Heriot-Watt University

Final Year Dissertation

Bias in News Articles

Author:
Marium Sarah

Supervisor:
Dr. Hani Ragab

A thesis submitted in fulfilment of the requirements for the degree of BSc.

in the

School of Mathematical and Computer Sciences

April 2020

Declaration of Authorship

I, Marium Sarah, declare that this thesis titled, 'Bias in News Articles' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: ***Marium Sarah***

Date: ***23nd April, 2020***

"Your time is limited, so don't waste it living someone else's life. Don't be trapped by dogma – which is living with the results of other people's thinking. Don't let the noise of other's opinions drown out your own inner voice. And most important, have the courage to follow your heart and intuition."

- Steve Jobs

Abstract

In recent times, deceptive news on social media has become prominent in shaping the public's perception, leading individuals to take opposing sides, thereby causing conflicts among communities.

The way recommendation algorithms are built on social media platforms allows people to learn about the information that reflects their personal interests. This may steer people away from understanding opposing views on similar topics causing division. Furthermore, during political election campaigning, News websites prevalently may favour a certain viewpoint contributing to articles consisting of fictitious information manipulated to support a political group for profitable or social gains. This may lead to misjudgment and wrong opinions among people.

The objective of this project is primarily to (i) calculate the bias score associated with an article and (ii) provide an analysis for readers to understand the bias aspects of the article. Persuasiveness through news articles is effective when the right people are targeted based on their weaknesses and interests. The aim of this project is to insulate people from being deceived and serve as an alert to help readers understand the bias levels of an article.

We look at various models using different feature-extraction and sentiment modeling techniques to contextualize the text present in articles and find the bias. We also look at web interfaces with the analysis used to assist readers in understanding the bias levels of articles they are reading.

Keywords: *Bias, Sentiment, Credibility, Fake News, Deception*

Acknowledgements

I would like to thank Dr. Hani Ragab for his valuable suggestions and for motivating me to successfully complete my research report.
I would also like to thank my family for encouraging me throughout this journey.

Contents

List of Figures

List of Tables

Abbreviations

RNN	**R**eccurent **N**eural **N**etwork
LSTM	**L**ong- **S**hort **T**erm **M**emory
NLP	**N**atural **L**anguage **P**rocessing
CNN	**C**onvolutional **N**eural **N**etwork
POS	**P**arts **O**f **S**peech
TF-IDF	**T**erm **F**requency **I**nverse **D**ocument **F**requency
LIWC	**L**inguistic **I**nquiry and **W**ord **C**ount

Chapter 1

Introduction

News Bias Score is a measure of how biased an article or news outlet is towards or against a certain topic, it is measured by calculating the ideological scores of news outlets/articles on certain topics [Zhang et al., 2011]. The presence of online news has made it easier for people to access and retrieve news on various topics. With this widespread access comes the possibility of an increase in deceptive news. In most cases, we expect news to be factual and fair in terms of representing various opposing sides of a story. However, due to news channels holding certain views, websites may have a sentiment bias towards a topic causing the article to be unfair and covering descriptions that portray their own interests. It is important to present fair and factual articles to allow people, especially those in a democratic society, to formulate their own opinions by viewing opposing sides.

1.1 Aim

The aim of this project is **to calculate the News Bias Score, and allow readers to understand the bias aspects of articles**. Our focus is primarily on exploring various methodologies used to calculate the News Bias Score and present the bias in the form of an analysis. The overall goal is to flag articles as potentially biased and serve as an alert to readers helping them evaluate their reading habits and be adamant about their views.

1.2 Objectives

The objective of this project is to avoid misleading people with aspects in an article that could negatively affect their judgment about a certain topic.

The following are the primary objectives for this thesis:

- Critically review existing news bias scoring techniques.
- Create a machine learning model that associates a New Bias Score to articles by integrating different sentence modeling techniques introduced in relevant research papers.
- Test the accuracy of the proposed news bias scoring models to other techniques previously implemented using a comparative analysis.
- Compare and Summarize the overall bias score levels of a selection of news outlets.
- Optionally, Create a web-based analysis to allow readers to evaluate bias in articles they read ensuring they are presented with opposing views on a certain topic.

1.3 Manuscript Organization

This project is organized to maintain a flow starting with the **Literature Review** covering the background of News Bias and previously implemented models relevant in this area of study. Next, we have the **Project Implementation**, covering the methodology used to implement the project and the final outcomes achieved. Following this is the **Critical Review** consisting of the methods that were adapted in the final model, the comparison of existing models and the critical evaluation of our findings. Finally, we have the **Conclusion and Future Work** that consists of a summary of the project implementations, challenges faced and Future work that can be carried on by other reseachers interested in this field.

Chapter 2

Literature Review

In this chapter, we will define, research, introduce data encoding techniques and machine learning models used to calculate News Bias in Articles.

2.1 Introduction to News Bias

News Bias is the use of opinions, personal ideology, and reflections in text written by journalists that require factual, reliable and fair information about a topic [White, 1950]. First, we will talk about the emergence of News bias through the spread of misinformation and Fake News.

The Spread of Misinformation

The term Fake News has gained a great deal of popularity in the year 2016 during the US Election. By the 2016 Election, social spam was rapidly evolving into a means of political influence through false or exaggerated captions and text. During this time, fabricated political posts were advertised to social media users in order to persuade them into sharing notorious articles. Among these posts include news claiming Hillary Clinton sold weapons to the Islamic State, that Pope Francis had endorsed Republican candidate Donald Trump, and (from the same source on the same day) that the Pope had endorsed Clinton [Waldrop, 2017].

After the Election, it was revealed that Cambridge Analytica had used personal data of Facebook users in order to build an algorithm to manipulate users' opinions by targeting different groups of people on Facebook and presenting political advertisements that sway them away from their

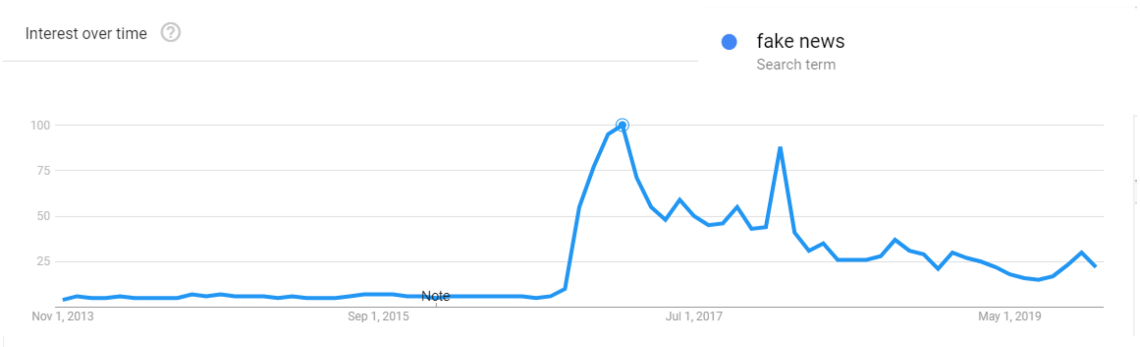


FIGURE 2.1: "fake news" Search Term Google Trends [Trends, 2019]

political interests [Chan, 2019].
What happened in the 2016 US Election is a prominent example that is vital to understand the importance of detecting News Bias.

The term *Fake News* is constantly used and has lead to the creation of various definitions and sub-categories. Common news channels describe Fake News as unreliable sources that impersonate news articles with the primary aim of profit through advertisements around the news.

According to a recent study by Tandoc Jr et al. [2018], Fake News is described to have two main motivations: Financial and Ideological. (1)**Financial**: Factitious stories that use clickbait and sensitive content to convert clicks into advertising revenue. (2) **Ideological**: Using a specific language and selective facts to promote and idea and gain people in favor.

In contrast, Wardle [2017] argues that the term "Fake News" is more specific towards factitious content and is not helpful in properly explaining the ecosystem and complexity involved in Disinformation. She introduces the 7 types of Mis/Disinformation (provided in figure 2.2) with the intent to deceive people in her article. This diagram is efficacious in itemizing a more generalized term into various sub-categories which intern helps researchers focus on a more specific topic in the abstract topic of *Fake News*.

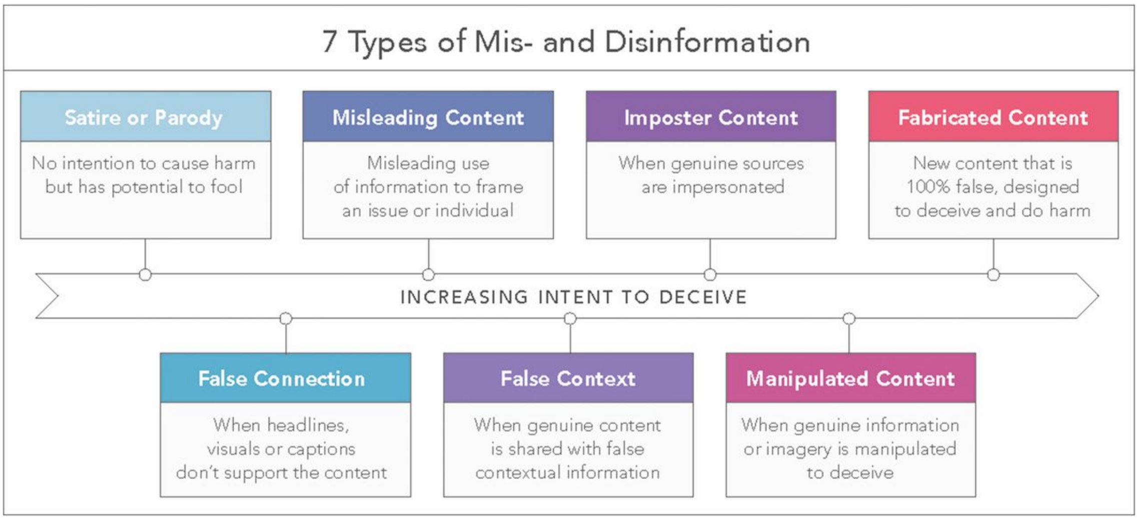


FIGURE 2.2: Types of Mis- and Disinformation by Clarie Wardle

In this paper, we will be focusing primarily on:

1. **Ideologically Motivated**: News used to promote certain ideas often by discrediting others.
2. **Manipulated Content** which refers to genuine information which is manipulated to deceive

Bias in News Articles

Blatant News bias has become a central theme of political maneuvering since the 1980s, playing with numerous economic and ideological demographics within societies [Quackenbush, 2013]. Analyzing the bias aspects of news articles was initially introduced in the 1950s. White [1950] coined the term "The Gate Keeper" as those journalists that intentionally make decisions on information that should be "in" or "out" of news articles which intern causes the shaping of Public opinion. In his study, he analyzes the decisions made by a Wire Editor(the final Gate Keeper) of a morning newspaper who accepts and rejects articles to include in the daily newspaper. A particularly interesting example of an article consisting of a comment about the press not doing enough to convey a controversial story during the time. The GateKeeper rejects this article and comments "Would not Pass- pure propaganda". This illustrates his sensitivity towards the topic and how the readers have presented this article purely due to the

gate keeper's opinion being against the comment made. Furthermore, according to a paper by Quackenbush [2013] on the Meta-Analysis in American News Outlets during the 2012 US Elections, Conservatives often claim that media gatekeepers intentionally shut out conservative ideas.
In the 1950s, stories are transmitted from one gatekeeper to another in a chain before reaching the end reader [White, 1950]. However, in recent times due to the increasing advancements in technology, there is not as much discussion and evaluation done before publishing an article on the Internet. The immense number of articles on a certain topic makes it harder to regulate and identify the ideology and bias in news articles.

Types of News bias

News bias is defined differently by various researchers depending on the context of the research they present. In this paper, the term "News bias" will be used to associate with the sentiment tendency of certain news channels to psychologically manipulate readers [Zhang et al., 2011].

A more general differentiation was provided by Hamborg et al. [2018]. There are two types of Media Bias, **Intentional news bias**, in which an individual or a party is involved in presenting news purposefully with a goal in mind. In contrast to, **Unintentional news bias**, which is caused due to factual news values that present a general visibility on a fair societal relevancy on a certain topic. An example of such news values is an article on an event in the local area of the consumers of the news channel.
In this project, we focus mainly on intentional news bias to help identify the intent of various news channels. However, in our case it may be difficult to differentiate the unintentional and intentional news articles, therefore, the variation of the two types of bias should be taken into consideration during implementation.

In contrast, according to Mullainathan and Shleifer [2002] there are two types of Media Bias, the more traditional one- Ideology, and the less traditional bias referred to as Spin. On one hand, **Ideology** is based on the preferences of the reporters of the article. On the other hand, **Spin** is the attempt to tell a memorable and concise story. In this project, identifying the type of bias an article is targeted towards on the basis of the simplicity and memorability could be a beneficial step during implementation, this will help as an attribute in our calculation of News bias score.

Another definition that is more commonly used is introduced by D'Alessio and Allen [2000] to distinguish News bias in Articles into three types are as follows:

1. **Gatekeeping bias** is concerned with journalists selecting stories that should be presented to the public with ideological intent based on a centric belief of the news outlet. This form of bias is directly linked to the manipulation of the reader through the flow and restriction of information.

   The idea is disadvantageous in practice if we consider a sample of opinionated stories that are selected by journalists from an entire population of all stories. This selection is considered to be biased, however, it is impossible to measure this due to the unidentifiable magnitude of stories available.

2. **Coverage bias** is concerned with measuring the amount of visibility of certain topics, each side of an issue is given.

   Similarly, when we look at Coverage Bias, the coverage of two opposing political parties may be easy to measure and balance out if we measure an equal distribution of coverage for the two sides. However, not all topics are on a binary spectrum such as Conservative and Liberal based articles. A good example provided by Dave Alessio is that of Abortion, there are various opinions a person might have on this simple argument. Making it difficult to quantify what level of deviation is considered to be fair. Therefore, coverage bias is immeasurable when we look beyond the categorization of two opposing sides.

3. **Statement bias** allows journalists of a news channel to input their own opinions and ideologies into the article covering a particular issue. This bias can take two categorical forms of "favourable" or "unfavourable".

   Statement Bias is based on the analysis of declarative and concise sentences present in an article. If a story is covered with an equal amount of statements towards two opposing views, the article is considered to be *Balanced* and *Neutral*, if not it is considered to be *One-Sided* and *Bias*

The three different types of bias presented above, will be beneficial later in identifying Lexicon Resources that will help find bias statements based on the words and parts of speech that are used in sentences.

2.2 News Analyses

It is important to answer the question, Is News Bias a recurring problem that needs to be addressed? Answering this question will solidify the need for the aim of this paper in identifying News bias and allowing for readers to visualize the bias aspects of news articles they read. The following are research papers, analysis and surveys with their results when identifying bias:

- **Comparative Analysis on Danish News Outlets**: In an analytical study done on two Danish Newspapers to identify the representation of two political parties. The results reveal that Information's Articles are more positively worded towards the Alternativet political party, whereas Berlingske are less politically bias towards the two political parties [Enevoldsen and Hansen, 2017].

- **Correlation of Controversy and Bias**: A data-driven approach was proposed by Yelena et Al. to understand emotional expression and biased language in the news. The results of this analysis conveys that, controversial issues consist of fewer positive words and more negative words. The analysis also uses a vocabulary of words signaling bias extracted from Wikipedia. It is concluded that bias terms are more frequently occurring in controversial topics. Furthermore, when it comes to controversial issues, the use of negative and bias language is prevalent [Mejova et al., 2014].

- **Media Bias with Sentiment**: Phalip [2017] discusses her findings on the sentiment levels of 4 well-known US News Channels in regard to a certain topic. For example, with the topic Obama we see that MSNBC has a positive Average Sentiment Score whereas Fox news has a negative score [Phalip, 2017].

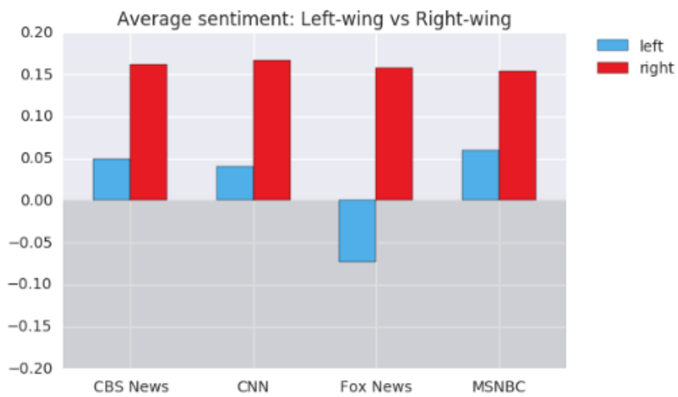


Figure 2.3: Average Sentiment bias Score

Here we see that most of the news channels talk excessively with a positive sentiment about the Conservative Political party, compared to the Liberal. We also notice that Fox News is the only channel to talk negatively about the Liberals.

Therefore, we see that various analyses have been carried out in the past to identify the spectrum of bias in news articles.

2.3 News Bias Datasets

In this section, we first look at an available dataset with articles and different forms of Bias Identification, we then look at other techniques used to extract news articles from websites and use techniques to create a labeled dataset including Bias.

Available News Bias Datasets

The following are a list of available datasets that include a Categorical News Bias identification for News Articles from different outlets.

- **BuzzFeed News Facebook Dataset**: A small dataset of 2000 articles that had high engagement during the 2016 Election. These posts are fact-checked by Buzzfeed journalists along with a political leaning towards right/left-wing identified for each article [BuzzFeedNews, 2016]. This dataset requires a lot of processing due to the diversity of posts allowing Images and Videos with no context.
- **FakeNewsCorpus**: This dataset consists of over 1M articles that are extremely biased. Extremely Bias is defined as "Sources that come from a particular point of view and may rely on propaganda, decontextualized information, and opinions distorted as facts.", the Extremely Bias category in this dataset is suitable for the objective of this paper [FakeNewsCorpus, 2017].
- **NELA-GT 2018**: A large multi-dataset labeled News Dataset for Misinformation in News Articles. In this paper by Norregaard et al. [2019], a dataset of 713K articles is collected in 2018. The articles are collected from 194 news outlets and various assessment sites have been used to identify the bias of the articles.

- **Controversial Words Dataset**: A small dataset of over 400 words consisting of strongly controversial/somewhat controversial and non-controversial terms. This dataset was collected using Crowdsourcing by employing 25 annotators to identify the classification of words [Mejova et al., 2014].
- **Biased Language Lexicon**: A list consists of over 600 bias-related terms extracted from the edit history of Wikipedia's NPOV Disputed Articles [Recasens et al., 2013].
- **Ideological Books Corpus**: This dataset consists of 4,062 sentences annotated for political ideology. It contains 2025 liberal and 1701 conservative sentences[Lazaridou and Krestel, 2016].
- **Unreliable News Data**: A compiled corpus of an unreliable news article from various Resources and with a label identifying the type of news. i.e Satire, Hoax, Propaganda. The **Propaganda** type is articles that mislead readers so that they believe a particular ideology [Rashkin et al., 2017].
- **News Bias through Crowdsourcing and Machine Learning Dataset**: Human judges were recruited via Amazon Mechanical Turk to analyze news articles from various outlets. They were asked questions about the political leaning towards Democrat and Republican parties on a 5-point-scale. Additionally they were also questioned on whether the news was descriptive or opinionated. [udak, 2019].

Although there are numerous datasets available that provide articles labelled as Fake News. In this paper we focus on identifying bias in news articles, therefore only datasets consisting of a bias label in any form will be considered.

2.4 Data processing

Language is powerful in expressing perspective on controversial topics. There is a broad domain of study in Computer Science dedicated to the study of language called **Natural Language Processing**. NLP is associated with the development of systems based on linguistic features, it is used to explore models to allow a computer to understand, manipulate natural spoken language and derive information from it [Reshamwala et al., 2013].

2.4.1 Data Pre-processing

Data pre-processing is used to process text into a representation comprehensible by Machine Learning models.

- **Tokenization**: Tokenization is used to process the sentences into a comprehensible representation for sentence modeling. It involves removing punctuation, unwanted commas, semicolons, ensuring lowercase words and then splitting the text into tokens.[SV. Shri Bharathi, 2019]. NLTK consists of a tokenizer that splits the text into sentences and sentences into typographic tokens.
- **Stop Word Removal**: Some authors remove stop words from the articles. Stop words include commonly used terms such as "the", "a", "an" and "in". This method is specifically used by papers that involve identifying bias words in a text.
  This is not beneficial in our model since we need to consider the context of each sentence.

Other techniques like Stemming are not used due to the importance of the specific parts of speech and word usage identification features in understanding how bias is presented in sentences.

2.4.2 Feature Extraction

The next step is to extract features that will help in identifying bias in news articles. This has been done in different ways by researchers depending on the model they have implemented. The following are a few methods commonly used for feature extraction.

- **Parts of Speech**: POS Identification involves extracting the parts of speech of each word in the article, like adjectives, verbs, and nouns [SV. Shri Bharathi, 2019]. POS tagging uses supervised learning trained on Penn Treebank consisting of Linguistic trees and assigns the POS of specific words based on the structure of a sentence. This is beneficial in understanding how bias in an article differs in terms of using different parts of speech. Additionally, this feature extraction is beneficial when building vector representations of sentences.

- **TF-IDF**: The tf-idf score for each word provides the relevance of the specific word in the context of the grouped articles [Qaiser and Ali, 2018]. It is used by Zhang et al. [2011] to rank each news outlet based on the tf-idf scores from relevant articles. This value is a combination of the number of times the word is mentioned (Term Frequency) and the total number of article divided by total articles with the term present(Inverse Document Frequency) [Qaiser and Ali, 2018].
  In the context of news bias detection, tf-idf can be used to understand which terms are more frequently used in the context of bias and unbias articles, furthermore, this feature can also be used to extract the top Sub-topics from articles.

- **word2vec**: The words in an article are initialized with values that reflect the meaning of the word type [Iyyer et al., 2014]. The word2vec Skip-gram tool kit is trained on a Google News Corpus of over 100b words. A **Skip-gram model** takes every word in a sentence and takes into account other words in the context window, feeding this into a neural network that predicts the probability for each word to be present next to others [Mikolov et al., 2013].

- **DUALIST**: DUALIST is an active learning tool that can be used to reduce the number of neutral sentences in an article [Iyyer et al., 2014]. This can be done by training the model on 200 bias and neutral dataset, the DUALIST classifier uses semi-supervised learning which can predict for a large amount of data with high accuracies using a small amount of labeled data. This method may be disadvantageous considering the sentences in articles are dependent on other sentences to derive meaning.

- **Bag of Words**: is a commonly used simple representation for documents. It involves creating a dictionary of all unique words present in an article with its associated frequency. This feature is disadvantageous as it ignores the richer linguistic context of an article, although it is good for evaluation in a baseline model[Iyyer et al., 2014].

- **Lexicon Resources**: Lexicon resources can be used to differentiate usual articles from Bias articles. Below are Lexicon resources used to categorize differences between articles [Rashkin et al., 2017]:

  1. **LIWC**: LIWC is used to count words in psychologically meaningful categories. It identifies each word in a sentence and associates it with a Linguistic category such as "Killer", "Anger", "Causation" etc. LIWC finally calculates the percentage of each LIWC category in the entire article [Tausczik and Pennebaker, 2010].

2. **Sentiment Lexicon**: Strong and weak subjective words can be extracted using Sentiment Lexicons. This is a beneficial feature to understand if journalists use words to dramatize stories [Rashkin et al., 2017]. It is based on a lexicon dictionary consisting of a list of subjective words with their polarity level.
3. **Hedging Lexicon**: Lexicons for Hedging can be used to understand if articles use vague or ambiguous language [Rashkin et al., 2017]. This may not make a difference in identifying bias since hedging is commonly based on rhetorical questions included in the text, which is usually not present in articles.
4. **Intensifying Lexicon**: Consists of five lists of words associated with a degree of dramatization each word with their measured presence. This can be used to understand whether deceptive news articles try to exaggerate stories to attract readers [Rashkin et al., 2017]. The lexicon resource was compiled by using Wiktionary and is published for use by [Rashkin et al., 2017]

2.4.3 Sentence Modelling

The aim of **Sentence modeling** is to represent the semantic content for classification representations used to encode text best suitable to distinguish bias in phrases and words. As mentioned by Kalchbrenner et al. [2014] in his paper on CNN Modelling, "The ability to accurately represent sentences is central to language understanding". There are various methods implemented in the past to model a paragraph into a Vector Representation. In this section, we will introduce text classification representations used to encode text best suitable to distinguish the bias in phrases and words.

- **RNN Semantic Composition**: A Recurrent Neural Network can model a sentence based on its semantic composition i.e. the meaning of a specific word is a combination of the meanings of the words surrounding it in a particular sentence [Iyyer et al., 2014]. To understand this, we first need to look at how an RNN works.

  Recurrent Neural Network: A Neural Network is an architecture that consists of an Input layer, Hidden layer(s) and Output layer. Given an input, the value is passed on through the hidden layers performing a mathematical calculation in each hidden layer after which the final associated output for the specific input is provided by the output layer. A Recurrent Neural Network consists of several similar neural networks where the input for each neural network is a single word in the article. Each neural network has a hidden layer connected to the successor neural network. This helps pass on the message to a successor for data that requires a successive dependency [Sherstinsky, 2018].

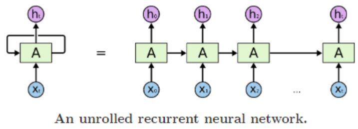


FIGURE 2.4: Neural Network to RNN [Purnasai, 2019]

For the textual representation, the basic RNN Model is used to convert each word into a vector representation in which each word is dependent on other words in the sentence. Each phrase is also associated with a vector in the form of a parse tree capturing the meaning of the phrase using individual word vectors.

- **LSTM Text Encoder**: A method introduced by Gangula et al. [2019] in 2019 uses a bidirectional LSTM (Long-Short Term Memory) to produce a contextual encoding of the headline of an article from both the directions. This will help ensure the connections of the words in the headline are entirely captured. The concatenated forward and backward LSTM together form the representation of the article headline.
  LSTM: An LSTM is a type of Recurrent Neural Network in which the hidden layer is simply replaced by an LSTM Cell and the connection of a Cell State is passed on to the next LSTM Cell for the Successor Neural Network(Figure below). This method was developed in 1991 by Hochreiter and Schmidhuber [1997], the idea of an LSTM Model is to connect previous information to the present task of input and ensure the dependency remains for a long time [Sherstinsky, 2018]. The LSTM Cell uses the **Sigmoid function**

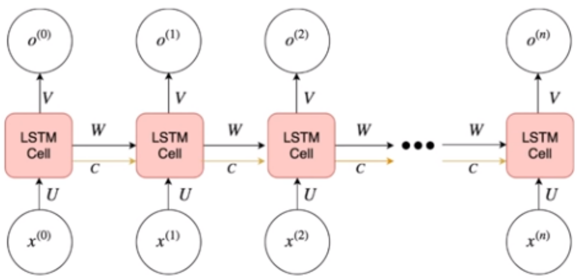


FIGURE 2.5: LSTM Neural Network

for each gate (input,output gates) to ensure the value is smooth in the range 0-1 and the

model remains differential. The **Tanh function** is used to update the cell state, this function ensures that the gradient is distributed and hence prevents the vanishing of the cell state ensuring the dependency between each neural network remains for long.
Bidirectional LSTM is just the concatenation of the output produced by the forward and backward input vectors. This model for encoding is advantageous because:

  - It ensures the contextual information of the text is withheld.
  - The use of a Bidirectional LSTM ensures that the summarization of neighboring words is captured while still ensuring the focus of each word's vector is on the specific word.
  - It performs much better compared to a usual Recurrent Neural Network due to the long term dependency maintained.

- **CNNs**: In a research paper by Kalchbrenner et al. [2014], a Dynamic Convolutional Neural Network is used for semantic modeling of sentences. The layers are formed by convolution operations for the 1D convolutional layers and then pooling operations for the dynamic k-max-pooling layers. **Dynamic k-max pooling** involves returning a set of maximum k values instead of a single max value. Followed by optimizing the k pooling value further. **Max pooling** in CNN is used to reduce the dimensionality of sentences after the convolutional layers apply 1-dimensional features on the sentences.
  This method may prove beneficial considering it reduces the computational cost by reducing the number of parameters.

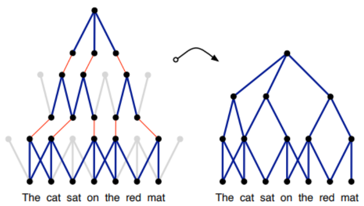


FIGURE 2.6: Dynamic Convolutional Neural Network: The left diagram identifies the pooled nodes. With pooling, phrases far apart in an input sentence can be related.

2.5 Sentiment Scoring for News Bias Analysis

In most cases, a news channel's News Bias is not perceptible. It is primarily through the understanding of choices made by writers on different subtopics for a specific topic that we are

able to identify the bias aspects of the article. A good example given by Zhang et al. [2011] is, given articles on the topic ”Obama”, a news channel may select only the positive subtopics, whereas a channel that does not support him may choose to write mainly about his negative subtopics. Therefore, understanding the Positive and Negative sentiment of various sub-topics in an article will help us access the bias of an article better.
To calculate the sentiment of words, the SentiWordNet package on python can be used. Commonly used for sentiment detection, SentiWordNet is a lexical resource consisting of an abundant list of English terms with their respective Negativity and Positivity score ranging from 0 to 1. In a paper published by Quijote et al. [2019], each word in an article is given a positive and negative score using SentiWordNet. The final Article Sentiment Score is then calculated by addition to obtaining the Positive and Negative score of each word. This score is then normalized by calculating the ratio of each score to the sum of both scores.

2.6 News Bias Detection Techniques

When it comes to News Bias Detection, existing research papers take diverse approaches in creating models. Although the data gathering and evaluation are similar, the models and resulting bias variables are unique depending on the time the paper was published and the data sets that were used. Below we discuss various approaches taken to model a News Bias Detection program collected from different research papers.

One of the earlier works of News Bias detection is a model created by Iyyer et al. [2014] using a **Supervised RNN model**, the basic idea is that a vector representation of a sentence with liberal bias is distinct from that of a conservative bias sentence's representation. This distinction is used to predict the bias using a Supervised RNN. In this method, the cross-entropy loss function is used to measure the error of the prediction. An optimization calculation called L-BFGS was used to optimize the model parameters, the gradient of the RNN was then calculated using backpropagation.
This method benefits from the word2vec sentence modeling representation, it identifies the bias of complex sentences that could not be otherwise detected by baseline models

Gangula et al. [2019] proposed a model that involves taking the **headline of the articles** into consideration. It uses a structure similar to a human's way of reading a news article. The hypothesis of this paper is that biased in an article can be detected by focusing on the headline and essential parts of articles.
In this method, the Bidirectional LSTM Sentence modeling technique is used to convert each word into a vector representation. Each article is given a weighted sum based on the LSTM value. The final vector for each article is then used to detect which political party the article is biased towards using the Softmax function. The training of loss is done with the Negative Log-Likelihood objective.
According to the evaluation performed against other commonly used models, this method outperforms traditional methods namely Naive Bayes, SVMs and CNN by 4%. Therefore, this shows that using headlines can help predict the bias of articles with better accuracy.

The **Inverse Reinforcement Learning model** for calculating Bias was first developed by Lauw et al. [2006]. **Mutual Dependency** of Bias and Controversy is required since identifying the bias of a news outlet requires knowing the controversial articles the channel has talked about and vice versa. The basic approach taken into consideration in this method is as follows:

- Bias: A writer is more biased if there is more deviation from other similar articles on the same topic.
- Controversy: An object is more controversial if there is more deviation by a less biased writer on a less controversial object.

The method produced by Quijote et al. [2019] uses Inverse Reinforcement Learning to calculate the Bias levels of articles from various outlets in the Philippines. (1) the Sentiment Scores are calculated for each article using SentiWordNet, (2) the sentiment values are used to calculate the deviation values for each article compared to other articles from different outlets. These deviation values are then initialized as the controversial values when calculating the Bias Value which is the Average Deviation value multiplied by the complement of the Controversial value for the outlet. (3) After this, the controversial value for each outlet is calculated using a similar formula and an iterative process is used to calculate Bias repeatedly until the bias value converges. The bias score is then subjected to a 4 Level Categorical Value, ranging from Extremely Bias to Allowable Bias. Using this model, the accuracy of 89% of observed when identifying

$$d_{ij} = \frac{1}{m_j - 1} \sum_{r_k \neq r_i} |e_{kj} - e_{ij}|$$

$$b_i = Avg_j \, d_{ij} \cdot (1 - c_j) = \frac{\sum_{j=1}^{n} d_{ij} \cdot (1 - c_j)}{n_i}$$

$$c_j = Avg_i \, d_{ij} \cdot (1 - b_i) = \frac{\sum_{i=1}^{m} d_{ij} \cdot (1 - b_i)}{m_j}$$

FIGURE 2.7: Inverse Reinforcement Learning Model

The Manila Times to be biased. The bias and controversial values are inversely proportional hence the name Inverse Reinforcement Model.
This model is beneficial since it takes into account the deviation of text from other articles on a similar topic. The proposed model is not commonly used and could be improved if we take the Headline and other features into consideration based on previous research mentioned.

In the Content-Based model introduced by SV. Shri Bharathi [2019], the articles are first pre-processed with each word and its consequent POS Tag passed into a Lexicon-Dictionary based approach to identify the opinion and polarity of specific words.
The three opinions are negative, positive and neutral each given values from 0 to 1. With the score for each word, the overall result for a sentence is called using the **Sentence Sentiment Algorithm**. With this, we can find the cumulative score for a sentence by summing up the individual polarity values for each word in the sentence. This cumulative score will help explore the result of biasedness of news articles, we will obtain value from -1.0 to 0.0 and 0.0 to 1.0 for each article. The testing of this algorithm is compared against the WiSARD Algorithm and 7% improvement is documented by this Content Sentiment Analysis Algorithm. This method used a basic calculation when detecting bias, this is beneficial as a baseline method to compare other models for comparison and evaluation.

A method introduced by Zhang et al. [2011] used sub-topic extraction to identify bias in Japanese news websites. This algorithm is unique and effective as it takes into consideration a four-dimension sentiment similar to that of human emotion, unlike the standard negative-positive sentiment. The Four dimensions of Sentiment that are observed are "Joy Sadness"," Acceptance Disgust", "Anticipation Surprise", "Fear Anger". This system firstly extracts all the sub-topics of an article after this, pre-processing is required to obtain the tf.idf and POS Value for each word in an article. A dictionary is then constructed consisting of each word and its sentiment

value including the Scale Value and Weight of all four dimensions. Using these features for the words, a sentiment value for each article on each dimension is calculated by averaging and the sentiment values of words that appear in the article. This is followed by a sentiment vector of four dimensions created for each article which is done by averaging the sentiment values of words appearing in the article. The final value is in the range -0.5 to +0.5, therefore a value close to +0.5 tends to be positive ("Joy","Acceptance","Anticipation", "Fear") while a value close to -0.5 means the sentiment tends to be negative ("Sadness","Disgust","Surprise", "Anger")
The evaluation for this methodology was conducted by comparing user and system on specific bias values of the dimension. According to this evaluation, the system has an average error of only 7%.
This method is advantageous as it considers the sub-topics of each article and identifies bias levels for each. This could be a vital step in understanding the overall bias level of the article.

The method introduced by Mejova et al. [2014] uses various lexical resources to understand the bias in news articles from 4 different US-Based news articles. The general methodology is to collect all articles containing a word (mentioned in controversial and non-controversial topics) from all sources of news media and create a compilation of a single super-article for the analysis. Finally, the bias aspects are analyzed by assessing how prominent bias and sentiment terms are used in controversial topics. The hypothesis of this study is, the use of bias terms are more likely to occur in controversial topics. However, this methodology can be useful in understanding the bias aspects of articles based on the lexicon resources provided.

In the Linguistic based model introduced by Recasens et al. [2013], a variety of linguistic features for each word are taken into consideration and trained on a Logistic Regression model. The main objective of this model is to identify the bias-induced word given a biased sentence. The features used in this model are advantageous for our aim of detecting bias in articles. The following are a few important features that have been used:

- Epistemological bias involves propositions that entail and presuppose things commonly agreed to be true or false. For example: "realized" in "He realized that the oppression of black people...." presupposes the truth of the people in oppression.

- Hedges are used to avoiding bold statements and commitments to the truth using vague terminology. For example: "may have a lower " in "The success of the company may have a lower impact on ..."

- Framing bias involves sentences consisting of one-sided terms towards a controversial issue. For example "liberated" in "Israeli forces liberated the eastern half of Jerusalem." shows the singe opposing view favoring Israel.

A logistic regression model is trained with over 65k features for each word in the NPOV Wikipedia dataset created by Recasens et al. [2013], with bias inducing word as positive and others as negative.
According to the evaluation of this model, this model outperforms all baseline models showing the importance of selecting these features. These features are advantageous as they take into consideration the context of articles.

2.7 News Bias Detection Interfaces

In this section, we will look at interfaces created to assist readers in visualizing the bias aspects of news articles.

1. **OpenMind**: OpenMind is a Google Extension created by two Yale students for a Hackathon Challenge. This program provides alerts when encountering fake news, it also tracks stories someone is reading and identifies bias with a rating of how positive or negative each sub-topic in the story is in the form of simple charts. This is a user-friendly interface serving as an alert system and recommendation system to allow people to look at different perspectives and ensure a balanced news diet.

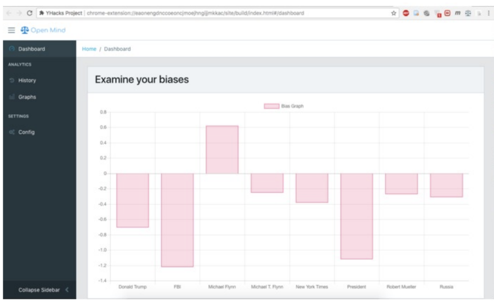


FIGURE 2.8: Examine bias of your reading history using OpenMind

Alternatively for the purpose of this project, this method can be used to understand the bias levels of various news outlets allowing readers to understand the topics that news outlets seemingly support.

2. **Philiphines News Paper Analysis**: A sentiment Bias system based on 4 dimensions of Sentiment was created by Zhang et al. [2011], to allow users to understand the values for each sentiment given a sub-keyword and a news website to analyze. In the figure below, the sentiment level for each news website is provided on all 4 dimensions given a specific keyword.
   A similar implementation can be created for the identification of Bias instead of Sentiment, allowing readers to choose a topic and news website to analyze.

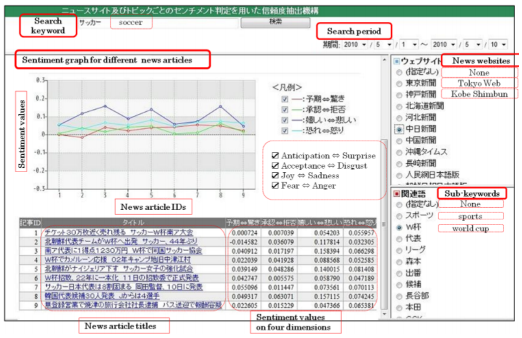


FIGURE 2.9: Sentiment Bias on 4 dimensions based on Keyword and News Website

3. **Bias Free Recommendation System**: A recommendation system created by Patankar et al. [2018] allows readers to understand the bias of articles they read and recommend similar articles on the same topic from different news channels. In this system, the bias score is calculated using the non-NPOV Lexicon resource provided by Recasens et al. [2013], by totaling the bias sentences divided by the number of sentences in the news article. This system will outline the bias score of the article as a user is reading in the form of a graph in comparison to other news outlets articles on the same topic.

The different functionalities provided in the listed three Web based interfaces can be integrated into a single Web-Interface using the bias score calculated.

Figure 2.10: Bias Recommendation System Desktop Version

2.8 Critical Review

In this section, the different approaches in News bias detection are summarized with their pre-processing techniques, methodology and evaluation strategies highlighted.

2.8.1 Dataset Collection

Most authors manually pick topics or keywords for articles to find bias levels in, after this web-scraping tools are used to extract articles from online news outlets. The topics that are often chosen are mainly related to political elections, protests, presidential campaigns, and controversial topics. Gangula et al. [2019] uses a similar technique, however, labeling of bias for each article is done by annotators who are paid to label each article depending on the political party the article supports.

The method introduced by Iyyer et al. [2014] modifies an existing dataset to fit the needs of the proposed model, the dataset consists of speeches from various Political members with their supporting party. This dataset was modified with each sentence from every speech extracted with its associated Political bias.

The following table consists of all the labelled datasets that are appropriate for the aim of this project i.e comparing existing systems in terms of their performance under the same conditions.

Hence we need to find the right dataset that is labelled properly to help us compare the performance of our models. This table covers the advantages and disadvantages of each of the datasets that were considered.

Bias can be identified on various levels, the following table summarizes the different approached used to associate bias to News Articles.

2.8.2 Data Pre-Processing

Most researchers use Tokenization to process the words into tokenized representations.

Additionally, the research papers that are specific towards finding bias terms such as the author Mejova et al. [2014], removal of stop words such as "the", "and","is" is done as part of preprocessing. This was mainly done to ensure that neutral words are removed and all characters are lowercase. Other authors that use Sentiment to detect bias, including the papers by SV. Shri Bharathi [2019] and Zhang et al. [2011], use POS Taggers from the NLTK package to identify opinionated words and their polarity. After preprocessing the data, the sentences in the articles are given in vector representation. The following are the different approaches for Sentence Modelling used by researchers.

2.8.3 Bias Scoring Technique

The following table covers the various approaches used to calculate the bias scores.

2.8.4 Bias Scoring Measure

2.8.5 Evaluation Strategies and Results

Often times the models that are developed are compared to other baseline methods with different features and models. In the paper by Iyyer et al. [2014], three different types of feature-extraction initialization are tested on the RNN model, and traditional baseline methods are compared. Similarly, the inverse reinforcement model by Gangula et al. [2019] was compared by using articles only,headline-only and articles and headline together, This was done to confirm the hypothesis stating headline plays a vital role in identifying bias.

In contrast, some authors evaluate based on results from the conversion values of questionnaires filled by human subjects. For the paper by Zhang et al. [2011], each person is asked to evaluate the sentiment tendency of 4 sentiments, and the evaluation is compared between user and system. The paper by Recasens et al. [2013] compares against traditional baseline models as well as answers provided by human subjects to compare the results of correctly predicted bias-inducing words. Another commonly used evaluation is Accuracy, Precision, and recall used by SV. Shri Bharathi [2019] and Quijote et al. [2019] to compare the accuracy of the proposed model and previously used WizARD Lexicon Library. A threshold of 80% was used as a basis for success.

- Accuracy: The total correctly predicted bias and unbias articles divided by the total number of articles
- Precision: The total correctly predicted bias articles divided by the total correctly predicted bias and wrongly predicted unbias articles.
- Recall: The total correctly predicted bias articles divided by the total correctly predicted bias and wrongly predicted bias articles.

In most cases, splitting the data with 30% Test set and 70% Training set is commonly used. Traditional Baseline Models used for comparison and evaluation include Naive Bayes, SVM,CNN, LSTM, Random , Sentiment(Feature) , Subjective(Feature)

Table 2.5 covers the results obtained from each of the papers that were researched, followed by comments to understand why it's effective and distinctive compared to other models:

2.8.6 Conclusion

Different researchers have chosen different paths when it comes to identifying bias. Some focus primarily on the sentiment of words, others take into account the deviation from articles on the same topic. While more complex techniques involve feature-extractions based on lexical resources that identify bias based on the structure of sentences.
To create a successful model, first, we need to ensure that each sentence's contextual aspects are grasped properly into a vector representation by using Sentence modeling and Bias-Lexical Resources. Second, the model should take other articles on the same topic into consideration. Third, the model should be able to identify bias for independent sentences.

Paper	Description	Advantages	Disadvantages
1 BuzzFeedNews [2016]	Dataset of about 2000 articles that had high engagement during the 2016 election. This includes posts on Facebook with political leaning towards right/left-wing	1. Labelled by trustable annotators paid to read articles and identify bias towards left-/right wing. 2. Contains tags to identify if bias is toward the liberal or conservative political party.	1. Only 2000 Facebook posts, requires scraping of articles from inside of Facebook posts. 2. Some posts may contain videos and pictures, therefore elimination of such articles is required. 3. Does not cover major trustable news channels 4. Most of the articles display NOT FOUND for right-wing articles from specific news outlets.
2 FakeNewsCorpus [2017]	Consists of around 1 million articles tagged as Bias.	1. Scraped using OpenSources which is trustable source of articles scraping. 2. Numerous articles from a variety of sources.	1. Dataset download causes errors, owner was contaced for access but received no response. 2. Not much information about how the bias levels were identified for each article.
3 Norregaard et al. [2019]	A large dataset of about 700K articles from various news sources	1. 194 News outlets 2. Articles are from a range of topics	1. The bias levels are identified at news outlet level but not for each specific article. Each article is not tagged individually.
4 Recasens et al. [2013]	Contains articles that are tagged as NPOV Disputes on Wikipedia. The Neutral Point of View policy by Wikipedia advocates that articles should be fair proportional and far from bias.	1. More than 7000 articles are provided. 2. Provides the words and revision history as converted by annotators to remove bias from articles.	1. The data is not clean, need to extract flagged articles without any revision to the NPOV Disputed articles. 2. Articles are in a seperate XML File that need to be processed into a CSV file properly. This may take a long time to process.
5 Rashkin et al. [2017]	30,000 articles that are tagged as propaganda are provided. Natural News and Activist Report are the two main sources that are used for this data.	1. Contains 7,000 propaganda articles and 1000 trusted articles. 2. Tagged based on news source and by applying various lexical resources from trusted and fake news articles.	1. Only two sources are used. 2. Bias articles are based on news source and not done manually by annotators.
6 udak [2019]	This dataset is generated through an analysis of news articles published online in 2013. Next, Annotators were hired from Amazon Mechanical Turk to identify bias levels towards Democratic and Republican parties on a 5 point scale.	1. A tag to identify if news is Descriptive or Opinionated is also provided. Helpful for further research. 2. Most of the news articles are available for scraping. Although some links display errors.	1. Web scraping for each news outlet is required. Since only the links are provided. Headlines would also need to be scraped. 2. Filtering to identify directly bias articles (e.g. Democrat - Negative, Republican - Positive) is required. 3. Majority of the news articles are from two news outlets.

Table 2.1: Dataset Options

Paper	**Bias Variable**
1 SV. Shri Bharathi [2019]	Ranges from -1 to 1
2 Zhang et al. [2011]	4 Dimensional Sentiment "Joy-Sadness", "Acceptance=Disgust", "Anticipation-Surprise", and "Fear-Anger" each with a discrete value.
3 Recasens et al. [2013]	Probablity of bias associated to words in a sentence.
4 Gangula et al. [2019]	Categorical value of Political Party the article is bias towards.
5 Mejova et al. [2014]	Count of bias terms in controversial topics
6 Quijote et al. [2019]	4 Categories Allowable Bias, Considerable Bias, Biased, Extremely Bias
7 Iyyer et al. [2014]	Political Party (Conservative, Liberal) articles are bias towards

TABLE 2.2: Bias Variable

Paper	**Sentence Modelling**
1 SV. Shri Bharathi [2019]	**Sentence Score Sentiment Algorithm** based on Polarity and Sentiment of words.
2 Zhang et al. [2011]	Each word is associated with its Sentiment value consisting of the **Scale value** which identifies the negativity and positivity of each sentiment dimension and the **Weight** that is proportional to the number of occurrences of the word in similar articles.
3 Recasens et al. [2013]	Feature-based extractions using Lexicon Resources to identify Epistemological and Framing Bias to assign weights to terms that are one-sides, subjective, factive etc.
4 Gangula et al. [2019]	Bidirectional LSTM is used to encode Headlines and the articles to ensure the context of the sentence is considered. This approach ensures the context of the sentence and the word's relationship with other words in the sentence is considered.
5 Mejova et al. [2014]	Lexicon Resources are used to identify sentiment and bias levels of each word in a sentence.
6 Quijote et al. [2019]	All words in the documents were assigned three scores for Positivity, Negativity and objectivity in the range 0-1. Each document was then classified as positive and negative based on this score.
7 Iyyer et al. [2014]	Words were categorized based on the LIWC categories which highlights strong words related to "Anger","Causation", "Kill verbs" etc.

TABLE 2.3: Data processing

Paper	**Bias scoring Technique**
1 SV. Shri Bharathi [2019]	Normalized total score for opinionated sentences in the article is represented as the bias score.
2 Zhang et al. [2011]	A sentiment vector of four dimensions is created by averaging the sentiment values of each word in the article.
3 Recasens et al. [2013]	A logistic regression model trained on the feature vector for each word that appears in the sentence in the training set, with bias inducing words tagged as positive and unbias words as negative.
4 Gangula et al. [2019]	The LSTM-based vector representation of the article is used to find which party the article is biased towards, this is done by using the softmax function with a bias hyperparameter followed by the training loss calculated using negative log-likelihood function of the correct label. This is then used to train the vector representation.
5 Mejova et al. [2014]	In this methodology, each article is tagged on a number of positive, negative and bias terms based on which an analysis is done to compare the controversy and bias relationships of articles. Finally, a logistic regression model is used to assign a score to each of the controversial and non-controversial terms. 1 associate to strongly controversial whereas 0 means non-controversial.
6 Quijote et al. [2019]	Inverse Reinforcement Model is used to detect bias based on the mutual dependency of controversial and bias in articles, it also considers the deviation of articles from other articles on the same topic. Furthermore, the initialization of the hyperparameters is done by calculating the controversial score which proves effective compared to randomization
7 Iyyer et al. [2014]	An RNN model is used on the Vector representation created, 10-fold cross-validation on the training data is then used to find the best RNN hyperparameters.

Table 2.4: Bias Scoring Techniques

Paper	**Result**	**Comments**
1 SV. Shri Bharathi [2019]	The accuracy level has more than 7% improvement for each News topic analyzed.	Due to the simple methodology, this could be used as a baseline model for comparison.
2 Zhang et al. [2011]	The average error is very small between 0.06 and 0.1 which indicates the system is successful in identifying sentiment values similar to that of humans.	4 Dimensions of Human emotions would provide a good graphical analysis for readers. Furthermore, this algorithm considers the deviation of word usage with that of other news articles.
3 Recasens et al. [2013]	The accuracy of this system is the highest at 58% compared to other models having accuracies below 35%. The system is 1% lower than Human annotators labeling bias.	Based on the results, the features used in this model are essential to understanding bias. These features can be used further to compute a biased score for articles. However, the variation of other articles under the same topic is not considered in this method.
4 Gangula et al. [2019]	The model outperforms baseline models by 4.22%, therefore the headline proves to be effective in picking out bias in news articles	With these results, we understand the importance of using the headline, furthermore, the Bidirectional LSTM model proves to be effective in assigning weights that are helpful in bias detection.
5 Mejova et al. [2014]	We notice that approximately 10% of controversial terms are incorrectly classified. The aim of this paper was to understand whether there is a correlation between controversy and bias which is proven to be true according to the analysis.	The findings of this paper suggest that understanding the controversy levels of a topic could help associate a possible bias. The complexity of this model lies mainly in the feature extraction which gives the intended high accuracy.
6 Quijote et al. [2019]	News outlets were deemed bias with an accuracy of 89%	This system is based entirely on the deviation of bias and controversy from other articles. The Inverse Reinforcement Model is not commonly used but gives good results.
7 Iyyer et al. [2014]	The RNN Models outperform the Logistic Regression model by 6%	The phrase annotations allows the model to detect bias accurately even on complex sentences that are not handled by baseline models.

Table 2.5: Results

Chapter 3

Project Implementation

In this section, we discuss the implementation of this project all the way from the Data Extraction to the final evaluation of results. We were able to compare and contrast a few research papers and understand the diversity in terms of identifying the Bias in articles. Since we have implemented different papers with slight variations in pre-processing techniques, the sections are organized as follows:
The first section explains the **Data Collection and Analysis**, followed by the **Generic Initial process** that is required before implementing any model. Finally, we have the three sections for the different implementations that were evaluated.

The plan in figure 3.1 below, was strategically followed for every methodology that was implemented. We used an iterative and incremental model as provided in figure A.2, this was done by iteratively implementing the research models and documenting the performance and evaluation along the way after the Data Extraction, Preprocessing and Data Analysis was complete.

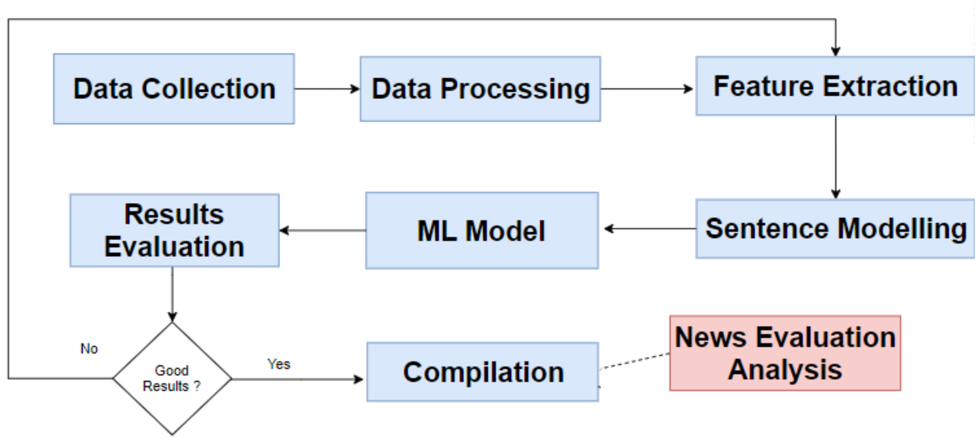


FIGURE 3.1: Implementation Methodology

3.1 Data Collection

Looking at the data sets available with Bias identified in different ways and due to the diversity in the techniques used with the associated output achieved by researchers, we decided to focus primarily on Political articles. This is because of the importance given to this topic in the recent papers that were researched and secondly the availability of data sets with bias levels identified for Political parties in articles.

After looking into the pros and cons of all the datasets (highlighted in Table 2.1). The final dataset that suited with the goal of this project was created by udak [2019] called "News Bias identified through Crowdsourcing and Machine Learning". In this process, Human judges were recruited via Amazon Mechanical Turk to analyze new articles by the top thirteen US news outlets. They were asked questions about the political leaning towards Democrat and Republican parties on a 5-point-scale [udak, 2019].

Since this dataset consisted of links to the article itself, further web-scraping was required to extract the Content and headline of each article. The following steps were followed to achieve this:

1. **Data Filtering and Manipulation**: The Raw Dataset provided needed some initial filtering and extraction to create our final Output field "Bias Towards" with the value Democrat/Republican to identify which political party the article favours.

url	q3	perceived	primary.topic	secondary.topic	democrat.vote	republican.vote
http://www.usatoday.com/story/news/nation/2013/07/15/stateli	other	1	Civil Rights	Civil Rights	Neutral	Neutral
http://www.huffingtonpost.com/2013/05/21/senate-immigration-	News	0	Civil Rights	Civil Rights	Neutral	Neutral
http://www.washingtonpost.com/opinions/dont-let-doma-fool-yo	Opinion	1	Civil Rights	Civil Rights	Neutral	Neutral
http://www.foxnews.com/politics/2013/04/03/obamacare-in-trou	Opinion	1	Civil Rights	Civil Rights	Neutral	Neutral
http://www.breitbart.com/Big-Government/2013/10/09/Exclusive	Opinion	1	Civil Rights	Civil Rights	Neutral	Neutral
http://www.cnn.com/2013/06/28/politics/obama-contraceptives/	Opinion	1	Civil Rights	Civil Rights	Neutral	Neutral
http://www.foxnews.com/politics/2013/10/14/amid-cuts-to-feder	News	0	Civil Rights	Civil Rights	Neutral	SomewhatPositive
http://www.bbc.co.uk/news/technology-22213379	Opinion	1	Civil Rights	Civil Rights	Neutral	Neutral
http://www.nbcnews.com/technology/microsoft-let-nsa-bypass-e	Opinion	1	Civil Rights	Civil Rights	Neutral	Neutral
http://radio.foxnews.com/toddstarnes/top-stories/military-blocks	other	0	Civil Rights	Civil Rights	Neutral	Neutral
http://www.bbc.co.uk/vietnamese/world/2013/08/130812_snow	News	1	Civil Rights	Civil Rights	Positive	Positive
http://www.usatoday.com/story/nation/2013/08/07/jill-kelley-ger	News	1	Civil Rights	Civil Rights	Neutral	Neutral
http://www.dailykos.com/story/2013/12/02/1259558/-Young-whi	News	1	Civil Rights	Civil Rights	SomewhatPositive	SomewhatNegative

FIGURE 3.2: Raw Dataset

Firstly, we filtered the data to remove all links that contained any videos only. Secondly, We used Excel based calculations to compute the BiasTowards category as Democrat/Republican. This was done by looking at the 5 Point scale value provided by the answers to questions about the bias towards Democrat and Republican. For example: One article is tagged as Republican: SomewhatPostive, Democrat: Negative This means the article is favouring the Republican Party.

2. **Pre-Analysis**: Next we identified the news outlets contributing the highest number of the news articles. Since web-scraping for every news outlet is not feasible, pivot tables on Excel were used to understand the total number of Democrat/Republican favouring articles.

Source	Total Articles	Total Republican Bias	Total Democrat Bias
www.dailykos	431	48	383
www.breitbart	303	276	27
www.washingtonpo	142	77	65
www.foxnews	115	114	1
www.huffingtonpos	101	26	75
online.wsj	92	85	7
www.nytimes	89	24	65
news.yahoo	72	30	42

FIGURE 3.3: Pre-Analysis of News Articles Availability

3. **Web-Scraping**: Based on the pre-analysis, we decided to apply web-scraping techniques on article links provided by DailyKos and Breitbart news outlets. To do this, we used the BeautifulSoup package on Python after looking at the articles and the pattern in their HTML Source code to identify the layout on how the content and text is provided.

Step 1: Add Headlines into the Dataframe for each article

```
# Add Headline
for index, row in articles_set.iterrows():
    print(index)
    url_1 = row["url"]
    try:
        page = request.urlopen(url_1)
    except request.HTTPError:
        articles_set.drop(articles_set.index[index])
        continue
    soup = BeautifulSoup(page)
    title = soup.find('title')
    row["headline"]=title.string
```

Step 2: Add Content into the Dataframe for each article

```
articles_set['content'] = ""
for index, row in articles_set.iterrows():
    print(index)
    url_1 = row["url"]
    try:
        page = request.urlopen(url_1)
    except request.HTTPError:
        continue
    soup = BeautifulSoup(page)
    title = soup.find('title')
    print(title.string)
    if(row['News Outlet']=="Breitbart"):
        article_text = ''
        article = soup.find("div", {"class" : "entry-content"}).findAll(['p','h2'])
        for element in article:
            article_text += '\n' + ''.join(element.findAll(text = True))
        row["content"]=article_text
    else:
        article_text = ''
        articles = soup.findAll("div", {"class" : "story-content"})
        for n in articles:
            article_text += '\n'+ n.get_text()
        row["content"]=article_text
```

Figure 3.4: Web Scraping Technique

The final dataset consists of the Headline and Content that needs to be further pre-processed to remove HTML Tags and other characters.

BiasTowards	headline	content
Democrat	Politico Coordinates with Democrats; Attacks V...	\nWhen the trifecta of scandals first broke ov...
Democrat	DOJ Defunds Youth Programs that Reference God	\n\nThe Department of Justice has summarily re...
Republican	Racists Go Wild Outside Obama Speech in Arizona	\n\nMost here saw the Tea Party assholes for w...
Republican	an Obama voter as 3/5 of a person	\n\n3/5 of a person. That should be familiar. ...
Republican	Texas Defends Voting Laws: "We Don't Want Demo...	\n\n\n\n\n\n\n\n\nGood luck with that, Texas\n...

Figure 3.5: Web Scraped Dataframe

3.2 Generic Initial Process

We implemented several papers with slight variations in the pre-processing and feature extraction techniques. In this section, we provide a generic overview of the steps taken that are generic to all implementations.

3.2.1 Data Pre-Processing

3.2.1.1 Balancing Dataset

To avoid situations where the model favours the Democratic Party with more instances, it is important that we balance the dataset and ensure the evaluation of the results are fair.

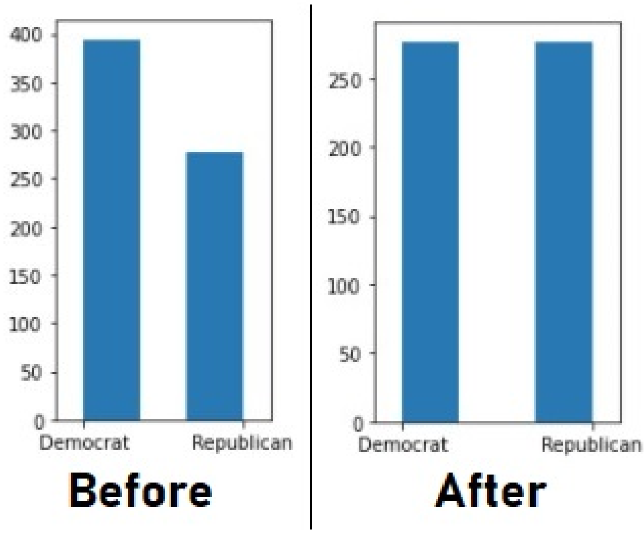


FIGURE 3.6: Balanced Dataset

3.2.1.2 Data Pre-Processing

Before we analyze the dataset that is created, an important step is to ensure that all unnecessary characters which do not play a significant role in classification are removed from the picture. Pre-processing helps simplify the text for our future algorithms implemented. The Pre-processing functions created and applied on our dataset include:

1. ***Removing unnecessary characters***: Since we have used Web Scraping techniques for the retrieval of articles, it is important to remove HTML Tags from the text. This was

done by using regular expression to identify the tags and remove them. Additionally, we also removed characters that could effect our classification.

2. ***Removal of Stop Words***: Removal of stop words was performed by some of the research papers as mentioned in the Literature review. Therefore, this functionality was implemented and evaluated to identify if the performance has an impact.

3. ***Tokenization***: After this, tokenization was performed to split the text into words/sentences depending on the algorithm we are trying to achieve. This helps differentiate and compare the implemented model to identify which performs better i.e Tokenization of words or Tokenization of sentences.

The above preprocessing techniques were applied on three different text representations. This includes:

1. Only Headline
2. Only Content
3. Both Headline and Content

This was done to compare the performance of our model to the Bidirectional-LSTM model implemented by Gangula et al. [2019]. This will further help us understand whether the claim made in the paper about the effectiveness of using only the Headline with an attention layer to predict the political party which the article is favouring.

3.2.2 Exploratory Data Analysis

An important step before implementing a model is to analyze and explore the dataset. This will help us understand how the Democratic and Republican favouring articles compare to each other in terms of word usage.

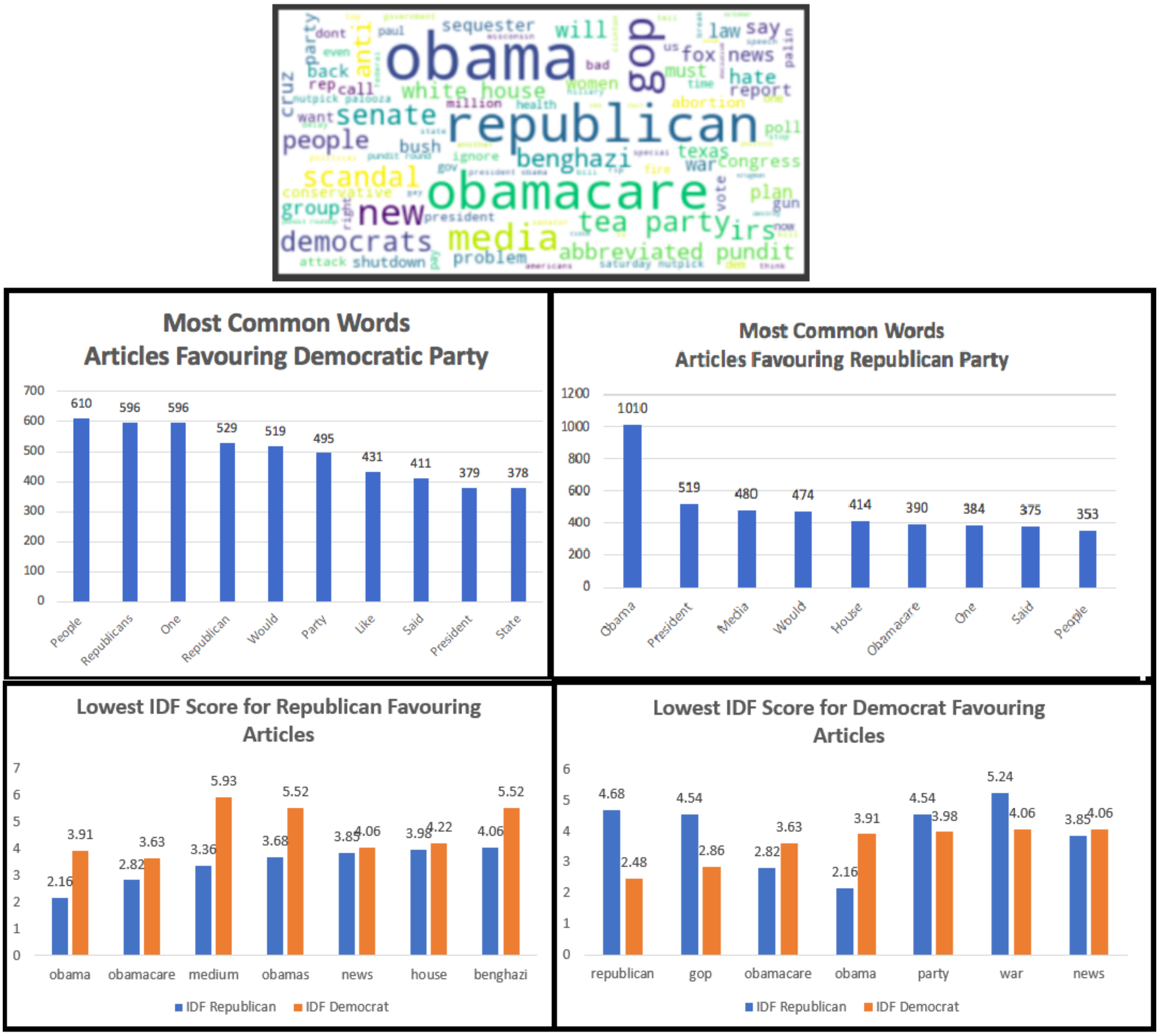


FIGURE 3.7: Exploratory Data Analysis Overview

The first image provides a visual representation of the textual data which in this case consists of all the headlines in the articles dataset. Since the articles are from 2013 - a rather controversial time during Obama's presidency - we see that the most common words include Obama. Furthermore, looking at the most common words in articles favouring the Democratic and Republican parties, we notice that the highest count is for Republican favouring articles towards the word Obama which is twice as much as compared to any other word count. Comparatively, for articles favouring the Democratic party, People and Republican have the highest word count. Analyzing words with the lowest IDF Score for both parties to identify the least unique words revealed that a few words have a high variation between the two parties. This concludes that

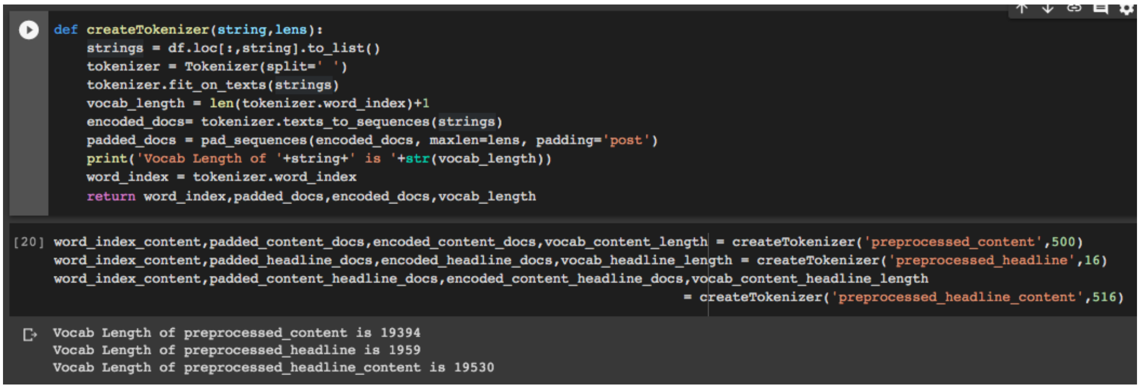
there are differences in the topics that journalists choose to write about from a certain point of view by supporting their views and talking about the opposing political party.
The overall analysis reveals that journalists of Brietbart and DailyKos tend to commonly publish about topics that are relevant to opposing political parties.

3.3 Political Bias using Headline Attention

In this section, we will follow the implementation of the Bidirectional LSTM model presented in the research paper by Gangula et al. [2019], to identify the political party that the article is biased towards.

3.3.1 Data Preprocessing

Vector Representation: To prepare the input for the model, we used a simple tokenizer that firstly, creates a corpus for all the words in the text. This allows us to correlate an array of integers with each of the unique integers in the array representing a word in the associated headline/content.
Padding Sequences: Since the articles can have headlines and content of different sizes, we used the padsequences() function to convert each sequence into a fixed length. The chosen fixed length was based on the average length of a headline and content of the article.



```
def createTokenizer(string,lens):
    strings = df.loc[:,string].to_list()
    tokenizer = Tokenizer(split=' ')
    tokenizer.fit_on_texts(strings)
    vocab_length = len(tokenizer.word_index)+1
    encoded_docs= tokenizer.texts_to_sequences(strings)
    padded_docs = pad_sequences(encoded_docs, maxlen=lens, padding='post')
    print('Vocab Length of '+string+' is '+str(vocab_length))
    word_index = tokenizer.word_index
    return word_index,padded_docs,encoded_docs,vocab_length

[20] word_index_content,padded_content_docs,encoded_content_docs,vocab_content_length = createTokenizer('preprocessed_content',500)
word_index_content,padded_headline_docs,encoded_headline_docs,vocab_headline_length = createTokenizer('preprocessed_headline',16)
word_index_content,padded_content_headline_docs,encoded_content_headline_docs,vocab_content_headline_length
    = createTokenizer('preprocessed_headline_content',516)

Vocab Length of preprocessed_content is 19394
Vocab Length of preprocessed_headline is 1959
Vocab Length of preprocessed_headline_content is 19530
```

FIGURE 3.8: Vector Representation

As seen above, we have created a vector representation in three ways: Headline only, Content only and for both headline and content. This was done to later access the performance on different representations.

3.3.2 Sentence Modelling

1. **Embedding Layer**: The first step, given the vector representation of the text, is to assign weights to each of the words using an embedding matrix. These weights assigned to each word are randomly initialized and later trained for based on the input.

   Regularization is an important step to ensure that the weights assigned to each embedding is not overfitting the model. Regularization is done by adding a term to the cost function during the training of the model [Jain, 2018]. L1 and L2 are two types of regularizations. In L1, the absolute values of the weights are allowed to be reduced to 0 however when using the L2 Regularizer, the weights are reduced to nearly 0, ensuring there are no 0 values for the weights.

In L2, we have:

$$Cost\,function = Loss + \frac{\lambda}{2m} * \sum \|w\|^2$$

FIGURE 3.9: L2 Regularizer Jain [2018]

   The key difference between these techniques is that L1 Regularization shrinks the less important feature's coefficient to zero thus removing some feature altogether [Nagpal, 2017]. So, this works well for feature selection in case we have a huge number of features, however, since we have less features of high importance, we will be using the L2 commonly used regularizer.

2. **Bidirectional LSTM**: The next step is to use Bidirectional LSTM to get the contextual encoding of the text in both the directions. By using Bidirectional LSTM, the forward LSTM uses the input of words in the text from start to finish and the backward LSTM uses the input of words from the end till the beginning of the text. Finally, the text is encoded by concatenating the forward and backward representation. This was implemented using Keras Bidirectional functionality applied to a usual LSTM Model similar to the following representation provided.

q_1:

$$x_i = W_e q_i, i \in [1, H] \qquad (1)$$

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(x_i) i \in [1, H] \qquad (2)$$

$$\overleftarrow{h_i} = \overleftarrow{LSTM}(x_i) i \in [H, 1] \qquad (3)$$

FIGURE 3.10: Bidirectional LSTM Sentence Modelling Gangula et al. [2019]

Finally, the output representation of the text is a concatenation of the the forward and backward LSTM computed.

3.3.3 Attention Layer

The next step is to create an Attention Layer to extract words that contribute to the classification of the political party.

The implementation followed for the attention layer was provided in a comprehensive review of Attention mechanism by Hore [2019a]. The goal with this attention layer is to capture how relevant the ith word in the text is, as compared to the other words. This will help the model focus on the most important words.

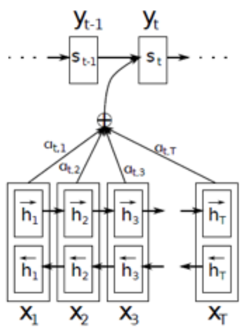


FIGURE 3.11: Attention Layer Hore [2019a]

The output of this attention layer is a vector for every word and its importance as compared to other words. These are called the Attention Vectors. In practice, a tan hyperbolic function is applied followed by a softmax function that is used to calculate the similarity with each word in the text.

3.3.4 Bias Detection

Finally, the vector v generated is used to detect the political bias using a **Sigmoid function** since we have a binary classification of Republican or Democrat. The model presented in the paper by Gangula et al. [2019] uses a Softmax function which is used due to the numerous political parties that are available in their dataset.

The loss function used to identify the error is Binary Cross Entropy. This function is used to understand how far away from the true value the prediction is for each instance. The average of all the instances is then identified as the final loss [Hore, 2019b].

$$L(y,\hat{y}) = -\frac{1}{N}\sum_{i=0}^{N}(y * log(\hat{y}_i) + (1-y) * log(1-\hat{y}_i))$$

FIGURE 3.12: Binary Cross Entropy Hore [2019a]

The below code represents the entire model that was inspired by the m, the if condition is used in the argument to allow a simple comparison between the performance using an Attention Layer and without using an Attention Layer.

```
inputs=Input(shape=(dimension,))
x1=Embedding(input_dim=vocab_length+1,output_dim=dimension,\
         input_length=dimension,embeddings_regularizer=l2(.001))(inputs)
if attention_flag==1:
    att_in=Bidirectional(LSTM(100,return_sequences=True),merge_mode='concat')(x1)
    x1=attention()(att_in)
else:
    x1=Bidirectional(LSTM(100),merge_mode='concat')(x1)
outputs=Dense(1,activation='sigmoid')(x1)
model1=Model(inputs,outputs)
```

FIGURE 3.13: Final Model inspired by Gangula et al. [2019]

The model presented by created by Gangula et al. [2019] consists of two different encoders one for the Headline and the Article as provided in this diagram below:

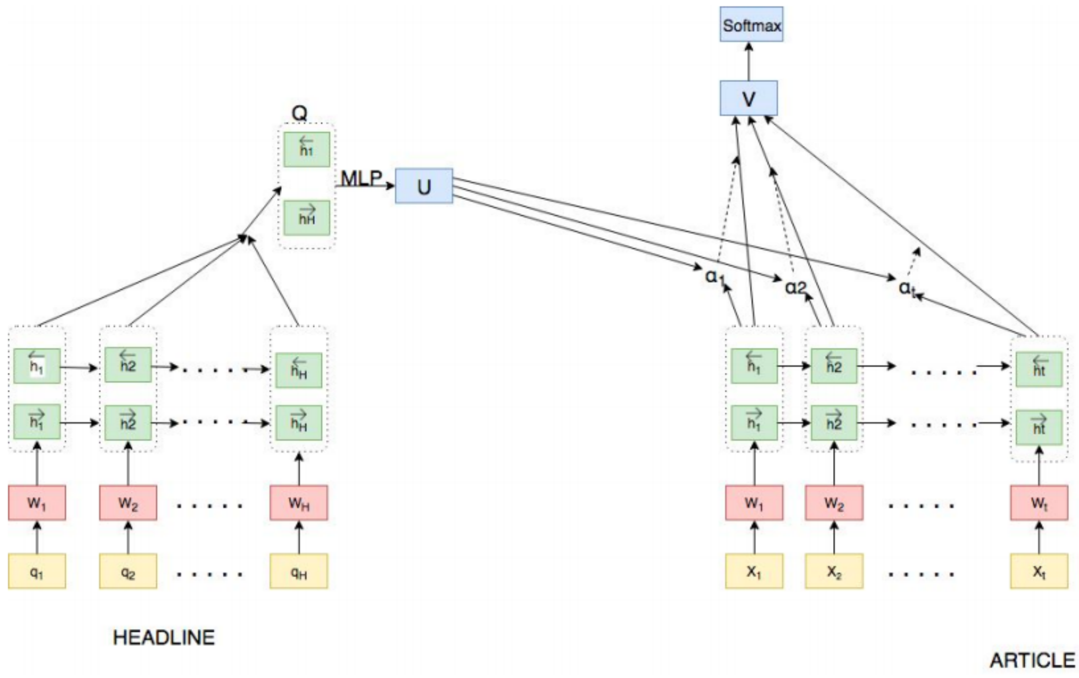


FIGURE 3.14: Headline and Article Bidirectional LSTM Model Gangula et al. [2019]

In the above diagram, the headline and article are encoded separately, followed by an attention layer applied on the headline only. The vector representation outputs of the two models of Article and Headline are then concatenated as input to the Sigmoid function that identifies the political party that the article is bias towards.

In the critical evaluation, we will look at the performance of the concatenated model as given by Gangula et al. [2019] and compare it to the model with the attention layer applied on the text of headline and article combined during the tokenization.

3.4 Sentiment Scoring on Sentence Level

In this implementation, we look at a feature based model inspired by the paper on Sentiment Scoring on Sentence Level by SV. Shri Bharathi [2019]. In this paper, a Sentence Score Sentiment Algorithm is used to identify the bias score of the entire article. Using the idea of a cumulative sentiment score associated for each sentence in the article, we will use a similar idea and train this as an input vector to a few models while comparing it to baseline models using CountVectorizer, TFIDF Vectorizer and Binary Vectorizer as the feature extractions of the words.

3.4.1 Feature Extraction

There are four different feature extraction techniques we have tested with on the headline, content and both headline and content consecutively. The following are the four feature extraction techniques used:

1. **Sentence Level Sentiment Scoring**: In this technique we assign the Polarity of each word and the sum is calculated to find the total score value for each sentence [SV. Shri Bharathi, 2019]. With this in mind, we created a vector representation associating each article to a list of integers corresponding to the sentiment score of each sentence. To ensure a consistent length for the model to train on, we identified the average number of sentences in articles and used it as a limit.

   To test this further, we did this specifically for Headlines, Content and Both Headline and Content to analyze the variation in performance.

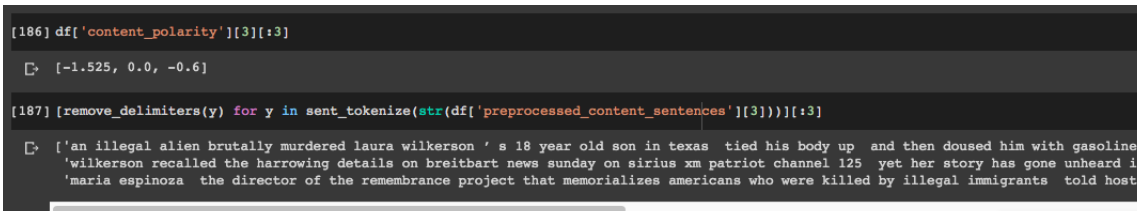


FIGURE 3.15: Sentiment Scores for Sentences

   As we see above, the first and third sentences talk about killings hence they have a negative score as compared to the second sentence which has a neutral score of 0.

2. **CountVectorizer**: Each sentence that is parsed is converted into an array of unique integers associated with each word in the sentence and a count value for the number of times the word appears in the context.

3. **Binary CountVectorizer**: Using the binary=True argument in CountVectorizer instead of a frequency based value of the count of words, will give us the 1/0 value depending on whether the word exists in the Sentence

4. **TFIDF Vectorizer**: The TFIDF vectorizer is used to associate a float value to each word in a sentence. This value identifies the TFIDF score to understand the importance of a word in the specific article as well as in the entire corpus.

3.4.2 Training Model

The 4 different Feature Extraction techniques were tested on a common Logistic Regression model as done in the research paper by SV. Shri Bharathi [2019]. This will help us understand if there are any dependencies or patterns in the input vectors. Furthermore, with Logistic regression we can analyze the probabilities that the article is favouring a certain Political Party.

```
X_train,X_test,feature_transformer=extract_features(df,field,training_data,testing_data,type=types)

# INIT LOGISTIC REGRESSION CLASSIFIER
scikit_log_reg = LogisticRegression(verbose=1, solver='liblinear',random_state=0, C=5,max_iter=1000)

model = scikit_log_reg.fit(X_train,Y_train)

predicted = model.predict(X_test)
```

FIGURE 3.16: Logisitic Regression Model

As for the Sentence Level Sentiment Vectorization, due to a low performance with the Logistic Regression Model, we further analyzed the performance with K-Means Clustering and a baseline Sequential Model to see if there was an improvement in performance.

3.5 Linguistic Models for Bias Detection

In this section we will look into the features based approach of using Lexicon Resources provided in the paper by Recasens et al. [2013] to identify if such features have an effect in the classification of Bias in articles. Additionally, the paper on Quantifying Bias by Hutto et al. [2015] have utilized these features and created a computational model with a high performance of 97%. Hence, we will be using the feature extractions provided by this paper in our implementation.

The following are the features we will be using as part of the features based on the JSON file provided by Hutto et al. [2015] which compiles all the important features needed.

1. *Presupposition Verbs* comprised of Coherence, Implicative, Factive and Assertation markers are those that presuppose the truth on a certain topic.

2. *Hedges* are used to present a topic with certain ambiguity to avoid bold statements asserting the truth.

3. *Bias/Partisan words* comprise of words that are considered to be bias based on the Wikipedia Neutral Point of View dataset.

4. *Opinionated words*

5. *Sentiment Scores* based on the Vader Sentiment package identify the polarity score of the sentences in an article.

6. *Figurative Expression*

7. *Pronounces* a lexicon to capture the pronouns used in the text.

8. *Word Count*: The Total word count and unique words were used as features to identify if there are any correlations with other features.

9. *Flesch-Kincaid Grade Level* formula is an important metric to identify how readable the article is. Given a higher grade level associates with a more difficult text [Hutto et al., 2015].

$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right)+11.8\left(\frac{\text{total syllables}}{\text{total words}}\right)-15.59$$ [15]

FIGURE 3.17: Flesch-Kincaid Grade Level Calculation

10. *Negative Perspective*: Identify the count of words in a sentence that consists of negative assertions such as "can't", "wasn't", "despite" etc. This will help understand if sentences usually have a negative perspective in the text.

11. *Quoted Content*: Identify if sentences consists of quotations or citations.

    In the paper by Hutto et al. [2015], the bias score is also computed based on a pre-trained Logistic regression model which showed a high performance.

    Using the features provided by this paper, we performed Logistic Regression on the features with the highest correlation based on the findings in the original paper by Hutto et al. [2015] to further analyze how they compare to the performance of the previous implementations.

Chapter 4

Critical Review

In this chapter, we analyze the performance of all three implementations with a critical evaluation on the findings. Additionally, we look at the testing and evaluation to understand the output of the implementation.

4.1 Performance Evaluation

To test our model and ensure an accurate performance is analyzed we used 3 Fold Cross Validation. This was done due to small dataset we are using. We will be accessing the performance of identifying the Political Party that an article is bias towards by looking at the Accuracy, Precision, Recall and F1 Score of all the models.

4.1.1 Political Bias using Headline Attention

The research paper by Gangula et al. [2019] focuses on using a Headline Attention Layer and Bidirectional LSTM Encoding to identify the political party that an article is bias towards. With this in mind, we evaluated the performance of our model inspired by the implementation in this paper.

The following graph concludes the findings of each model variation in terms of accuracy:

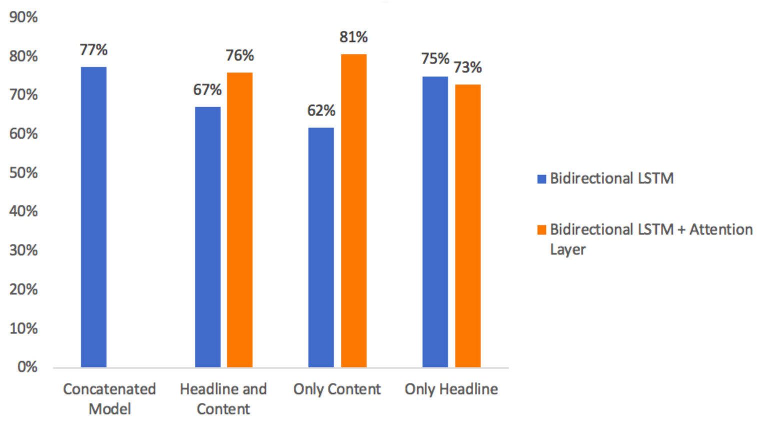


FIGURE 4.1: Accuracy vs Input

Firstly, we analyze the Average Accuracy of the models when running it on 3-Fold Cross Validation. The following are the findings evaluated based on the performance.

- It was concluded by Gangula et al. [2019] that adding an attention layer to the headline specifically will improve the accuracy of the model and help capture the bias in articles. However, we further looked at the effects of using the Attention layer on the content of the article as compared to only the headline. This improved the accuracy by 9% whereas using both the headline and content as input to the Attention Layer had a lower accuracy of 76%.

- As compared to the implementation in the research paper of first encoding the headline and content separately and then concatenating the two to identify the bias, alternatively, we decided to concatenate the headline and content before hand to see how the performance varies. Doing this led to an increase in accuracy by 5%.

- Finally, looking at the different inputs to the model. We noticed that without an attention layer the highest accuracy is achieved by the model that was trained only on the Headline at 75%. This may indicate that the Bidirectional LSTM captures the dependencies of the headline properly to identify the bias of the article. However, adding an attention layer to this model reduces the accuracy as compared to using content as input.

S.No.	Model	Input	Accuracy	F1 Score	Precision	Recall
1	Bidirectional LSTM	Only Headline	74.91%	74.11%	76.56%	71.85%
1	Bidirectional LSTM + Attention Layer	Only Headline	72.74%	71.40%	76.60%	69.29%
1	Bidirectional LSTM	Only Content	61.74%	55.29%	72.15%	47.68%
1	Bidirectional LSTM + Attention Layer	Only Content	80.52%	80.31%	81.51%	79.30%
1	Bidirectional LSTM	Headline and Content	66.97%	58.62%	78.64%	47.06%
1	Bidirectional LSTM + Attention Layer	Headline and Content	75.81%	77.21%	73.89%	82.88%
1	Bidirectional LSTM	Concatenated Model	77.26%	78.52%	74.78%	82.69%

FIGURE 4.2: Comparison of Performance

The above table provides the accuracy, F1 Score, Precision and Recall for all the models that were tested. From this we conclude that the best performing model is when using only Content as input and including an Attention Layer.

After this we did the Train Test split of 70-30% and trained the model to achieve an overall accuracy of 87% on the Test data. The results of testing this on a new input is given in the System Output section.

4.1.2 Logistic Regression Models

For the feature extraction based research papers that were implemented and inspired by Recasens et al. [2013] and SV. Shri Bharathi [2019] we evaluated the performance of the model based on the average Test set accuracy after performing 3 Fold cross validation.

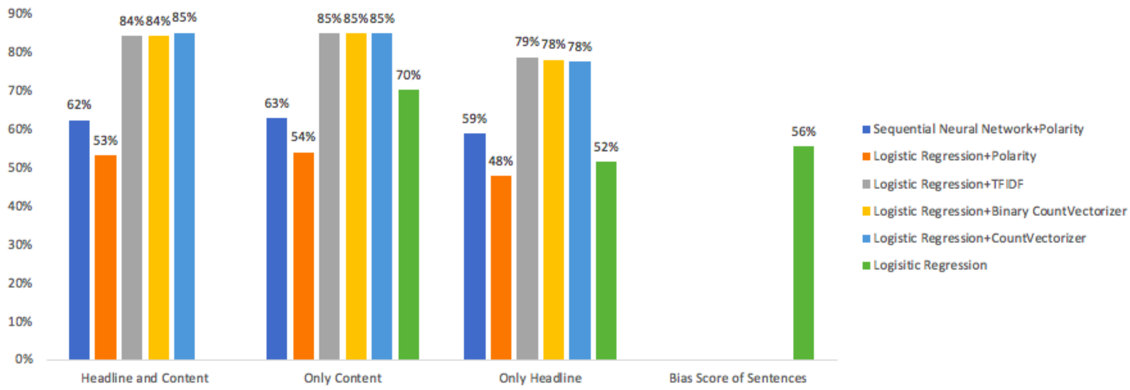


FIGURE 4.3: Comparison of Performance - Logistic Regression

Analyzing the performance of the Logistic Regression model, we see that the best performing model are the ones using the simplistic approach of feature extractions. Using the TFIDF scores gave the highest accuracy followed by Binary Count Vectorizer and Count Vectorizer.

Looking at the performance of the implementation inspired by SV. Shri Bharathi [2019], we noticed that using only sentiment as a feature has no effect in detecting the political party the article is bias towards. In conclusion, the results of this model performed on Sentence level, using only headline or only content still achieves a performance closely similar to that of a randomized prediction.

Analyzing the performance of using the various Lexicon based features to identify the bias in articles inspired by the paper by Recasens et al. [2013] and Hutto et al. [2015]. Evaluating the graph (Lexicon Based Approach in Green) we see that using Logistic Regression specifically on the Lexicon based features of the Headline does not perform as required. However, using the features based on the entire content of the article and specifically looking into the features that provide high correlation as provided by the Hutto et al. [2015] we were able to obtain an accuracy of 70% on this Lexicon Based approach model.

4.2 Performance Comparison

The following table provides the Accuracy, F1 Score, Precision and Recall for each of the methodologies that were implemented and compared.

S.No.	Model	Input	Accuracy	F1 Score	Precision	Recall
1	Bidirectional LSTM	Only Headline	74.91%	74.11%	76.56%	71.85%
1	Bidirectional LSTM + Attention Layer	Only Headline	72.74%	71.40%	76.60%	69.29%
1	Bidirectional LSTM	Only Content	61.74%	55.29%	72.15%	47.68%
1	Bidirectional LSTM + Attention Layer	Only Content	80.52%	80.31%	81.51%	79.30%
1	Bidirectional LSTM	Headline and Content	66.97%	58.62%	78.64%	47.06%
1	Bidirectional LSTM + Attention Layer	Headline and Content	75.81%	77.21%	73.89%	82.88%
1	Bidirectional LSTM	Concatenated Model	77.26%	78.52%	74.78%	82.69%
2.1	Sequential Neural Network+Polarity	Only Headline	59.03%	46.99%	66.52%	36.88%
2.1	Logistic Regression+Polarity	Only Headline	48.00%	37.26%	53.27%	37.39%
2.1	Sequential Neural Network+Polarity	Only Content	62.99%	63.60%	62.60%	65.40%
2.1	Logistic Regression+Polarity	Only Content	54.14%	51.47%	55.77%	48.88%
2.1	Sequential Neural Network+Polarity	Headline and Content	62.45%	62.11%	62.68%	62.12%
2.1	Logistic Regression+Polarity	Headline and Content	53.32%	50.18%	54.72%	47.49%
2.2	Logistic Regression+TFIDF	Only Headline	78.80%	79.20%	77.30%	79.90%
2.2	Logistic Regression+Binary CountVect	Only Headline	78.16%	78.53%	77.39%	79.91%
2.2	Logistic Regression+CountVectorizer	Only Headline	77.61%	77.81%	77.39%	78.40%
2.2	Logistic Regression+TFIDF	Only Content	85.00%	85.55%	82.77%	88.62%
2.2	Logistic Regression+Binary CountVect	Only Content	85.00%	85.00%	84.94%	85.32%
2.2	Logistic Regression+CountVectorizer	Only Content	85.00%	85.00%	85.82%	84.36%
2.2	Logistic Regression+TFIDF	Headline and Content	84.29%	84.89%	81.90%	88.21%
2.2	Logistic Regression+Binary CountVect	Headline and Content	84.47%	84.50%	85.13%	83.88%
2.2	Logistic Regression+CountVectorizer	Headline and Content	85.20%	85.29%	85.19%	85.48%
3	Logisitic Regression	Bias Score of Sentences	55.55%	55.20%	56.12%	55.61%
3	Logisitic Regression	Features Content	70.39%	69.32%	72.05%	67.40%
3	Logisitic Regression	Features Headline	51.62%	51.00%	52.99%	51.50%

FIGURE 4.4: Performance Comparison

Evaluating the accuracy in each section, we used the best performing models from section one and three to obtain the final output of our Bias Detection system. We did not integrate any models from the second implementation due to the low accuracy when implementing the Polarity Sentence Scoring approach. Even though the baseline implementations in the second approach has a good performance, the output of this model does not provide information that the readers can use to analyze the articles they are about to read, hence we did not include this in our final output.

4.3 System Output

Implementing and identifying the categorical bias of Democrat and Republican was done in three very different ways. Firsly, using an LSTM Model. Secondly, using a sentence level sentiment based approach followed by an entirely Lexicon based approach.

With these three different implementations, we compiled what the final output would look like by evaluating the output given two articles as input. These two articles were published in the month of April, 2020. One of the articles is favouring Democrat and the other is favouring Republican.

To take a step back, our main goal was to allow readers to understand the bias levels of an article before they make the decision on whether to read the article. With this in mind, the following are the compiled outputs that will help readers analyze the bias in news articles.

This first article talks about a Senate Majority leader's views against the actions taken by the Democratic Party during the COVID-19 Pandemic that is currently circulating.
Republican Favouring article Link **Published on April 21st, 2020**

FIGURE 4.5: Republican Party Article

To analyze the information the system would provide for this article above. The information on the *right* is the output of our first implementation of the Bidrectional LSTM model with an attention layer. And the output provided on the *left* is given based on the Lexicon features applied to the content of the article. The features were carefully selected to display based on the information provided by Hutto et al. [2015] about the features that are most highly correlated to the bias of news articles. This includes the following

- Sentiment Score: Associated with the sentiment score of the entire article with a metric value from -1 to 1 with -1 representing an article which is very negative and +1 represents a positive article.
- Negative Perspective Score: This value is calculated based on the number of negative words used in the text. The higher the value the more negative the article is.
- Doubt,Partisan,Opinion Words %: This value represents the overall percentage of words in the dataset that fall into the categories based on the Lexicon dataset we have available.

This second article talks about the Corona virus pandemic's effects on the Economy and how the Republican Party is willing to take any action regardless of the consequences to save the economy.. This article is clearly favouring the Democratic Party

Democrat Favouring Article link - **Published on April 13st, 2020**

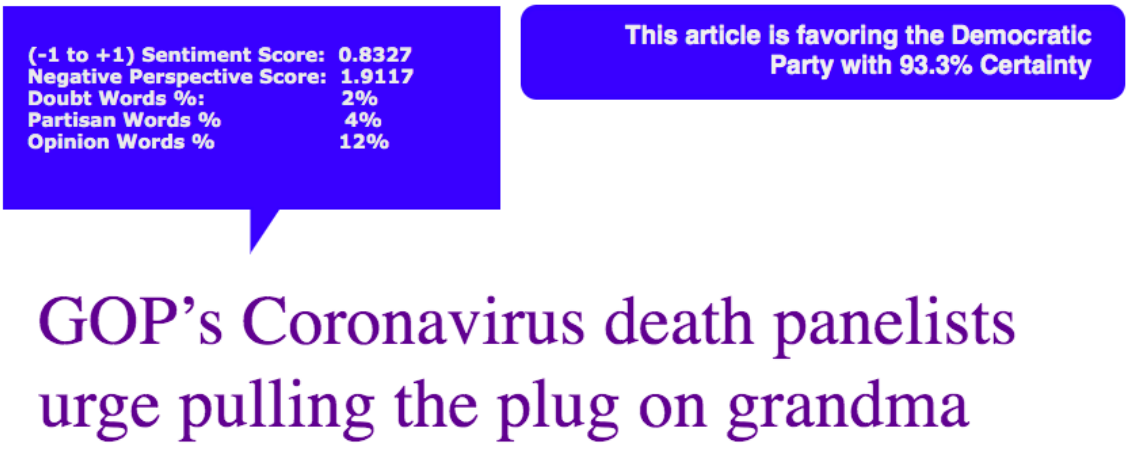


FIGURE 4.6: Democratic Party Article

Based on the Lexicon resources computation in the final algorithm. Every sentence has an associated score similar to how it is provided for the entire article.

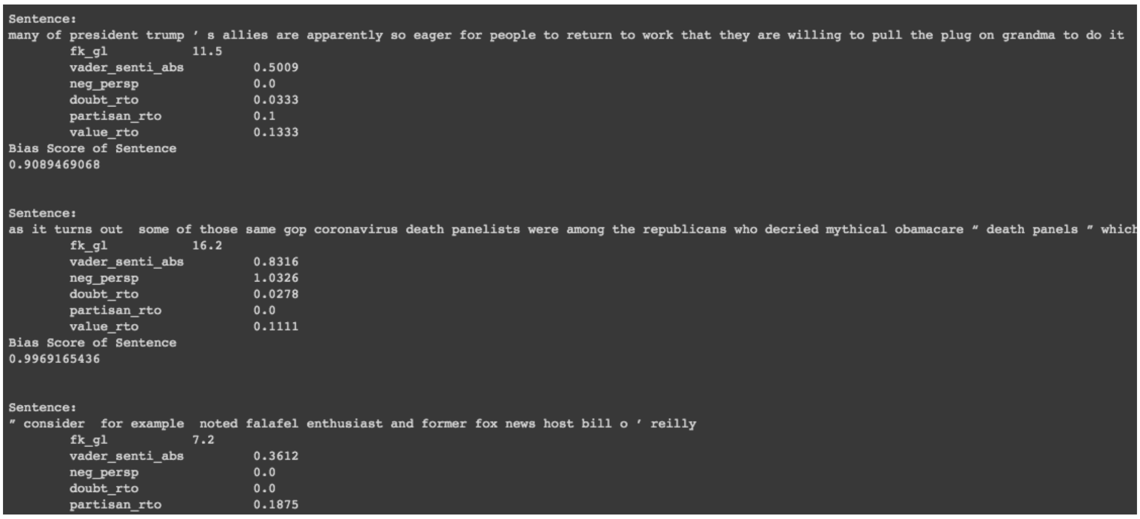


FIGURE 4.7: Sentence Level Analysis

Finally, one of the objectives was to assist users in understanding if the news outlet they are subscribed to or follow have any form of bias in their articles. However, since our dataset consists of only Breitbart and DailyKos articles we extracted the following differences in the features of both the articles. This was done by calculating the average score for each of the features in the articles.

	Breitbart	Daily Kos
Avg. Negative Perspective Score	2.172903704	4.675159807
Avg. Vader Sentiment	0.800798765	0.865100643
Avg. Doubt Words %	0.014942387	0.01682508
Avg. Partisan Words %	0.074125514	0.06696881
Avg. Value Words %	0.122079424	0.141667203
FK_GL	228.673251	284.8202572
Total number of articles	243	311

FIGURE 4.8: News Outlet Analysis

From the overall pre-analysis after looking at the number of Democrat and Republican favouring articles in both outlets, we can say that Breitbart is a Republican Favouring news outlet, as compared to DailyKos which supports the Democratic Party. This is because 88% of all articles in our dataset that are published by DailyKos are favouring the Democratic party and 91% of articles published by Breitbart are favouring the Republican Party. Furthermore, from the features provided we see that DailyKos has a higher negative perspective score as compared to Breitbart. Using our model, these values can be updated in real time to help users understand the overall bias score of each outlet.

4.4 Limitations

The following are a few limitations of the project that was implemented.

- Due to the inavailability of datasets with bias scores identified for articles. The current dataset is only limited to the topic of Politics related to Democratic and Republican parties in the US.
- The first model implemented which consists of sentences modelled by using Bidirectional LSTM and an attention layer, this model consists of weights that can help identify the strengths of each word towards the detection of bias. However, the project right now is limited to the lexicon based features analyzed on sentence level.
- To summarize and compare the bias levels in each outlet it is better performed given a larger dataset with more news outlets to compare. However, in this project, due to the inavailability of datasets labelled with bias the comparison is limited to only two news outlets.

Chapter 5

Conclusion and Future Work

In this section, we will look at an overview of the functional requirements that were fulfilled, followed by the conclusion of this paper, access to the source code, the challenges faced and the scope for future work.

5.1 Requirements Validation

We have successfully completed the Functional requirements that were prioritized from the first deliverable given in the figure below.

S.No	Description	Status
FR-1	**Data Collection based on Keyword**: Collect articles from news outlets based on a specific keyword provided.	***Done***
FR-2	**Encode News Articles**: Pre-Process and encode articles into a vector representation that can be used as input to the model.	***Done***
FR-3	**Compare Encoding Techniques**: Compare the precision of the model using different encoding techniques of the articles into vector-based representations as discussed.	***Done***
FR-4	**News Bias Score**: A percentage value describing the bias level of the article based on the intensity of bias words. Create a model to input the vector representation and identify associated bias score.	***Done***
FR-5	**News Outlet Analysis**: Process several articles from news outlets and compare the average bias score, to evaluate possible financial or ideological bias.	***Done***
FR-6	**Comparative Analysis**: Compare the baseline traditional models other researchers have compared against; models that have been used in other research papers with the model introduced in this project.	***Done***
FR-7	**Web Interface - Analysis**: Provide an interface to identify the sentence level bias score of an article	***Incomplete-Future Work***
FR-8	**Bias Articles dataset**: Create a dataset by web scraping numerous articles and their headlines from various news outlets with their associated bias score, to keep the door open for future work.	***Done***

TABLE 5.1: Status of Functional Requirements

5.2 Conclusion

Different researchers have chosen different paths when it comes to identifying bias. Some focus primarily on the sentiment of words, others take into account the deviation from articles on the same topic, while more complex techniques involve feature-extractions based on lexical resources that identify bias based on the structure of sentences.

Since the goal with this project was to allow readers to view the bias in articles before they read it. We have created a successful model by presenting the bias certainty towards a specific political party using the the Bidirectional LSTM with an attention layer. And we also effectively use the Bias-Lexical Resources computed to help users analyze the article on a sentence/headline and content level.

Additionally, we have created a dataset with headlines and content of articles web-scraped from an existing dataset that consisted of links provided by udak [2019]. This dataset can be used by researchers who wish to solve this problem with a different approach.

5.3 Project Source Code

Implementation Environment: The implementation was done in Jupyter Notebook using the Python Anaconda Navigator and on Google Colab. The source code provided is in the Python Programming language. This code can be used and modified without permission.

The file provided consists of a python script that can be viewed on Google Colab with the code associated with each of the implementations. The different methodologies can be easily accessed by using the table of contents to navigate. This file consists of python scripts for Web-Scraping, and another file comprising of the Exploratory Data Analysis and all the algorithms implemented.

5.4 Challenges

Overall, learning about Natural Language Processing was enriching, however, there were a few difficulties that we faced during the implementation of this project.

Data set Availability Issues:

Looking into the list of datasets that were available and labelled with Bias Levels based on our research in the first semester, the primary dataset we were hoping to use for our implementation (Buzzfeed, Facebook) was outdated and a majority of the articles were Unavailable since they were published in 2015-16. Hence, further research was required to identify a suitable labelled dataset. We faced a similar issue looking at the other available datasets, some were with unavailable articles while other datasets were generalized to the bias levels of News Outlets rather than specific articles.

Implementing Existing Models:
Some of the papers that were previously researched on the topic of Natural Language Processing and Bias detection did not provide code linked to their implementation. This added more research and implementation from scratch for the prior algorithms. Furthermore, some papers are not specific about their implementation, hence we were compelled to create these algorithms based on common practice.

5.5 Future Work

We have evaluated the performance of different algorithms in detecting the Bias in Political News Articles. However, additional features can be added to this model to improve the quality and presentation of the final output.

1. **Dataset Exploration**: For research purposes, the dataset collected in this project is specific towards Democrat and Republican articles. However, looking at the current situation with COVID-19 and the immense news coverage involved in these circumstances, various other articles can be scraped from news outlets and used in to train the model and identify bias levels.

2. **Recommendation System**: Similar to the implementation of the Open Mind software, this model can be used to recommend articles to users based on the bias levels of the articles they have previously read. This will help ensure users have a good recommendation of both sides of the argument on specific topics.

3. **User Interface**: An analytical web-interface as a Google Chrome Extension can be created to allow users to understand the topic that the article is favouring and provide more analytical information based on word usage and sentiment.

4. **Lexicon Based Features**: Identifying the usage of specific words that lead to controversy, negative speech, and other hedging lexicons based on newly available data sets will allow us to explore the possibility of bias due to specific word framing done by journalists.

5. **Sub-Topic Integration**: Allow readers to analyze the bias levels based on the average bias score associated with different news outlets given a certain topic as input.

Appendix A

Project Management

A.1 Requirements Analysis

A.1.1 Functional Requirements

Functional requirements are highlighted to understand the components and objectives of the project. An overview of the expected behavior and output of the project is contextualized. The evaluation of each of the Functional requirements is provided in the next table with a more in-depth Evaluation strategy given in the Project Implementation and Evaluation section.

A.1.2 Non-functional Requirements

Non-functional requirements specify the constraints that the program should follow and provide along with the functionalities.

A.2 Risk Management

It is important to reflect on the possible risks involved in the implementation of this project. Documenting the risks beforehand will ensure we stay on track during the implementation of the project regardless of the risk.
There are 4 steps involved when it comes to Risk Management:

1. **Risk Identification**: We understand the possible risks and their associated Risk Type. The Risk type can be one of the following:

Tools	People	Time	Requirements

2. **Risk Analysis**: Then we understand probability of the risk to occur(Likelihood) as well as the level of impact the risk will have if it does occur. The likelihood and impact levels of the risks are color co-ordinated as:

Color	**Likelihood**	**Impact**
	High	Catastrophic
	Medium	Serious
	Low	Tolerable

3. **Risk Planning and Monitoring**: Documenting the different levels of strategies in mitigating this risk and facing the circumstances strategically. Risk planning consists of 4 important aspects:

   - **Risk Monitoring and Avoidance**: Possible signs that help understand if the risk is likely to happen in the near future and reduce the chances of the risk to occur.
   - **Risk Contingency and Minimization**: Dealing with the threat associated with the risk and minimizing the impact that the risk has.

The following table covers all the aspects of Risk Analysis as discussed above

Risk Management		
	Planning and Monitoring	
Risk, Type **Likelihood, Impact**	**Monitoring** **and Avoidance**	**Contingency** **and Minimization**
Data Loss Type: **Tools** **Likelihood** **Impact**	**Monitoring**: Saving files takes a long time and minor parts are often lost. **Avoidance** Backing up data daily to Github and maintain proper version control.	**Contingency** Start the project all over again from scratch. **Minimize** Use the documentation prepared while building project and dedicate a day for completion without distrupting plan.

Data-Processing Type: **Time** **Likelihood** **Impact**	**Monitoring** Processing of articles for encoding takes more than 15 minutes, delaying the project plan.**Avoidance** Ensure code is optimized for data processing.	**Contingency** Ensure continuation of other tasks on another device. **Minimize** Use other less time consuming tools discussed for processing articles
Insufficient tools Type: **Tools** **Likelihood** **Impact**	**Monitoring** The tools planned are not compatible and cause errors on our specific problem, resulting in time-consuming unrelated work. **Avoidance** Ensure the tool has been used previously for our specific intent	**Contingency** Use alternative tools as planned. **Minimize** Ensure alternative software tools are mentioned in case of failure with the initially planned tool, ensure code is created compatible for easy change of tools.
Dataset informity Type: **Requirements** **Likelihood** **Impact**	**Monitoring** The originally planned dataset is not structured as required or with enough instances to train the model. **Avoidance** Perform data summarization of all the datasets before using them.	**Contingency** Use alternative datasets researched and ensure code is optimized to allow changing datasets easily. **Minimize** Ensure alternative datasets and web scraping methods are researched
Lagging in Project Plan Type: **Time** **Likelihood** **Impact**	**Monitoring** Not enough time dedicated to all tasks, rushing to completion. **Avoidance** Start early and stay ahead of the Project Plan.	**Contingency** Start implementing the next task after one is over before the start date to save time for later tasks and possible risks. **Minimize** Write a draft version of plan along the process and review, compile later.
Health and Deadlines Type: **People** **Likelihood** **Impact**	**Monitoring** The deadlines announced overlap with project plan. **Avoidance** Document the tasks completed the prior week to show in the meetings.	**Contingency** Ensure completion of tasks are planned according to possible deadline. **Minimize** Maintain a healthy diet and re-organize timeline according to other deadlines.

Creating existing models Type: **Requirements** **Likelihood** **Impact**	**Monitoring** The models researched are described vaguely making it difficult to implement and compare. **Avoidance** Focus primarily on the well-constructed models researched.	**Contingency** Modify the models with other similar models if not properly described and compare the accuracy mentioned in the research papers. **Minimize** Spend less time on these models to allow for more time on other well described models.

TABLE A.3: Risk Management

A.3 Project Implementation and Analysis

This chapter is a compilation of the ideation of the model to be implemented, the data pre-processing, software to use, and finally evaluation strategy.

A.3.1 Implementation Process

The following diagram presents the flow chart of the process involved during the implementation of this project.

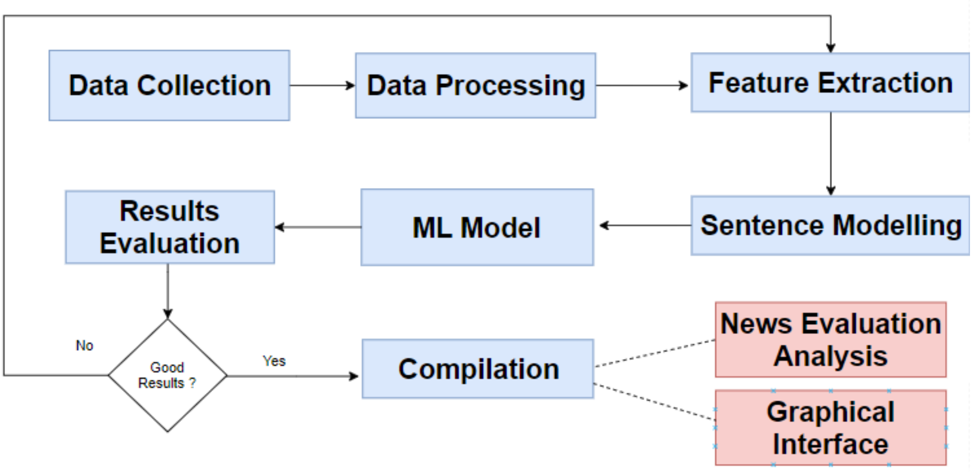


FIGURE A.1: Implementation Diagram

A.3.2 Implementation and Evaluation

Below is a summarization of the methodology and evaluation that will be carried out during the next semester.

1. **Data Collection**

   From the literature review, we see that most authors use web-scraping to manually extract articles based on a specific topic. A similar implementation will be done to collect articles from 4 different news outlets in the US, with its date, headline and the text of the article in the form of a dictionary on Python. The articles will be extracted considering highly controversial and non-controversial topics, this is because bias is more commonly present in topics of high controversy according to the hypothesis proven true in the research paper by Mejova et al.

   - **Testing**: Other datasets will also be tested on our final model including the "Unreliable News data" corpus introduced in the literature review, which provides articles tagged "Propoganda" having ideological views to mislead users.
   - **Evaluation**: Based on the testing on different datasets, the results will be compared to ensure our model is not overfitting the scraped articles. This evaluation will be done using a single traditional model such as Naive Bayes with 10-Fold Cross-Validation on different datasets and our extracted articles. The articles are acceptable if the results of the model are similar to that of other datasets.

2. **Data Processing**

   Basic Tokenization, removal of punctuation and lowercasing sentences will be used to process the articles and headlines.

   - **Testing**: The representation should be tested to ensure that it does not cause issues when inputting into the model. Applying Stop word removal will also be tested to evaluate if this affects our model negatively or increases the accuracy due to the fewer words used.
   - **Evaluation**: A sentence modeling technique will be applied to ensure the tokenization is valid and the sentences are contextualized properly. Furthermore, a comparative analysis will be performed to identify if removing stop words effects the results of the final bias scoring model.

3. **Sentence Modelling**

   The tokenized articles and their headlines will be converted into vector representations that take the context of the sentences into consideration.

   - **Testing**: Various feature-based, and machine learning models will be applied to compare and understand which model gives the best results.
   - **Evaluation**: Applying different combinations of the sentence modeling techniques the best process will be documented and used for our model.

4. **Model Creation**

   We notice that Gangula et al. uses a Bidirectional LSTM that ensures the context of sentences are properly vectorized but the model and the Lexical Resources for feature extraction introduced by Recasens et al. identifies the bias levels and assigns a weight to each word based on how dramatic and ambiguous the sentences are. Furthermore, the model introduced by Quijote et al. uses Inverse reinforcement learning considering the deviation of the articles compared to other articles on the same topic. Integrating these three different methodologies into a single model will allow us to ensure (1) The deviation of articles is considered (2) The context and relationship of words with other words in sentences is considered and finally (3) The bias weights assigned to each word based on Lexicon based feature extractions are considered.

   - **Testing**: Testing this again other combinations of sentence modeling and machine learning models will be done to see if better results are given by other methods. The results will also be compared to other researched methodologies.
   - **Evaluation**: To evaluate the effectiveness and performance of this model Precision, Recall and accuracy will be used to compare against the traditional baseline model commonly used by other researchers to compare their models against, such as Random Baseline Model, Linear Regression Model, etc. In the end, the results will be provided in the form of comparative analysis.

A.4 Development Methodology

This project will follow an Agile based **Iterative and Incremental model**, which is a feature-driven methodology allowing us to interleave different aspects of testing, implementation and evaluation together following an iterative process in which a single functionality is produced or an existing functionality is modified in the span of a week based on the meeting with the supervisor. At the end of each week, we will have a working functionality complete with the evaluation, testing and the steps documented.

This **Agile development methodology** will help ensure the functional requirements are evolving in an iterative process allowing for different functionalities. This model is advantageous due to the following reasons:

1. It helps deliver functionalities faster and ensure different components of the project are independent of each other.
2. Specific features of the model are dedicated an entire iteration for implementation,evaluation and testing.
3. At the end of each week a functionality is completed, in contrast to, completing the project all at once resulting in the tasks getting crammed up towards the end.

.

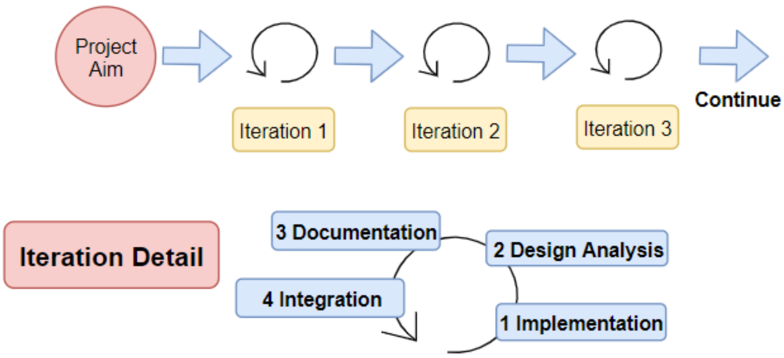


FIGURE A.2: Iterative and Incremental Model

A.5 Project Plan

In this section, we look at the Project Plan for the next semester covering the tasks we will be completing in each iteration. This project timetable was created using TeamGantt, Trello boards will also be used to ensure progress every week.

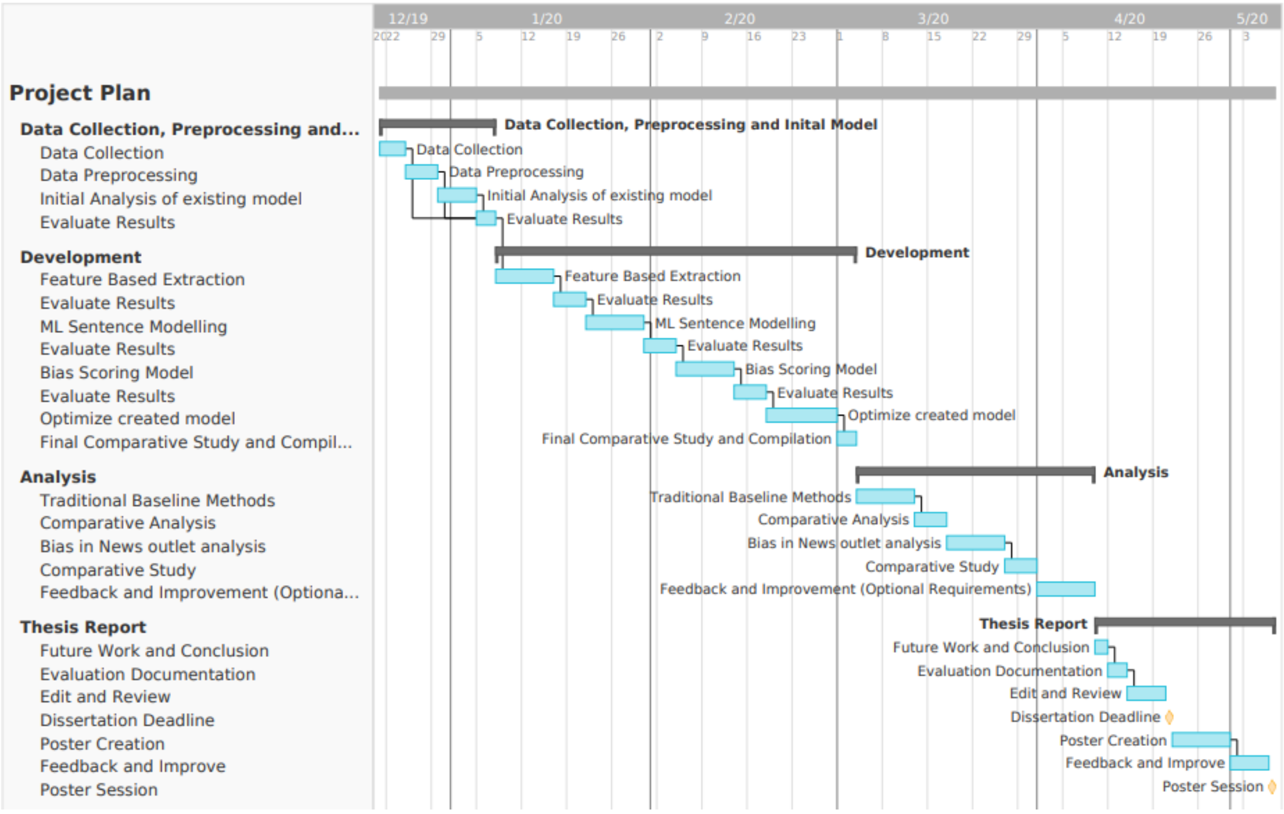


FIGURE A.3: Project Timetable

Functional Requirements		
S.No	**Description**	**Priority**
FR-1	**Data Collection based on Keyword** Collect articles from news outlets based on a specific keyword provided.	**M**
FR-2	**Encode News Articles** Pre-Process and encode articles into a vector representation that can be used as input to the model.	**M**
FR-3	**Compare Encoding Techniques** Compare the precision of the model using different encoding techniques of the articles into vector-based representations as discussed.	**S**
FR-4	**News Bias Score** A percentage value describing the bias level of the article based on the intensity of bias words. Create a model to input the vector representation and identify associated bias score. This model should integrate commonly used methods identified through research.	**M**
FR-5	**News Outlet Analysis** Process several articles from news outlets and compare the average bias score, to evaluate possible financial or ideological bias.	**S**
FR-6	**Comparative Analysis** Compare the baseline traditional models other researchers have compared against, models that have been used in other research papers with the model introduced in this project.	**M**
FR-7	**Web Interface - Analysis** Provide an interface to identify the overall bias score of an article, analyze the sub-topics vs bias levels, gradient highlighting to present the bias levels of specific sentences and recommend unbiased articles.	**C**
FR-8	**Bias Articles dataset** Create a dataset by web scraping numerous articles and their headlines from various news outlets with their associated bias score, to keep the door open for future work.	**C**

Evaluation Strategy	
FR-1	**Data Collection based on Keyword** The articles extracted will be evaluated based on the exactness and completeness of the text and headline.
FR-2	**Sentence Modelling** Different combinations of sentence modelling techniques will be applied and the results will be compared.
FR-3	**Compare Sentence Modelling Techniques** The evaluation will be done on the results of FR-2 based on Precision, Recall and Accuracy.
FR-4	**News Bias Score** The model created in this project will be compared to other baseline models like Logistic Regression, Naive Bayes, SVMs and CNN. Furthermore, models implemented in research papers will also be implemented and compared to.
FR-5	**News Outlet Analysis** The results will be compared against analyses provided online for news bias in News outlets (refer to News Analyses in Literature Review)
FR-6	**Comparative Analysis** Performance, Accuracy and Effectiveness using various News Bias scoring + Sentence Modelling Techniques will be documented and compared with our model.

Table A.1: Functional Requirements and Evaluation Strategies

Non-Functional Requirements		
S.No	**Description**	**Priority**
NFR-1	**Cross-Platform** The Python Script developed should be cross-platform and easily accessible on different Operating Systems. A jupyterNotebook script should be published, Jupyter is an IDE that is flexible allowing comments and code for easy organization.	**M**
NFR-2	**Accessibility and Version Control** The code should be open-source and published on GitHub to allow other people to view and build improved versions. All code versions should be organized using Version Control.	**M**
NFR-3	**Instructions Guide** The code should be well commented and an instruction guide should highlight the steps to run the code.	**M**
NFR-4	**Web Interface** The web interface for readers to view bias levels in articles should be easy to use and intuitive, providing an analysis that is easy to understand and access. The interface should be user-friendly with responsive features.	**C**
NFR-5	**Scalability** The code should be designed to allow for future work and improvements. Additionally, the report should be structured to provide a good overview of past and most recent work, to encourage researchers to test their own versions of the model.	**S**
NFR-6	**Cohesiveness** Important components of the system should be structured independent from each other in the code and report, ensuring researchers can access and run specific components based on their interests rather than running the entire program.	**S**
NFR-7	**Reusability** The program developed should be built with the idea that components could be reused by other developers. This means comments and explanations should be including along with the code.	**S**

Table A.2: Non-Functional Requirements

Appendix B

Discussion of Professional, Legal, Ethical and Social Issues

B.1 Professional and Legal Issues

Research papers that have been used will be properly referenced. If code from other sources are used it will be documented and acknowledged.
The source code is published under a GNU General Public License as follows:
This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation,version 3.
This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantibility or fitness for a particular purpose. See the GNU General Public License for more details.

B.2 Ethical and Social Issues

This thesis does not involve testing with human subjects, personal or confidential data. The dataset used does not include personal or confidential data, it consists of articles that are publicly available on online news websites, therefore, does not require permission for use. The news websites that are used will be properly referenced and acknowledged. No risk is involved since all the data used is freely available online.
In regards to this project, readers exposed to the bias score on articles they are reading will be

able to understand the intentions behind the article helping them make an informed decision on the view they wish to support. Deceptive news is targeted at people who are easy manipulated but with this project, people can be more careful and adamant about their views. In conclusion, this will help the society connect with each other, be on better terms, and respect each others opposing views.

Bibliography

BuzzFeedNews (2016). Fact-checking facebook politics pages dataset. url-https://github.com/buzzfeednews/2016-10-facebook-fact-check. *BuzzFeedNews*.

Chan, R. (2019). The cambridge analytica whistleblower explains how the firm used facebook data to sway elections. *Business Insider*.

D'Alessio, D. and Allen, M. (2000). Media bias in presidential elections: A meta-analysis. *Journal of communication*, 50(4):133–156.

Enevoldsen, K. and Hansen, L. (2017). Analysing political biases in danish newspapers using sentiment analysis. *Journal of Language Works - Sprogvidenskabeligt Studentertidsskrift*, 2(2):87–98.

FakeNewsCorpus (2017). Fakenewscorpus dataset using opensource.co. url-https://github.com/several27/fakenewscorpus. *FakeNewsCorpus*.

Gangula, R. R. R., Duggenpudi, S. R., and Mamidi, R. (2019). Detecting political bias in news articles using headline attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 77–84, Florence, Italy. Association for Computational Linguistics.

Hamborg, F., Donnay, K., and Gipp, B. (2018). Automated identification of media bias in news articles: an interdisciplinary literature review. *International Journal on Digital Libraries*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Hore, P. (2019a). Attention mechanism deep learning. *Analytics Vidhya*.

Hore, P. (2019b). Attention mechanism deep learning. *Pelatrion*.

Hutto, C., Folds, D., and Appling, D. (2015). Computationally detecting and quantifying the degree of bias in sentence-level text of news stories.

Iyyer, M., Enns, P., Boyd-Graber, J., and Resnik, P. (2014). Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1113–1122.

Jain, S. (2018). An overview of regularization techniques in deep learning. *AnalyticsVidhya*.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

Lauw, H., Lim, E.-P., and Wang, K. (2006). Bias and controversy beyond the statistical deviation. In *Bias and controversy beyond the statistical deviation.*, pages 625–630.

Lazaridou, K. and Krestel, R. (2016). Identifying political bias in news articles. *Bulletin of the IEEE TCDL*, 12.

Mejova, Y., Zhang, A. X., Diakopoulos, N., and Castillo, C. (2014). Controversy and sentiment in online news. *arXiv preprint arXiv:1409.8152*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mullainathan, S. and Shleifer, A. (2002). Media bias. Working Paper 9295, National Bureau of Economic Research.

Nagpal, A. (2017). L1 and l2 regularization methods. *Towards Datascience*.

Norregaard, J., Horne, B. D., and Adali, S. (2019). NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles. *CoRR*, abs/1904.01546.

Patankar, A. A., Bose, J., and Khanna, H. (2018). A bias aware news recommendation system. *CoRR*, abs/1803.03428.

Phalip, J. (2017). Identifying bias in the media with sentiment. *Julien Phalip Blog*.

Purnasai, G. (2019). Recurrent neural networks and lstm explained.

Qaiser, S. and Ali, R. (2018). Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181.

Quackenbush, D. (2013). Public perceptions of media bias: A meta-analysis of american media outlets during the 2012 presidential election. In *Public Perceptions of Media Bias: A Meta-Analysis of American Media Outlets During the 2012 Presidential Election.*

Quijote, T. A., Zamoras, A. D., and Ceniza, A. (2019). Bias detection in philippine political news articles using SentiWordNet and inverse reinforcement model. *IOP Conference Series: Materials Science and Engineering*, 482:012036.

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., and Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937.

Recasens, M., Danescu-Niculescu-Mizil, C., and Jurafsky, D. (2013). Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659, Sofia, Bulgaria. Association for Computational Linguistics.

Reshamwala, A., Mishra, D., and Pawar, P. (2013). Review on natural language processing. *IRACST – Engineering Science and Technology: An International Journal (ESTIJ)*, 3:113–116.

Sherstinsky, A. (2018). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314.

SV. Shri Bharathi, A. G. (2019). Determination of news biasedness using content sentiment analysis algorithm. *Indonesian Journal of Electrical Engineering and Computer Scinece.*

Tandoc Jr, E. C., Lim, Z. W., and Ling, R. (2018). Defining "fake news" a typology of scholarly definitions. *Digital journalism*, 6(2):137–153.

Tausczik, Y. R. and Pennebaker, J. W. (2010). The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

Trends, G. (2019). Explore what the world is searching with google trends. https://trends.google.com/trends/?geo=us. *Google Trends.*

udak, C., G. S. R. J. (2019). Quantifying news media bias through crowdsourcing and machine learning dataset. university of michigan - deep blue.

Waldrop, M. M. (2017). News feature: The genuine problem of fake news. *Proceedings of the National Academy of Sciences*, 114(48):12631–12634.

Wardle, C. (2017). Fake news, it's complicated. *First Draft*.

White, D. M. (1950). The "gate keeper": A case study in the selection of news. *Journalism Bulletin*, 27(4):383–390.

Zhang, J., Kawai, Y., Nakajima, S., Matsumoto, Y., and Tanaka, K. (2011). Sentiment bias detection in support of news credibility judgment. In *SBDSNC*, pages 1–10.