahata, S. K., Dutta, S., Das, D., & Bandyopadhyay, S. (2020). Performance gain in low resource mt with transfer learning: An analysis concerning language families. *Forum for Information Retrieval Evaluation*, 58–61.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The stanford corenlp natural language processing toolkit. *Proceedings*

of 52nd annual meeting of the association for computational linguistics: system demonstrations, 55–60.

Mishra, S., Srivastava, M., Raj, M., Bisoy, S. K., & Khansama, R. R. (2022). Fake news detection using lightweight machine learning models. In M. N. Mohanty & S. Das (Eds.), *Advances in intelligent computing and communication* (pp. 53–62). Springer Nature Singapore.

Nguyen, T. T., Huynh, T. T., Yin, H., Weidlich, M., Nguyen, T. T., Mai, T. S., & Nguyen, Q. V. H. (2022). Detecting rumours with latency guarantees using massive streaming data. *The VLDB Journal*, 1–19.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. http://www.aclweb.org/anthology/D14-1162

Pérez-Santiago, J., Villaseñor-Pineda, L., & Montes-y-Gómez, M. (2022). We will know them byătheir style: Fake news detection based onămasked n-grams. In O. O. Vergara-Villegas, V. G. Cruz-Sánchez, J. H. Sossa-Azuela, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, & J. A. Olvera-López (Eds.), *Pattern recognition* (pp. 245–254). Springer International Publishing.

Ramshaw, L. A., & Marcus, M. P. (1995). Text chunking using transformation-based learning. *CoRR*, *cmp-lg/9505040*. http://arxiv.org/abs/cmp-lg/9505040

Raza, S., & Ding, C. (2022). Fake news detection based on news content and social contexts: A transformer-based approach. *International Journal of Data Science and Analytics*, *13*(4), 335–362.

Rish, I. et al. (2001). An empirical study of the naive bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, *3*(22), 41–46.

Sahoo, S. R., & Gupta, B. B. (2021). Multiple features based approach for automatic fake news detection on social networks using deep learning. *Applied Soft Computing*, *100*, 106983.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, *19*(1), 22–36.

Singhal, S., Shah, R. R., & Kumaraguru, P. (2022). Factdrill: A data repository of fact-checked social media content to study fake news incidents in india. *Proceedings of the International AAAI Conference on Web and Social Media*, *16*, 1322–1331.

Taylor, A., Marcus, M., & Santorini, B. (2003). The penn treebank: An overview. *Treebanks*, 5–22.

Vishwakarma, D. K., & Jain, C. (2020). Recent state-of-the-art of fake news detection: A review. *2020 International Conference for Emerging Technology (INCET)*, 1–6.

Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, *53*(5), 1–40.

Appendix A

Shared Task Participation

In this shared task(INLG 2022 Generation Challenge (GenChal) on Quality Evaluation of the Low-Resource Synthetically Generated Code-Mixed Hinglish Text!), organisers seek the participating teams to investigate the factors influencing the quality of the code-mixed text generation systems. They have synthetically generated code-mixed Hinglish sentences using two distinct approaches and employ human annotators to rate the generation quality. They had proposed two subtasks, quality rating prediction and annotators disagreement prediction of the synthetic Hinglish dataset. The proposed subtasks will put forward the reasoning and explanation of the factors influencing the quality and human perception of the code-mixed text.

In this task Guha et al., 2022, we have described a system submitted to the INLG 2022 Generation Challenge (GenChal) on Quality Evaluation of the Low-Resource Synthetically Generated Code-Mixed Hinglish Text. We have implemented a Bi-LSTM-based neural network model to predict the Average rating score and Disagreement score of the synthetic Hinglish dataset. In our models, we used word embeddings for English and Hindi data, and one hot encodings for Hinglish data. We achieved a F1 score of 0.11, and mean squared error of 6.0 in the average rating score prediction task. In the task of Disagreement score prediction, we achieve a F1 score of 0.18, and mean squared error of 5.0.

We have used a sequence of Glove embeddings as input for English and Hindi sentences. However, for Hinglish sentences we used one hot vector as inputs. We fed the English and Hindi embeddings to separate Bi-lstm's[l-e, l-h], and retrieved sequence output from them. To capture the word sequences of different Hindi and English sentences we have used two different LSTMs. Then we concatenated these 2 outputs and passed it through another Lstm layer to get a fixed (not sequence) vector output [l-h-e].

We fed the one hot vector from the Hinglish data to a dense layer and received a vector output [d-he]. Since one hot vector does not capture the sequential information, we

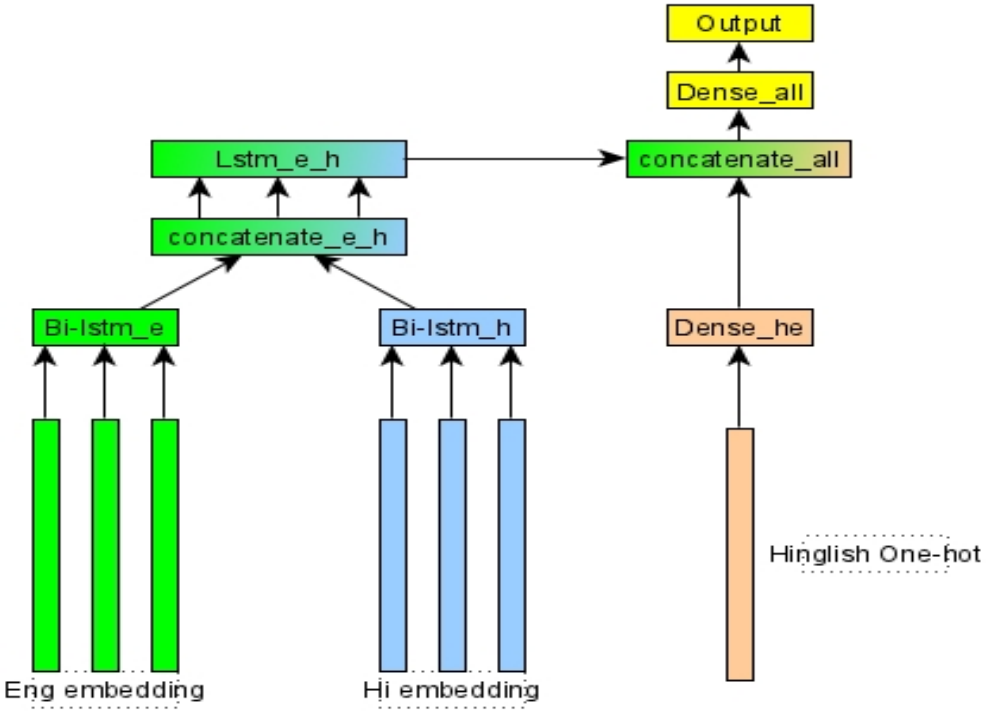

Figure A.1: system architecture

No. of data	F1-Score	Cohen's Kappa	Mean Squared Error
791	0.11582	0.00337	6.00

Table A.1: This result is obtained from 791 test data for Sub-task 1(Average rating score)

have used a dense layer. We then concatenated these two [l-h-e and d-he] vectors, and passed it through a dense layer to get a final class (score between 1 to 10). We used the same model for both the tasks. Please refer to Fig-A.1 for complete system architecture.

On 791 test data, our system is able to predict Average rating for corresponding inputs. Please refer to Table: A.1 for detailed results related to this validation. On 791 data, our system is able to predict Disagreement score for corresponding inputs. Please refer to Table: A.2 for detailed results related to this validation. Among all the participations, We stood 9th in the Sub-task-1 and 6th in the Sub-task-2. The arxiv link for this task can be found here https://arxiv.org/abs/2206.08053.

Future Scope : We already have a annotated dataset from the above mentioned shared task that is containing two scores, i) Average rating , ii) Disagreemnt score.

No. of data	F1-Score	Mean Squared Error
791	0.18331	5.00

Table A.2: This result is obtained from 791 test data for Sub-task 2(Disagreement score)

Those two score is for the generated codemix data(Hinglish) from original Hindi and English data. The significance of the dataset is that, they have generated the Hinglish codemix data from the original Hindi and English data, so the quality of the generated data is measured by those two scores. In our thesis, we have less number of codemix data. But form the Hindi and the English data we can translate both and from the both pair we can generate the codemix data for our purpose. Our shared task's model can able to predict the quality score of the data generated. Using that model we can assign scores to our generated data. And we can add those data which is having good score to our codemix data corpus. That will increase the performance of the proposed models in the thesis.