Master of Science

석사학위논문

IMPROVING DETECTION OF SUBJECTIVE BIAS USING BERT AND BILSTM

BERT 및 양방향 LSTM을 사용한 주관적 치우침 탐지 개선

울산대학교 대학원
컴퓨터정보통신공학과
Ebipatei Victoria Tunyan

IMPROVING DETECTION OF SUBJECTIVE BIAS USING BERT AND BILSTM

Supervisor: Prof. Cheol-Young Ock

A Dissertation

Submitted to
the Graduate School of the University of Ulsan
In Partial Fulfillment of the Requirements
for the Degree of

Master of Science

by

Ebipatei Victoria Tunyan

Department of IT Convergence
Ulsan, Korea
Jun 2021

Ebipatei Victoria Tunyan의 공학석사 학위논문을 인준함

심사위원장 권 영 근 (인)

심사위원 옥 철 영 (인)

심사위원 정 진 호 (인)

울산대학교 대학원

2021년 6월

Abstract

The task of detecting subjectively biased statement is critical. This is because bias in text or other types of knowledge delivery media, such as news, social media, science texts, and even encyclopedias, can erode consumer confidence in the information and trigger conflicts. Subjective bias detection is vital for many Natural Language Processing (NLP) tasks like sentiment analysis, opinion identification, and bias neutralization. Having a system that can adequately detect subjectivity in text would noticeably aid research in the aforementioned fields. It can also be useful for platforms such as Wikipedia, where the use of neutral language is critical. The aim of this thesis is to identify subjectively biased language in text, not just at the sentence level but also at document levels.

With deep learning, we can solve complex AI problems, making it a good fit for the problem of subjective bias detection. Training a classifier based on BERT (Bidirectional Encoder Representations from Transformers) as an upstream model is an essential factor in this approach. BERT may be used as an all-round classifier on its own; however, in this research, it is used as a data preprocessor and embedding generator for a Bi-LSTM (Bidirectional Long Short-Term Memory) downstream model with an attention mechanism. This method yields a more accurate and comprehensive

classifier. I assess the efficacy of the proposed model by comparing it to current methods using the Wiki Neutrality Corpus (WNC), which was compiled from Wikipedia edits that excluded myriad biased instances from sentences as a benchmark dataset. Our model attained state-of-the-art (SOTA) performance (sentence-level accuracy of 89% with F1 of 90% and document-level accuracy of 89% with F1 of 91%) in identifying subjective bias, per the results of our experiments. This model may be fine-tuned to support other languages, as this analysis focuses on English language.

Keywords: Subjective Bias Detection, Machine Learning, BERT–BiLSTM–Attention, Text Classification, Natural Language Processing.

Acknowledgments

I am very grateful to the Korean government for awarding me the Global Korean Scholarship to carry out my Master's research in Korea. The scholarship program accorded me the opportunity to not only complete my research with industry and academic professionals such as my advising professor, but to also learn the Korean language.

I would like to express my sincere gratitude to my research advisor, Professor Ock, Cheol-Young, for his expert advice and friendly guidance throughout my Master's degree program at the University of Ulsan. He gave me the opportunity of research under his supervision and never hesitated to support me where necessary. Especially during the execution of my experiments, his encouragement and guidance were instrumental to my success.

In addition, I would also like to extend my sincere thanks to the other professors in my thesis committee, Professor Kwon, Yung-Keun, and Professor Chung, Jin-Ho for dedicating their time to review this thesis. Their insightful comments were beneficial contributions to the completion of this work.

I am grateful to all my lab mates for their kindness and ever available assistantship with the Korean language whenever I struggled. They helped me get familiar with life in Korea, and we often shared interests about life and research. I also must acknowledge the academic, and technical support from staff of the University of Ulsan.

Lastly, I am very thankful to my loved ones, family, and friends for their continuous love, support, faith, and encouragement throughout my studies.

Dedication

This work is first and foremost dedicated to God Almighty. You oh Lord are my strength! You gave me not just the strength but the resources to complete this chapter of my life.

And it is also dedicated to whomever finds satisfaction in the fulfillment of this work.

Contents

List of figures

List of tables

List of abbreviations

NLP	Natural Language Processing
ML	Machine Learning
NPOV	Neutral Point of View
WNC	Wiki Neutrality Corpus
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
GRU	Gated Recurrent Unit
RNN	Recurrent Neural Network
Word2vec	Word to vector representation
Glove	Global Vectors for Word Representation
GPU	Graphics Processing Unit

Chapter 1

Introduction

1.1. Motivation

The presence of subjective bias in information content is a big challenge in all types of media, especially the news media. Though writers & editors endeavor to avoid biased language in their content, bias is yet pervasive in these contents. Much more than other forms of bias, we see subjective bias more prevalent. When writers put out information content, more often than not we notice how subjective information that is supposed to be objective have become, though they may appear subtle. By definition, Subjective bias occurs when personal feelings or opinion are stated as fact, whether consciously or unconsciously. Similarly, expressing general fact can be seen as objective. For example, [1] the first statement quoted below is biased because the phrase *disappears from* presents the writers personal opinion (perceived reality) in the sentence, rather than reality itself. Whereas the second statement is neutral. This is because it does not contain any person opinions of the writer and can be confirmed as a fact that indeed such a meeting took place.

[1] https://www.africanews.com/

"Jack Ma disappears from African TV Show fueling whereabouts questions"

"Botswana-China Talks to strengthen bilateral relations and cooperation between the two nations"

Ambiguity of the English language, human error, among other factors have made it increasingly difficult to identify these biases in their subtlety. With intelligent systems, we can mitigate some of these issues, such as the human error that writers and editors face as they strive to avoid bias. Previous attempts have been made to solve this problem, but these systems suffer performance inadequacies. These factors contribute to my motivation to create a system that will not only identify and classify subjective bias on the sentence level, but also on the document level with state-of-the-art performance. The work presented in [1] also inspired the development of this project. My aim is to improve the accuracy obtained in their work.

1.2. Problem Statement

Media bias continues to gain the interest of many as one can often find at least one kind of bias in information from various media. A myriad of biases exists in our world today, ranging from gender bias, to racial bias, to religious bias, etc. And these biases usually come off in a subjective sense. Hence the need to detect and subsequently mitigate subjective bias from our media space. Writers and content editors in their own way, make efforts to avoid bias in their contents but human error is sometimes unavoidable. However, with the help of machine learning, we can curb this challenge. Using deep learning techniques in detecting bias, trained deep neural networks can catch these often unnoticed human errors.

Moreover, as reported by Gallup news [2], a large percentage of adult Americans believe that up to 62% of the traditional news media and up to 80% of social media news is not objective. These high figures drive the need for a solution. It indicates that it has now become imperative to differentiate subjective language from neutral language. Therefore, having a system like the one proposed in this work to detect subjective bias will go a long way in solving this problem.

1.3. Existing Solutions

Attempts have been made by industry professionals and researchers alike to solve the problem of identifying subjectively biased text in the past. And through these attempts, several approaches have been proposed. Some of these approaches are included in the following.

- Statistical methods
- Linguistic feature-based methods
- Glove-based methods
- Word2vec-based methods
- Fasttext-based methods
- BERT and BERT ensemble-based methods

Identifying subjectivity is an important aspect of bias detection, which has expectedly caught the interest of many researchers, who have gone ahead to publish their findings. We find from these that more efficient techniques armed with larger datasets are needed to precisely gain better performance.

1.4. Research Objectives

Many find it difficult trusting the content of the news as objective [3]. I seek to change this narrative. Hence, the goal of this study is to automatically identify the subjectively biased language in text (i.e., the neutral point of view (NPOV) violations) on the document level. A system that can automatically identify the subjectivity of a document or its absence thereof (neutrality) will be a valuable tool for editors analyzing huge collections of documents for NPOV violations. With the use of deep learning, we can better and further solve complex AI problems such as this.

To achieve this objective, a benchmark dataset known as the Wiki Neutrality Corpus (WNC) was used. WNC is a dataset that was put together by [4] in accordance with the Wikipedia's neutral point of view (NPOV) policy[2] . This policy is one of three Wikipedia policies that guide Wikipedia writers. The dataset consists of over 360k

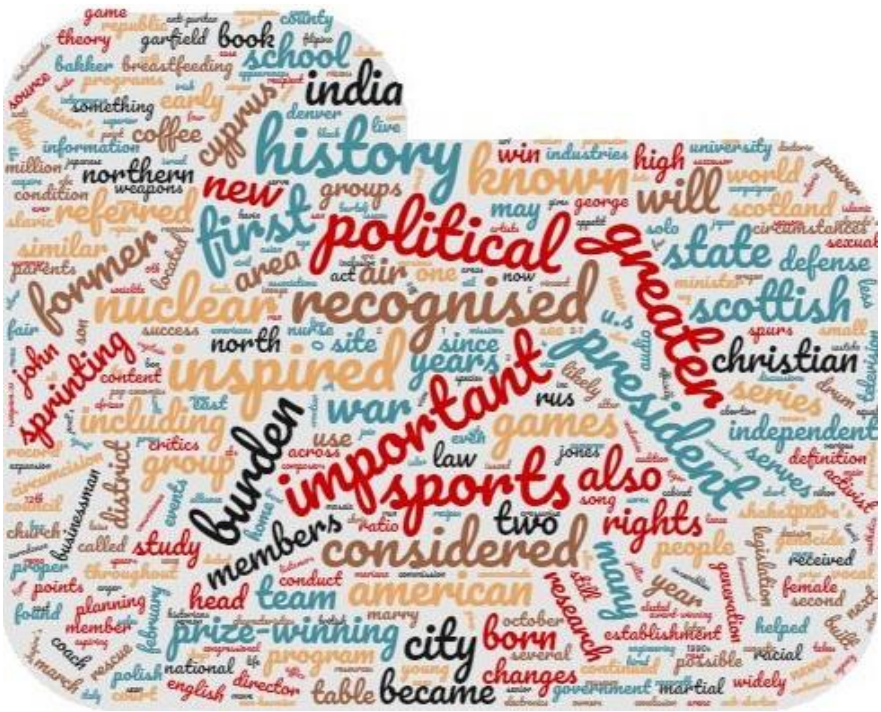

Figure 1. WNC Word Cloud.

[2] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

sentences by English Wikipedia editors crawled from 423,823 Wikipedia revisions over a 15year period. Analyzing the dataset, I observed that subjective bias is more prevalent in areas such as politics, sports, geography, and history than others. The word cloud in Fig.1 gives an insight.

1.5. Thesis Outline

This thesis is composed of five chapters, outlined as follows:

Chapter 1, which is the current chapter, introduces the subject of the thesis. It states the problem and motivation towards solving the stated problem. It further outlines existing solutions to the problem and our objective for conducting this research

Chapter 2 presents an overview and review of literature on bias detection. In this chapter I discuss some of the existing approaches to bias detection and works related to subjectivity detection that have been carried out in the past.

Chapter 3 describes the proposed methodology for detecting subjective bias on the document level. It gives a detailed breakdown on each component that makes up the proposed model, and how it all comes together to form a deep neural network that efficiently classifies document level subjective texts.

Chapter 4 reports on the experiments carried out, and the results realized from these experiments. In this chapter, we will see how the proposed model performs in comparison to existing baselines, for both sentence level and document level classification.

Chapter 5 concludes this thesis with a summary of the work done in the course of the research and provides directions for future studies.

Chapter 2

Backgrounds

2.1. Bias Detection Overview

Bias in terms of information can be defined as the deviation of information from the truth, or the processes leading to such deviations. In the field of Natural Language Processing (NLP), Bias detection can be regarded as a text classification problem. Text classification, also known as text categorization or text tagging is the process of categorizing text into organized groups. Using NLP, text classifiers can automatically analyze text and assign pre-defined tags or categories to the text, based on its content. Over the years, many NLP techniques have been used to create bias detection models, ranging from statistical methods to traditional machine learning methods, and more recently, deep learning techniques. These approaches have at different points produced state-of-the-art performances on various tasks (including the subject matter), as one approach is an improvement on the other. In this thesis, I used the deep learning technique as deep learning gave the best performance at the time of writing.

In this research, subjective bias detection is a binary text classification task. The output of a binary classification model is usually one of two classes; zero or one (or can

be approximated as such), positive or negative, or subjective or neutral/objective, as is our case in this study.

2.2. Related Works

Considerable work has been carried out on identifying subjectivity using several text classification techniques. It is no wonder, as text classification is a fundamental aspect of Natural Language Processing (NLP). While some of these techniques are based on statistical methods, a majority use machine learning methods for text classification. And some (such as the proposed model) go a step further with deep learning. The task presented is a binary classification task that should give either a subjective or objective class as output. Recasens, Danescu-Niculescu-Mizil, and Jurafsky [5] use a logistic regression-based model with linguistic feature to achieve this. They designed their model to detect the bias-inducing words in a statement. This method follows the NPOV policy which the authors introduced for this task. Pryzant et al. [4] follows a similar pattern as it builds upon the approach of [5]. Tanvi Dadu, Kartikey Pant, Radhika Mamidi [6] propose the use of contextualized word embeddings to achieve this task on a sentence level. They conclude that an ensemble of these contextualized embedding models produces a higher accuracy as opposed their single counterparts. Although this comes as an improvement on the single word detection method of [4] and [5], the accuracy they present, I believe can be improved upon.

Similarly, on the document level, a good number of research has been conducted for text classification. Yang et. al. [1] implemented a hierarchical network which utilizes BiGRU [7] and attention for document classification. The authors seeded the embedding layer with Glove embeddings [8]. This approach produced decent results but could be better, which is what contributed to the motivation of this study, where I

rather used BERT [9] to initialize the model for better performance. Authors Xiaoming Shi and Ran Lu [10] executed an approach similar to that of [1]. However, they replaced the BiGRU with BiLSTM [7] that resulted in a slightly better performance.

Several other models exist for detecting subjectivity in text data. The authors in [11] investigated major semi supervised learning methods for identifying opinionated sentences. Riloff and Wiebe [12] focused on a bootstrapping algorithms for sentence level subjectivity detection. They argue that since the subjective and objective expression patterns are based on syntactic structures, they provide more flexibility than single words or n-grams. Furthermore, they propose a dataset called the MPQA Opinion Corpus, which is a dataset containing about 5,000 subjective and 5,000 objective sentences. Compared to the dataset used in my proposed model, this dataset is much smaller. Authors in [13] - [15] proposed models that use other word embedding methods such as Word2vec [16], Glove [8], and fastText [17] to get vector representations of their input data. In this work, I utilized the BERT [9] contextualized embeddings to generate vector representations. This approach produces better performance as it takes into consideration the context of the input text. Last but not the least, [18] and [19] present classification models based on BERT – Bi-LSTM. I have extended this by further adding an attention mechanism to capture the importance of the words in a sentence based on context.

Chapter 3

Methodology

For the task of subjectivity detection, I propose a deep neural network comprising of three components, a BERT model [9], a Bi-LSTM model [11], and an attention mechanism [20] (i.e., BERT + Bi-LSTM + Attention). This section gives detailed description of each component, and how they integrate to achieve the proposed model. Note that this approach adopts BERT as the upstream of the model and Bi-LSTM with attention as downstream of the model.

The presented method builds upon the approach of Tunyan et al. [21], a sentence classifier that also uses BERT as the model upstream. I have taken it further towards document classification. On the sentence level, the model identifies bias inducing words and phrases alike in the sentence, whereas the document level model classifies a document as subjectively biased based on the biased sentences in the document. Figure 2 and 3 highlights the differences in the model architecture of the sentence level classifier and the document level classifier.

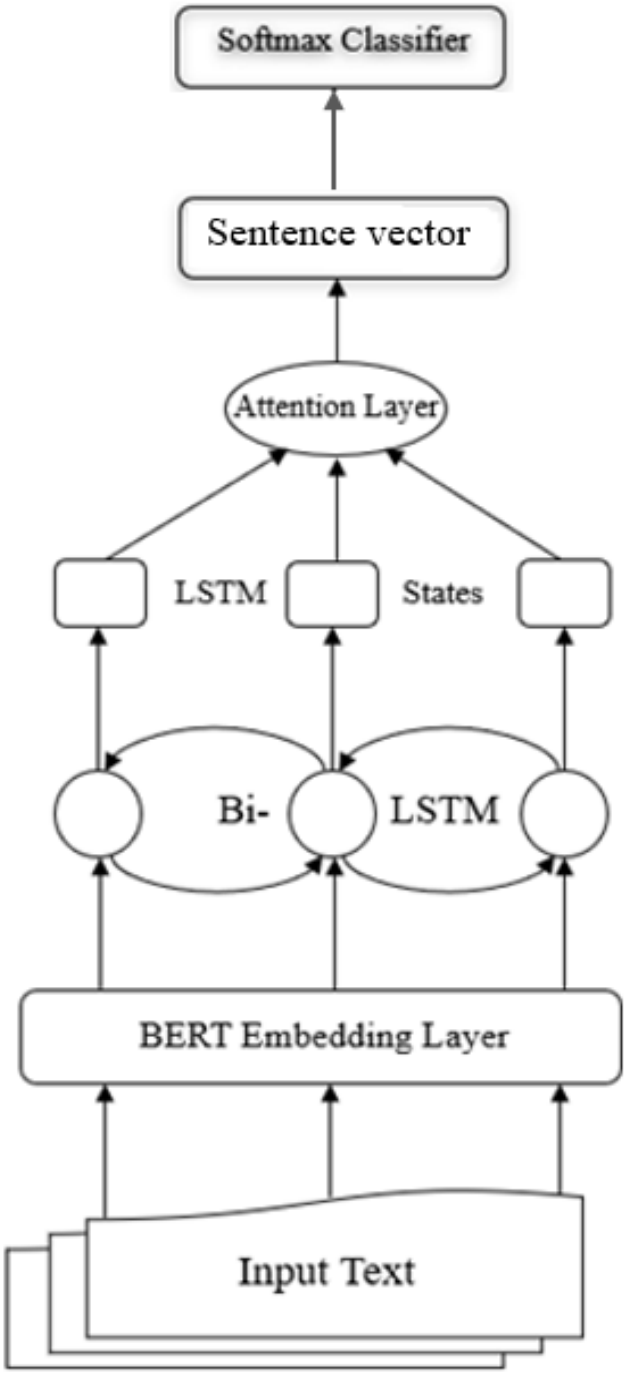

Figure 2. Sentence Level Model Architecture.

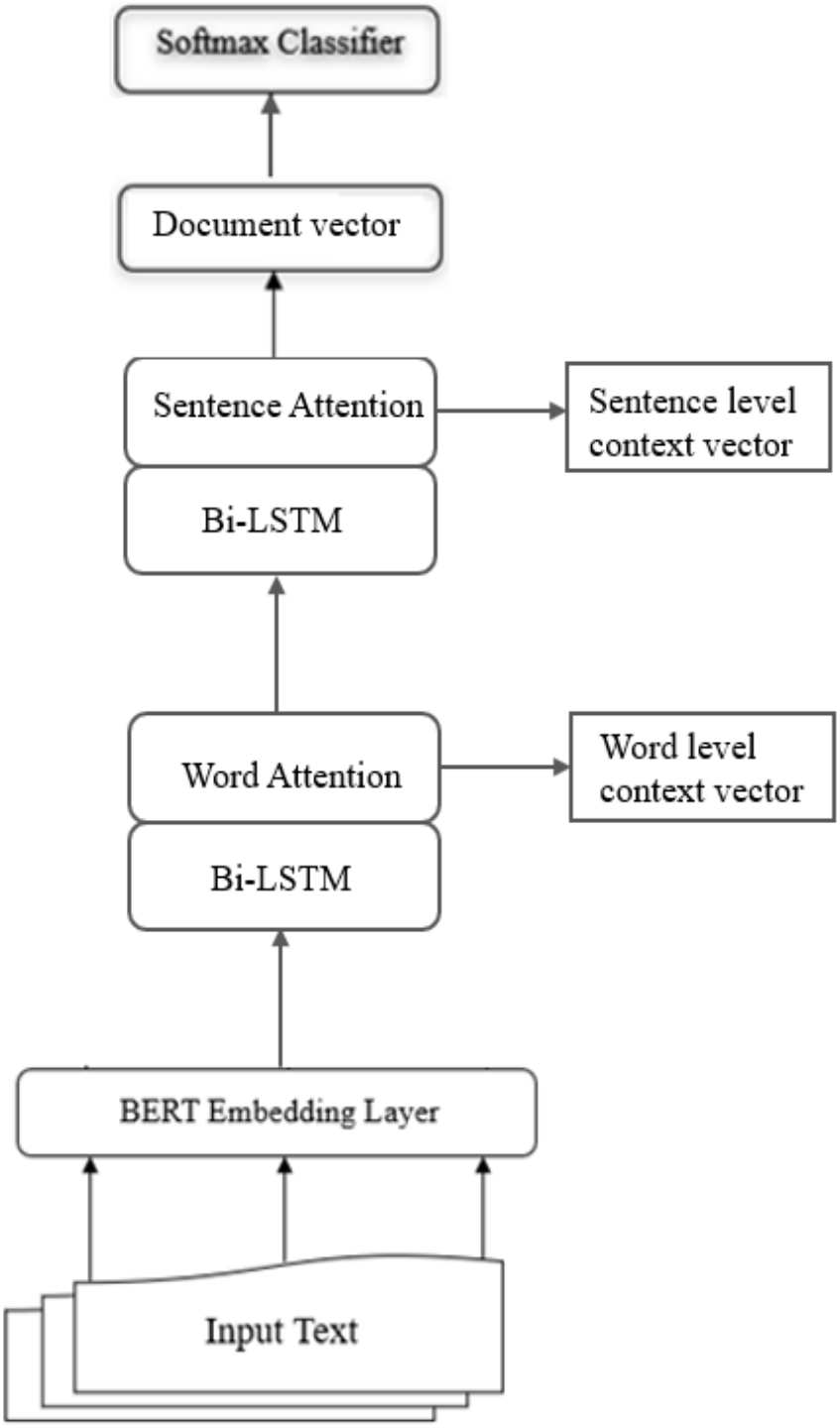

Figure 3. Proposed Document Level Model Architecture.

3.1. BERT

BERT is a pretrained contextualized text representation model that uses the bidirectional transformer network structure, as proposed by Devlin et al. [9]. Because of its bidirectional nature, it can take into account both directions of words in a sentence for context. It was pre-trained on a corpus of more than 3 billion words. BERT has a number of advantages over other approaches, making it a better fit for the task at hand. BERT excels at detecting the meaning of a language sequence based on context because of its contextual nature. It can understand subtle variations in phrasing because of this advantage. A remarkable feature of BERT is that merely using the BERT model and fine-tuning it can generate relatively good results, although building upon it gives even better performance. Another major benefit of BERT is that it requires significantly less data preprocessing compared to existing methods. I used the BERT base model version since it is smaller and takes lesser time to process.

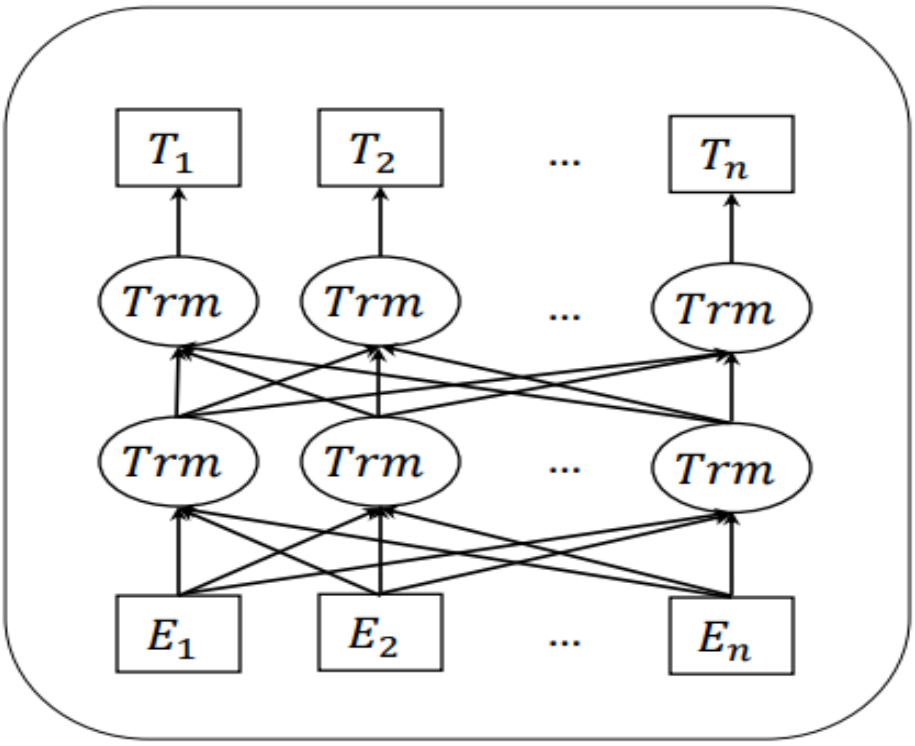

Figure 4. BERT Structure [21]

Given an input sentence sequence representing a document, which consists of w_{it} words of length T as in w_{it}, $\mathrm{t} \in [1, \mathrm{T}]$, the first step after the model accepts this input is preprocessing using BERT's tokenizer. It tokenizes the text and maps each token into a unique ID. These token IDs are then accepted into the BERT model in the second phase of the network. This phase embeds the words to vectors through BERT. Hence, the output of the of the BERT layer are highly contextualized vector embeddings of 768 dimensions for each word.

$$x_{it} = BERT(w_{it}),\ \mathrm{t} \in [1, \mathrm{T}] \tag{1}$$

3.2. Bi-LSTM

3.2.1. Overview

Requisite to discuss Bi-LSTM is an understanding of RNN (Recurrent Neural Network) [22], LSTM (Long Short-Term Memory) [23], and GRU (Gated Recurrent Unit) [24], and their relationship as it relates to Bidirectional LSTM. Both LSTM and GRU are classes or variations of RNN. And BiLSTM as well as BiGRU are bidirectional variations of RNN, also known as BRNN [7]. It is composed of two RNNs, one in the forward direction and the other in the backward direction. The following sub-sections give a brief but encompassing understanding of each class.

3.2.2. RNN

In NLP, dealing with majority of text data require sequential processing, and regular feedforward neural networks do not handle sequences properly. RNN is a special kind of neural network, which can at a high level process sequences of data one element at a time by keeping a memory state. The idea behind memory state is to keep track of

previous inputs and computations to perform a given task on every element dependent on all previous computations. That is, the memory state simply keeps track of what has come previously in the sequence. RNN is designed to mimic the human way of processing sequences, in that we consider the entire sentence when forming a response and not just individual words.

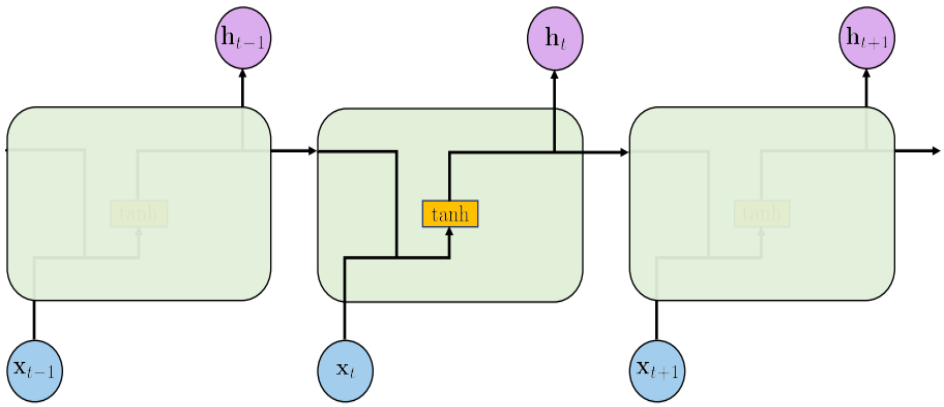

Figure 5. RNN Cell [25]

The term recurrent in this context means the output at the current time step becomes the input to the next time step. At each time step, the model considers not just the current input element, but also what it remembers about the preceding elements. Figure 6 shows this in an unrolled recurrent loop. And RNNs hidden state can be calculated using equation 2. All RNNs including LSTMs and GRUs have feedback loops in the recurrent layer which allow them to carry information from the past in the network model. However, RNN suffers from vanishing gradient problem which happens during backpropagation. The LSTM architecture was introduced to solve this problem.

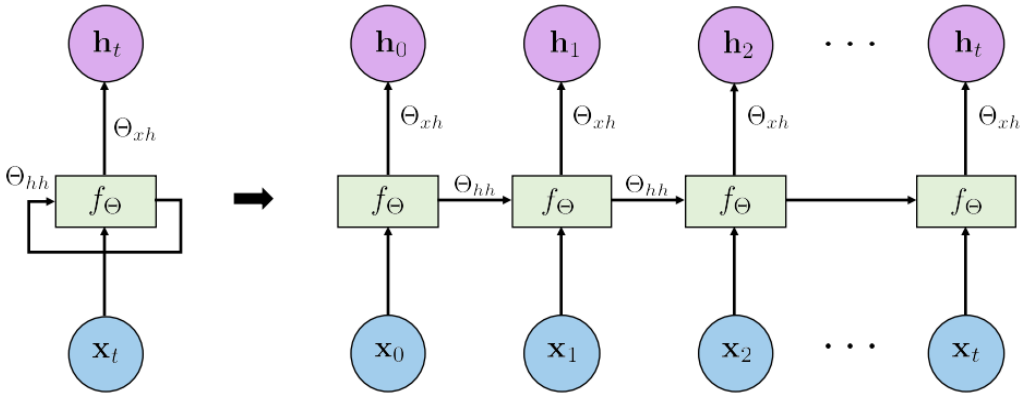

Figure 6. Unrolled Vanilla RNN [25]

$$h_t = f(h_{t-1}, x_t; \theta) \quad \downarrow$$

$$h_t = tanh(\theta_{hh} h_{t-1} + \theta_{xh} x_t) \tag{2}$$

3.2.3. LSTM & GRU

LSTM is a kind of RNN which is capable of learning and handling long term dependencies of sequential data to overcome the issues with RNN. With the architectural changes implemented in LSTM, the error propagated through time will not vanish or explode and the inputs received a long time ago still plays an important role despite the time the input was received. Hence, solving the vanishing/exploding gradient problem. At the core of LSTM is a gated mechanism that controls the flow of data by selectively passing information across individual time steps.

GRUs are similar to LSTMs but use a simplified cell structure. They also use a set of gates to control the flow of information, but they use fewer gates. These gates are also neural networks, and each gate has its own weights and biases. The major difference is that GRU has two gates: the reset gate r_t that regulates the flow of new

input to the previous memory, and the update gate z_t that determines how much of the previous memory to keep; whereas an LSTM has three gates: a forget gate for discarding irrelevant information, an input gate for handling the current input, and an output gate for producing predictions at each time step. The difference in their architecture is shown in Fig.7.

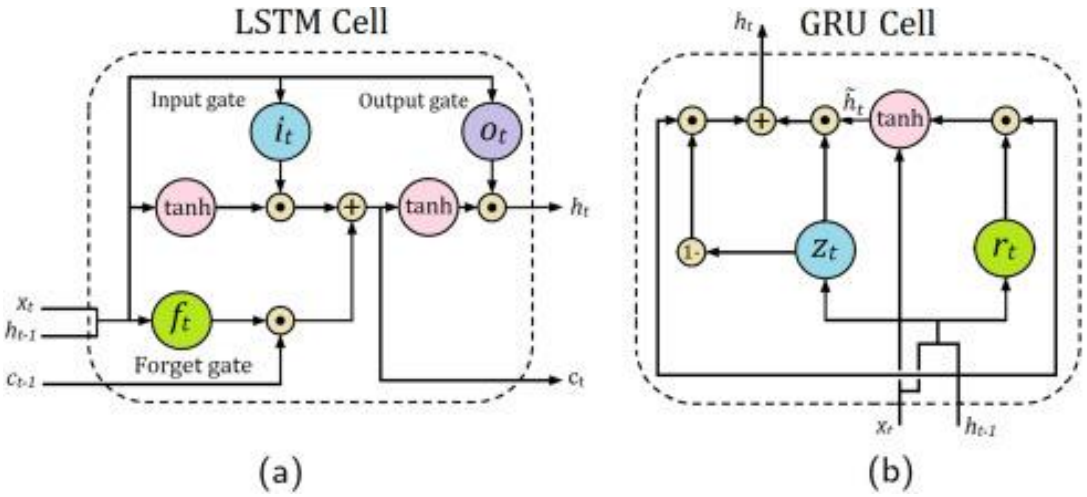

Figure 7. LSTM & GRU Cell [25]

LSTM hidden state is calculated by:

$$f_t = \sigma\left(w_f \cdot \left[h_{t-1,} x_t\right] + b_f\right)$$

$$i_t = \sigma\left(w_i \cdot \left[h_{t-1,} x_t\right] + b_i\right)$$

$$\widetilde{C}_t = tanh\left(w_c \cdot \left[h_{t-1,} x_t\right] + b_c\right)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$

$$o_t = \sigma\left(w_o \cdot \left[h_{t-1,} x_t\right] + b_o\right)$$

$$h_t = o_t * tanh(C_t) \quad (3)$$

On the other hand, GRU hidden state is calculated by:

$$Z_t = \sigma\left(w_z \cdot \left[h_{t-1,} x_t\right]\right)$$

$$r_t = \sigma\left(w_r \cdot \left[h_{t-1,} x_t\right]\right)$$

$$\tilde{h}_t = tanh\left(w \cdot \left[r_t \ * \ h_{t-1,} x_t\right]\right)$$

$$h_t = \ (1 \ - \ Z_t) * h_{t-1} \ + \ Z_t \ * \ \tilde{h}_t \qquad (4)$$

Note that weights and bias for all nodes in one layer are same, and all gates use sigmoid activation function. In addition, for structural comparison, GRU's update gate can be seen as the combination of LSTM's input gate and forget gate. GRU also merges the hidden state and cell state. But because LSTM has these as separate, it gives more controllability (i.e., flexibility in controlling the outputs) and thus, better results.

3.2.4. Bidirectional LSTM

Bi-LSTM comes as an improvement to the LSTM architecture. For most tasks, Bidirectional LSTMs learns faster than the unidirectional LSTM networks, and have also proven to be a better. Also, LSTM and GRU can adequately remember sequences of 10s and 100s but not sequences of 1000s or more, BiLSTM/BiGRU comes to the rescue. One disadvantage though is their increased computational complexity compared to the one-directional LSTM/GRU. Hence using a GPU (Graphical Processing Unit) is recommended.

Just like a vanilla RNNs, Bi-LSTMs are able to connect previous information to the present task. And they do it better, due to their bidirectional property. It can be understood as two separate LSTMs processing sequences forward and backward, and hidden layers at each time step are concatenated to form the cell output [26]. With this

bidirectional advantage, BiLSTMs have achieved success in machine translation, speech recognition, text summarization, and other NLP tasks. In this project we use Bi-LSTM because it handles bidirectional long term dependencies and remembers even longer sequences. Bi-LSTM initialized with the Glove word embeddings as embedding weights is capable of performing binary and even multi class text classification, however, as would be seen in the subsequent chapter, it gives a significantly lesser performance compared to that initialized using BERT.

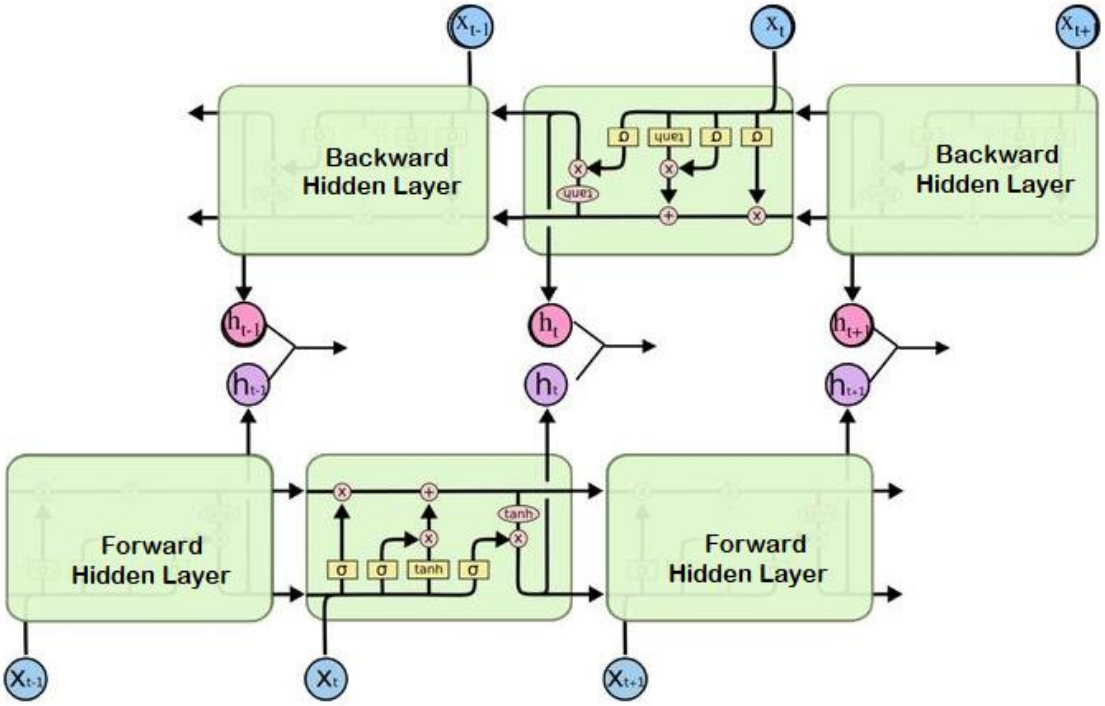

Figure 8. Bi-LSTM Architecture [21]

3.3. Attention Mechanism

First proposed by Bahdanau *et al*. [20] for the task of machine translation, where different attention weights are scored based on source words to implicitly learn the alignment for translation [20]. It solves the problem of sematic information and co-relation of words during the generation of output using attention. And it has since been

applied to many other Natural Language Processing tasks, such as this. Attention mechanism in this context is a mechanism that enables the network to learn to attend to information at different positions in the sequence of inputs during processing. This is especially important because not all words contribute equally to the meaning of a sentence, as some words may be more informative than others. By design [20], the attention layer gains access to the entire hidden state output of the BiLSTM layer as opposed only the final hidden state. That way, it considers all the input tokens for better modelling of long-distance relationships and gives attention to them accordingly. Attention mechanism has shown notable significance in sequence processing, hence its application in this project. The system uses the attention mechanism to capture distinct information from the context words.

An advantage that the attention mechanism presents is the visualization of the more important feature tokens by providing a means to extract and visualize the attention weights. Therefore, in using attention mechanism, one can derive visual insights on which tokens (words) the model learns to focus on or attend to in each document. A visual attention can be seen in chapter 4 below.

3.4. Document Classification

Upon using BERT to seed the Embedding layer of the Neural Network, I begin the document classification by progressively building the document level context vectors in a hierarchical manner, following the hierarchical structure in fig 3.

3.4.1 Word Annotation

The Word-level Bi-LSTM layer takes as input the output of the BERT encoder from equation 1 to create hidden state h_{it} at each time step, which acts as its memory of the input sequence. Following its bidirectional nature, h_{it} will be updated from both the

forward and the backward direction. The forward LSTM layer denoted as $\overrightarrow{\text{LSTM}}$ reads the sentence s_i from w_{i1} to w_{iT} and the backward LSTM layer denoted as $\overleftarrow{\text{LSTM}}$ processes the sentence in the reverse direction. The resulting hidden state h_{it} is a concatenation of the forward hidden state $\overrightarrow{h_{it}}$ and the backward hidden state $\overleftarrow{h_{it}}$.

$$\overrightarrow{h_{it}} = \overrightarrow{\text{LSTM}}(x_{it}), \qquad \text{t} \in [1, \text{T}] \tag{5}$$

$$\overleftarrow{h_{it}} = \overleftarrow{\text{LSTM}}(x_{it}), \qquad \text{t} \in [\text{T}, 1] \tag{6}$$

$$h_{it} = \left[\overrightarrow{h_{it}}; \overleftarrow{h_{it}}\right] \tag{7}$$

3.4.2 Word Attention

Following the network flow, the output h_{it} of the previous layer becomes the input of the attention layer. u_{it} is a one layer MLP that serves as the final representation of word annotation h_{it}, parameterized by word bias weight b_w and learned parameter W_w. The word attention is measured as the normalization of the correlation between the final hidden state h_{it} and a randomly initialized (learned) context vector u_w. The word annotation is then weighted and aggregated based on the attention weight vector to generate the high-level sentence vector representation s_i.

$$u_{it} = tanh(W_w h_{it} + b_w) \tag{8}$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)} \tag{9}$$

$$s_i = \sum_t \alpha_{it} h_{it} \tag{10}$$

Where α_{it} is the attention weight matrix that contains different degree of weights for corresponding words in a sentence. At this point is where [21] concludes and

proceeds to the SoftMax layer for sentence level classification. Continuing, this work has taken it further to document level classification.

3.4.3 Sentence Annotation

Progressively, given the sentence vectors s_i, the document vector can be obtained following a similar method as in 3.4.1. Using the bidirectional LSTM, s_i is trained on both positive and negative time directions, with the two parallel layers propagating in both directions. The final hidden state h_i, which is the output of the sentence-level Bi-LSTM layer (sentence annotations) is the concatenation of these two layers $\overrightarrow{h_i}$ & $\overleftarrow{h_i}$, as presented in the following equations, where *L* is the length of the document.

$$\overrightarrow{h_i} = \overrightarrow{\text{LSTM}}(s_i), \qquad \text{i} \in [1, \text{L}] \tag{11}$$

$$\overleftarrow{h_i} = \overleftarrow{\text{LSTM}}(s_i), \qquad \text{i} \in [\text{L}, 1] \tag{12}$$

$$h_i = \left[\overrightarrow{h_i}; \overleftarrow{h_i}\right] \tag{13}$$

3.4.4 Sentence Attention

Continuing in the network's architecture, the sentence level annotations are embedded into the sentence attention layer to select highly attended sentences (i.e., the sentences that serve as clues to correctly classify a document), then the document-level features are incorporated for the final document classification. Therefore, similarly, the attention score and a sentence level context vector u_s is again used to highlight the relative importance of the words in the document.

$$u_i = tanh(W_s h_i + b_s) \tag{14}$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)} \tag{15}$$

$$v = \sum_i \alpha_i h_i \tag{16}$$

Where v is the document-level vector representation that summarizes all information within sentences in a document.

3.4.5 Final Model

Finally, using (17), document vector v —having passed through the full BERT-BiLSTM-Attention model— is fed to a single-layer fully connected SoftMax classifier (a normalized logic function) to obtain the predicted probability distributions of classes i.e., the label for each document.

$$P = softmax(W_c v + b_c) \tag{17}$$

Where P is the output of the model. w_c and b_c are the learned parameters of the classification layer. As the network trains, it minimizes the binary cross-entropy loss with respect to the evaluation metrics. The binary cross-entropy loss function satisfactorily defines the objective of the network during training, as it measures the distance between the probability distributions.

To summarize the overall system methodology and its training process depicted in Fig. 3, first I carryout preliminary analysis of the input data to clean the data and also get rid of NaN values (i.e., rows with empty text) that may appear in the dataset. Then I preprocess the cleaned dataset using BERT to tokenize and obtain the contextualized word embeddings. After which the embeddings are sent to the word and sentence level BiLSTM and attention layers consecutively for processing. A SoftMax function is then applied over it at the SoftMax classification layer to subsequently produce the output.

Chapter 4

Experiments, Results, & Discussions

4.1. Dataset

For experimental purposes, I utilized a subset the WNC dataset which consists of 180k subjectively biased sentences and an equal 180k neutral sentences. Having an equal proportion of both classes make for a balanced dataset, as shown in Fig. 9. We can further see in Table I the proportion (in percentage) of the kinds of bias present in the biased dataset. We divided the dataset into development set, validation set, and test set, at 70%, 20%, and 10% respectively.

Table 1. Proportion of Bias Subcategories in the Biased Sentences [4]

Sr. No	Subcategory	Percent
1	Framing	57.7
2	Demographic	11.7
3	Epistemological	25.0

The WNC dataset in its originality contains well over 360k sentences, but for the purpose of this research, I utilized only a subset of the dataset, which amounted to the 360k that was finally experimented upon.

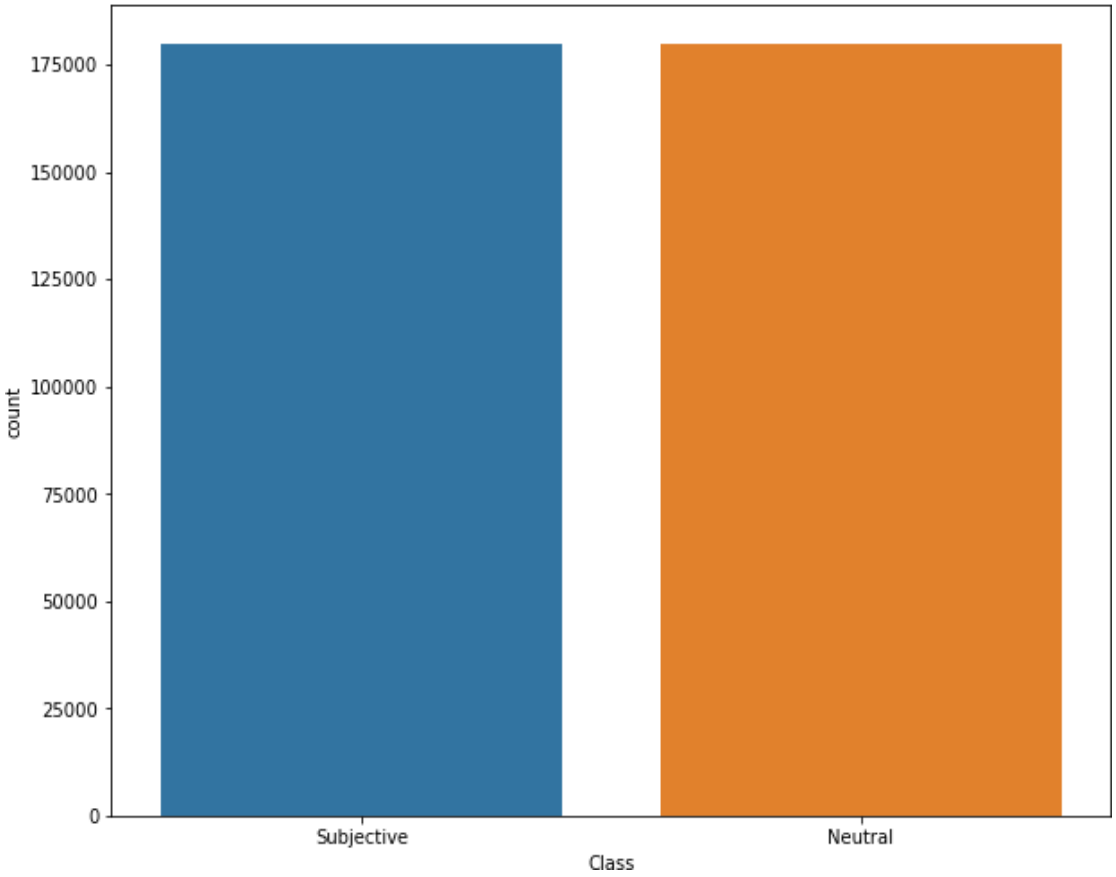

Figure 9. Data Class Proportion.

4.2. Experiments

The model was built on an experimental setup that utilized the Python environment, and Keras API with Tensorflow as backend utility. I employed Grid search technique to determine the best hyperparameters for the model, such as learning rate, dropout probability (which prevents overfitting) [27], batch size, etc. Since the model is built upon BERT, the bert-base-uncased version was selected for this task, and the BERT tokenizer was used as a tokenization tool for the dataset that consisted of the combined biased and neutral documents. A document here can be seen as a collection of sentences. The documents were labelled as 1 for the subjective texts and 0 for the neutral texts. Training of the model was set for 30 epochs. All computations were performed on a single RTX3090 GPU.

4.3. Results

As seen from Table 3, the proposed model's performance is compared with existing methods. These existing methods are widely used for solving classification problems. And since this project is a classification task, the proposed model is compared to the baseline classification models. All the presented models were implemented on the WNC dataset described in 4.1 above, using it as a benchmark for result comparison. For simplicity, we present the results based on two major evaluation metrics: Accuracy and F1-Score. To calculate the F1-score we need to first obtain the precision and recall of the test. All applied metrics are described below.

- Accuracy: is a popular evaluation indicator in classification tasks. To calculate accuracy, we divide the correctly classified samples by the total number of samples.

$$Accuracy = \frac{Correct_{\ Samples}}{Total_{\ Samples}} \tag{18}$$

- Precision: is the ratio of the correct predictions knowns as true positive (TP) to the sum of the correct predictions and the incorrect positive predictions knowns as false positive (FP).

$$Precision = \frac{TP}{TP\ +\ FP} \tag{19}$$

- Recall: is the measure of the correct predictions from the sum of the incorrect negative predictions known as false negatives (FN) and the correct predictions.

$$Recall = \frac{TP}{TP\ +\ FN} \tag{20}$$

- F1-Score: is a function of the precision and recall of the test. After obtaining the precision and recall, the F1-Score is then calculated as the harmonic mean of precision and recall.

$$F_1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (21)$$

Table 2. Sentence-Level Result Comparison.

Models	Accuracy	F1
BERT	84%	87%
BiLSTM + Glove	81%	80%
BERT + BiLSTM w/o Att	86%	86%
BERT + BiLSTM w/ Att	**89**%	**90**%

Table 3. Document-Level Result Comparison.

Models	Accuracy	F1
DocBERT (Adhikari et al., 2019)	72.7%	72.5%
Glove - BiLSTM w/ Att (Shi et al., 2019)	68.4%	67.8%
Glove - BiGRU w/ Att (Yang et al., 2016)	65.7%	65.9%
BERT - BiLSTM w/o Att	85.3%	80.3%
BERT - BiLSTM w/ Att (proposed model)	**89.5%**	**91.1%**

4.4. Discussions

Bi-GRU initialized by the Glove embeddings achieved the lowest performance with an accuracy of 65%. Followed closely by Bi-LSTM also using the Glove embeddings, which outperformed its GRU counterpart by a 3% margin. This is subsequently

followed by the BERT document classifier as presented by [28] which achieved an accuracy of 72%. Although BERT - BiLSTM without attention produced impressive results, BERT - BiLSTM with attention (which is the proposed model) produced a state-of-the-art performance, outperforming all the other models with an accuracy of 89.5%. High precision and recall were recorded on some of the baselines, however, our proposed model consistently outperformed them across all metrics. I also report performance results on the BiGRU version of our model (i.e., BERT – BiGRU - Att). From that experiment (though not recorded in this paper), we find that BiGRU achieves a comparable performance to our proposed BiLSTM in a slightly shorter time. However, BiLSTM generalizes better to the dataset thereby producing better results. To briefly touch on the sentence-level results, it can be clearly seen that the model initialized with BERT, operating with BiLSTM and attention as the downstream similarly outperformed the existing baselines. This shows that on the task of subjective bias detection, for the both sentence-level and document-levels, the proposed methodology is a better fit. These results support our motivation for seeking an improvement to existing subjectivity detection techniques.

4.5. Robustness Test

In ML, it is important to test for the robustness of a given model, as it would give you an idea of how well it would fare in a real environment, essentially giving you a measure of confidence. Robustness testing by definition is any quality assurance methodology centered around checking whether a system is robust. It is a test of the model's ability. There are certain scenarios for which robustness is tested, ranging from invalid inputs to unforeseen or never before seen inputs. It is important to note that the test for robustness varies depending on the system under evaluation. For the proposed system,

previously unseen real-world inputs were used to test for robustness. Hence, even though the proposed model was tested on the entire test set that was set aside before training, I performed a test again, this time for robustness. I utilized short documents from Wikipedia revisions of real-world samples as input to the model.

To visualize the importance and effectiveness of the attention mechanism, I integrated the attention visualization code implementation of *https://github.com/shreydesai/attention-viz* in the model, the resultant output in Fig. 10 is a visual representation of the robustness test, showing the attention applied to the sentences in each document and the corresponding label for that document.

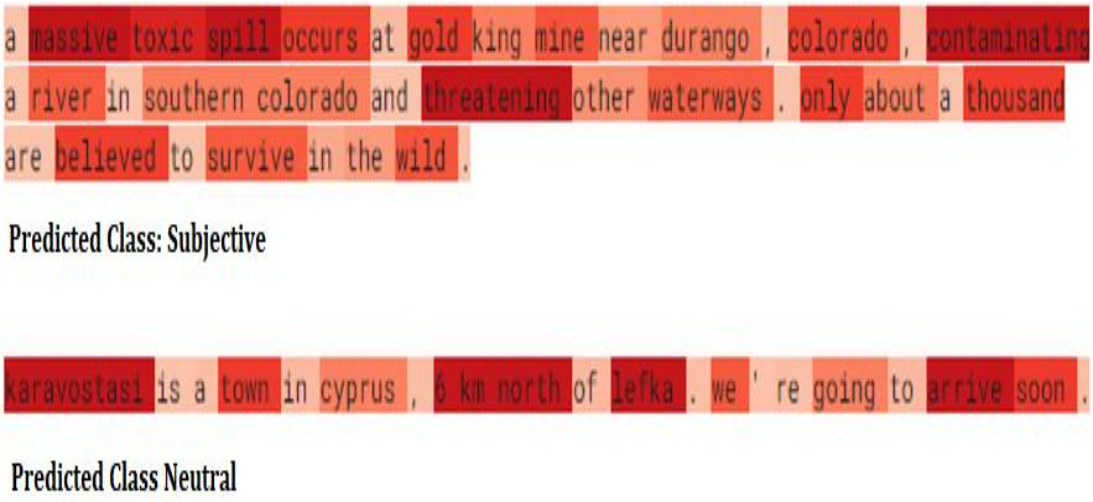

Figure 10. Attention Visualization.

Chapter 5

Conclusion & Future Studies

5.1. Conclusion

This research examined what distinguishes subjective text from the neutral counterpart. For which I utilized an approach that combines BERT, BiLSTM, and attention mechanism (BERT + BiLSTM + Attention). Using BERT as the upstream enhances the performance of downstream model. I used the WNC (which is a corpus of subjective and neutral texts alike) to benchmark the proposed model and compare it to previous approaches since the model is designed to detect subjective bias in text. The following approach levels were examined:

- Sentence level subjective bias detection: Here, the system accepts a sentence or a sequence of sentences and labels it as being subjective or neutral.
- Document level subjective bias detection: In this case, which is the main contribution of this thesis, the system accepts document sequences as opposed only individual sentences as in the first case. Here, a document is referred to as a collection of sentences. The model identifies a document as subjectively biased or objective based on the biases found in sentences that make up the document, and labels it accordingly.

The significant introduction of upstream BERT presented an improved solution to the problem of subjective bias detection, both at the sentence and document levels. Experimental results on the benchmark dataset show that the proposed model has strong representation power, giving highly competitive performance compared to baseline models with similar parameters.

5.2. Future Studies

In the future, I intend to carry out some work on refining the network to accommodate multilingual texts since this research only focused on English text representation and classification. A starting point towards achieving this will be the integration of the multilingual BERT as the networks upstream. Furthermore, I would like to work sentence and document level bias neutralization.

Bibliography

YANG, Zichao et al. Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016. p. 1480-1489., San Diego, CA, USA.

Gallup. 2018. Americans: Much misinformation, bias, inaccuracy in news. https://news.gallup.com/opinion/gallup/235796/\americansmisinformation-bias-\inaccuracy-news.aspx.

Foundation, O. S. 2018. Indicators of news media trust. https://kf-site-production.s3.amazonaws.com/media_elements/files/000/000/216/original/KnightFoundation_Panel4_Trust_Indicators_FINAL.pdf.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2019. "Automatically Neutralizing Subjective Bias in Text," ArXiv Preprint arXiv:1911.09709.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. "Linguistic Models for Analyzing and Detecting Biased Language. In Proceedings of the Association for Computer Linguistics," 1650–1659.

Kartikey Pant, Tanvi Dadu, and Radhika Mamidi. 2020. "Towards detection of subjective bias using contextualized word embeddings," In Companion Proceedings of

the Web Conference 2020, WWW 20, page 7576, New York,NY, USA. Association for Computing Machinery.

Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." Signal Processing, IEEE Transactions on 45.11 (1997): 2673-2681.2.

J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: PreTraining of Deep Bidirectional Transformers for Language Understanding.” ArXiv Preprint ArXiv:1810.04805, 2018.

X. Shi and R. Lu, "Attention-Based Bidirectional Hierarchical LSTM Networks for Text Semantic Classification," 2019 10th International Conference on Information Technology in Medicine and Education (ITME), 2019, pp. 618-622, doi: 10.1109/ITME.2019.00143.

N. Yu, and S. Kübler, 2011. “Filling the gap: Semi-supervised learning for opinion detection across domains,” In Proceedings of the 15th Conference on Computational Natural Language Learning (CoNLL 2011) (pp. 200– 209).

J. Wiebe and E. Riloff. 2005. “Creating subjective and objective sentence classifiers from unannotated texts,” In Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing (CICLing), volume 3406, pages 486–497. Springer.

W. Li, D. Li, H. Yin, L. Zhang, Z. Zhu, P. Liu, “Lexicon-Enhanced Attention Network Based on Text Representation for Sentiment Classification,” Appl. Sci. 2019, 9, 3717. https://doi.org/10.3390/app9183717.

Desislava Aleksandrova, François Lareau, Pierre André Ménard. “Multilingual sentence-level bias detection in Wikipedia,” 2019. Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2019) Varna, Bulgaria, 2019, pp. 42–51, doi: 10.26615/978-954-452-056-4_006.

Aniruddha Ghosh and Tony Veale. “Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal,” In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 482–491, 2017Phrase.

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean (2013). “Efficient Estimation of Word Representations in Vector Space.” arXiv:1301.3781 (cs.CL) 7 Sep 2013.

T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin. 2018. Advances in Pre-Training Distributed Word Representations in Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

R. Cai et al., "Sentiment Analysis About Investors and Consumers in Energy Market Based on BERT-BiLSTM," in IEEE Access, vol. 8, pp. 171408-171415, 2020, doi: 10.1109/ACCESS.2020.3024750.

D. Liu, Z. Zhao and L. Gan, “Intention Detection Based on Bert-Bilstm in Taskoriented Dialogue System,” 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, Chengdu, China, 2019, pp. 187-191, doi: 10.1109/ICCWAMTIP47768.2019.9067660.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.

Ebipatei Tunyan, T. A. Cao, and Cheol Young Ock. 2021. Improving Subjective Bias Detection Using BERT and Bidirectional LSTM. Conference Proceedings of International Conference on Empirical Methods in Natural Language Processing, Venice Italy Apr 12-13, 2021, Part V. Open Science Index, Cognitive and Language Sciences.

S. Grossberg. Recurrent neural networks. Scholarpedia, 8(2):1888, 2013. revision #138057.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio Hochreiter,2014. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling” arXiv:1412.3555 [cs.NE].

Alinlab 2018. 180912_Lec6_RNN_architectures(LSTM, GRU 등 최신 RNN 모델, 최형원+정종헌). http://alinlab.kaist.ac.kr/resource/Lec6_RNN_architectures.pdf.

Alinlab 2018. 180912_Lec6_RNN_architectures(LSTM, GRU 등 최신 RNN 모델, 최형원+정종헌). http://alinlab.kaist.ac.kr/resource/Lec6_RNN_architectures.pdf.

Adela Randall. 2017. CS 388: Natural Language Processing: LSTM Recurrent Neural Networks. https://slideplayer.com/slide/12965275/.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014. “Dropout: a simple way to prevent neural networks from overfitting,” The Journal of Machine Learning Research 15(1):1929–1958.

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, Jimmy Lin. 2019. “DocBERT: BERT for Document Classification” arXiv:1904.08398.

Appendix

Korean Abstract

주관적으로 편향된 문장을 탐지하는 작업은 매우 중요하다. 이는 텍스트나 뉴스, 소셜 미디어, 과학 텍스트, 백과사전 같은 다른 유형의 지식 전달 매체의 편향이 정보에 대한 소비자의 신뢰를 잠식하고 갈등을 촉발할 수 있기 때문이다. 주관적 편향 감지는 정서 분석, 의견 식별 및 치우침 중화 같은 많은 자연어 처리(NLP) 작업에 필수적이다. 텍스트에서 주관성을 적절하게 감지할 수 있는 시스템을 갖추는 것은 앞서 언급한 분야의 연구에 현저하게 도움이 될 것입니다. 중립 언어의 사용이 중요한 위키백과와 같은 플랫폼에도 유용할 수 있습니다. 이 논문은 문장 수준뿐만 아니라 문서 수준에서도 주관적으로 편향된 언어를 식별하는 것을 목적으로 한다.

기계 학습으로 주관적 편향 감지 문제와 같은 복잡한 AI 문제를 해결할 수 있다. 업스트림 모델로 BERT(Bidirectional Encoder Representations from Transformers)를 기반으로 분류기를 훈련하는 것은 이 접근 방식의 필수적인

요소이다. BERT 는 자체적으로 분류기로 사용될 수 있지만, 본 연구에서는 주의(attention) 메커니즘을 가진 Bi-LSTM(Bidirectional Long Short-Term Memory) 다운스트림 모델의 데이터 전처리기 및 임베딩 생성기로 사용한다. 이 방법은 보다 정확하고 포괄적인 분류기를 제공한다. 문장에서 수많은 편향된 인스턴스를 제외시킨 위키백과를 편집한 Wiki Neutrality Corpus (WNC) (WNC)를 사용하여 모델의 효율성을 평가하였다. 제안된 모델은 문장 수준에서 89%의 정확률(F1 90%)과 문서 수준에서 89.5 의 정확률(F1 91.1%)을 보여 주관적 편견을 식별하는 분야에서 현재 최고의 성능을 달성하였다. 이 모델은 다른 언어를 지원하도록 미세 조정될 수 있지만, 이 분석은 영어에 초점을 맞추었다.