Bias Article Detection in Natural Language Processing

Sai Sowmya Kovur
MS,Computer Science
University of Central Florida
kovursaisowmya@knights.ucf.edu

Ruchya Sudke
MS,Computer Science
University of Central Florida
ruchyasudke@knights.ucf.edu

ABSTRACT

Language is a powerful tool used as a medium to transmit facts, express thoughts, views, and opinions. Media can strongly impact the individual and public perception of news events. Utmost, we see a subtle bias towards or against someone or something. Adding bias leads to various problems, moreover, if those are supported by the officials then it results in the spreading of false news to the readers. This misinterpretation can lead to the generation of biased news and conspiracy theories. Hence, it is very important to identify such news articles where the information passed is wrong with added bias. The main objective of the project is to find biased news articles. Our approach is based on two distinctive characteristics: (i) Classifying the news articles into the respective publications by understanding the semantics of the sentences in the input content (ii) Detecting the bias of the news articles. We considered four different publications based on the political inclination of the news via allslides.com which classifies whether the input publications fed is left lean or right lean. To understand the semantics of the sentences and to compare the accuracy scores we used two pre-trained models. Identifying the best results after comparison we understand which publications are more inclined towards bias, which can alarm the readers.

1. MOTIVATION & PROBLEM STATEMENT

The main motive of this paper is to elucidate the role of bias, as it leads to complexity enhancement in real-world outlines. Many researcher's vigilance has proved that these biases considerably influence individuals/groups which leads to bogus nationalism. As media is one of the means for transmitting the information, adding bias leads to various problems, moreover if those are supported by the officials then it results in the spreading of false news to the readers. It is very much important to identify such articles or newspapers where the projection of data or information is wrong. Hence, the primary objective of the project is to classify whether the publications of the news articles are biased by anticipating, recognizing, dealing so that the bias is mitigated, and reliable information is communicated to the readers by analyzing and classifying whether the publications of the news articles are biased.

The project uses a Python environment with TensorFlow and Keras to build a Neural Network capable of identifying if News Stories are Left or Right-leaning. Further, the model not only identifies bias but is also capable of finding the source of the article.

2. RELATED WORK

Many researchers have researched to build efficient machine learning models which classify the bias. We examine how annotators' insensitivity towards the disparities leads to bias and we propose a method to prime dialect, to mitigate the bias in annotation via the speech detection model. Initially, a benchmark is defined by segregating the input dataset extracted from 'All the News' referring to understand the political inclination of the news via allslides.com which classifies whether the publications fed is left lean or right lean. The work presented in the paper 'Detecting Political Bias in News Articles Using Headline Attention' presents an approach to detect the political inclination of news articles Using Headlines[3]. They use an attention mechanism to detect the inclination. But their work is limited to only detecting bias and their model achieved an F1-score of 0.71. Another paper "Political Bias Analysis"[6] detects the political inclination based on the given text. Their model uses RNN and was able to obtain an accuracy of 0.69. Our work is relatable to their work, but our model is capable to classify the news articles based on the input news publications as well as their inclination. We were able to obtain an F1-score of 0.87 for Outlet and 0.87 for Media.

3. DATASET

The dataset used in this project is extracted from 'All the News' comprising various news articles from different publications, referring to understand the political inclination of the news via allslides.com which classifies whether the input(publications) fed is left lean or right lean.There are various fields(columns) present in the dataset like Id - Keeps track of the articles, title - Title of the article, publication - The name of the publication house, author - Name of the author of the article, date - Signifies the date of publication of article, and content - The article content.As we are finding the bias from the whole content related to their respective publications, we consider Publication and Content columns from the dataset and the rest columns are dropped. There

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

id	title	publication	author	date	year	month	url	content
17283	House Republicans Fret About Winning Their Hea...	New York Times	Carl Hulse	2016-12-31	2016.0	12.0	NaN	WASHINGTON — Congressional Republicans have...
17284	Rift Between Officers and Residents as Killing...	New York Times	Benjamin Mueller and Al Baker	2017-06-19	2017.0	6.0	NaN	After the bullet shells get counted, the blood...
17285	Tyrus Wong, 'Bambi' Artist Thwarted by Racial ...	New York Times	Margalit Fox	2017-01-06	2017.0	1.0	NaN	When Walt Disney's "Bambi" opened in 1942, cri...
17286	Among Deaths in 2016, a Heavy Toll in Pop Musi...	New York Times	William McDonald	2017-04-10	2017.0	4.0	NaN	Death may be the great equalizer, but it isn't...
17287	Kim Jong-un Says North Korea Is Preparing to T...	New York Times	Choe Sang-Hun	2017-01-02	2017.0	1.0	NaN	SEOUL, South Korea — North Korea's leader, ...

Figure 1: Dataset

are 3 CSV files, each approximately containing articles1.csv ≈ 50,000 news articles articles2.csv ≈ 49,999 news articles articles3.csv ≈ 100,001 news articles.

4. METHODOLOGY

4.1 Objective

The two main purpose of the paper is to classify the news articles into respective publications by understanding the semantics of the sentences in the input content and the other purpose is to detect the bias of an article or the newspaper.

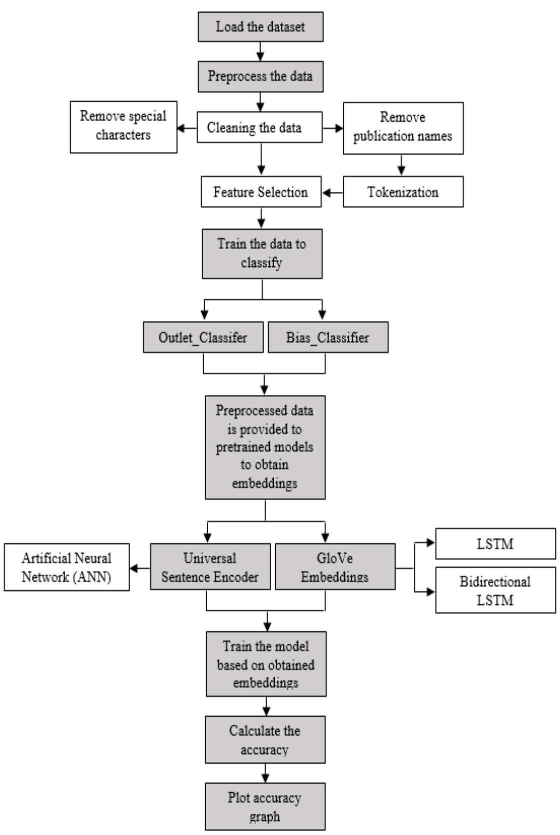

Figure 2: Pipeline

The news articles are feeded as input and are pre-processed. Then they are classified into Outlet classification and Bias classification embedded through pre-trained models upon which Natural Language Processing and Machine Learning algorithms are performed finding the best accurate model.

Referring 2 we first load the data and perform the pre-processing step which includes cleaning the data, removing special characters and publication names. Once the data is tokenized, features are selected, and the data is trained to classify into outlet classifer and bias classifier. The cleaned data is then input to the pretrained models to obtain the embeddings. Here we used two models Universal Sentence Encoder which uses ANN model and GloVe Embeddings used by LSTM and Bi-directional LSTM. Then the model is trained based on the embeddings obtained and then the accuracy is calculated and compared to find the best model for each classifier by plotting the accuracy graph.

4.2 Pre-Processing

For faster processing some random news articles are taken. These articles are classified and are assigned a unique value. Another value is assigned to the articles based on its left inclination or right inclination. If the content contains any name of the publications, we eliminate the same to reduce the noise and data redundancy.

Every publication in the data is assigned a unique value. We create a column "label" which contains values from 1 to 4 indicating the unique value assigned to every publication. The articles are classified and are assigned as 1 if they are inclined towards right and 2 if they are inclined towards left. These values are stored in column "bias".

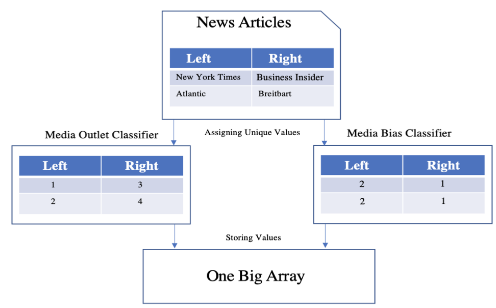

Figure 3: Pre-processing

To provide input to the model we collect the content of all the news articles and store it into one single array called one_big_array for tokenizing the words and removing the redundant and the stop words[4].

Further we create two class arrays. One array called outlet classifier will contain unique values assigned to every article which is required for outlet classification and the other array called media classifier will contain unique value assigned to every article based on their inclination either left or right. We then split the data for training and testing purposes, in this case, for testing our model. All the above steps involve pre-processing the data.

4.3 Pre-trained models

To understand the semantics of the sentence we used two pre-trained models and compared their scores to understand which model is the best fit for the data provided. One of the pre-trained models used is the Universal Sentence Encoder, which converts the input to a high dimensional vector. These high dimensional vectors are trained using Artificial Neural Network (ANN). Another model that is Glove Embeddings that are 100 dimensional vectors to understand the semantics are used. These 100-dimensional vectors are trained using LSTM and Bidirectional LSTM models.

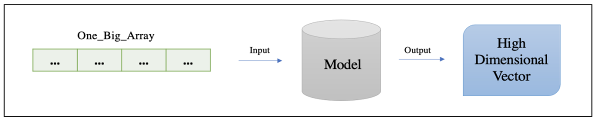

Figure 4: Model Overview

4.3.1 *Universal Sentence Encoder*

The Universal Sentence Encoder (USE) is used where the sentences from the article content are modified and are embedded to high dimensional vectors of length 512 based on the semantics of the sentence. The input provided to the model is based on the output of the Universal Sentence Encoder. This model is provided with an input of a high dimensional vector of size 512 neurons (one per embedding value) and is trained with two hidden layers which include 40 neurons each, to produce the output as a SoftMax function. The output layer involves 6 neurons out of which 4 of them are for the Media Outlet classification and the other 2 neurons are for the Media bias classification[1][8].

4.3.2 *GloVe Embedding*

Glove is an unsupervised algorithm to obtain the vector representation of words. The input to the Glove is the preprocessed data and the output is the 100-dimension vector.These 100-dimensional vectors are trained to learn the dependency of the sentences using LSTM and Bidirectional LSTM model[7].

5. EVALUATION

5.1 Artificial Neural Network

5.1.1 *Introduction to ANN Model*

The Artificial Neural Network[5] is designed to work in a similar manner as the human brain capability to process the information. The model is inputted with $x_i W_i$ where x_i is the input and W_i is the weight. The Cell Body is the summation of $x_i W_i$ added with bias b to generate an activation function f. The output is a Cell Body is the SoftMax Function z which is the product of f and summation of $x_i W_i$ added with bias b. The equation for z is as below $z = f(\sum_i x_0 w_0 + b)$

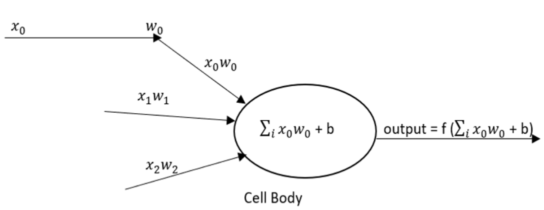

Figure 5: ANN

5.1.2 *ANN Model with Universal Sentence Encoder*

In our project we have embedded the input using Universal Sentence Encoder. This pretrained model will Generate 512 high-dimensional vectors. These 512 high dimensional vectors are trained using ANN model with two hidden layers of 40 neurons each. The output of the ANN model is a Softmax Function.

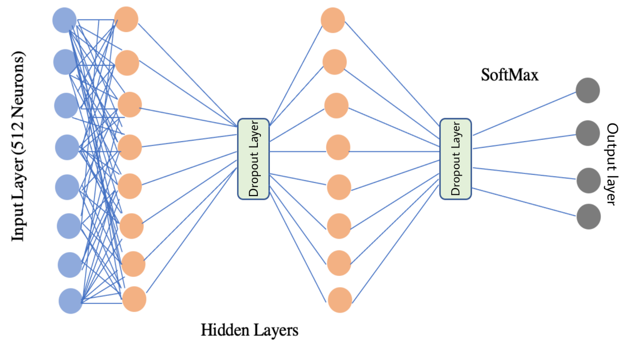

Figure 6: News Bias Classifier with ANN Visualizer

For Media Outlet Classification the number of output neurons will depend on the unique number of publications. In our experiment we considered only 4 publications and a result the output of the ANN model for Outlet Classification is 4 neurons. Each of these neurons is related to a publication. In case of Media Bias Classification, the output of the ANN model is a Softmax Function which contains two neurons. These two neurons indicate the left bias and right bias.

5.2 LSTM Model

5.2.1 *Introduction to LSTM Model*

It is type of RNN proposed by Hochreiter and Schmidhuber in 1997. It was primarily introduced to solve the problem of Vanishing Gradient which was observed in the RNN models. For every step t, there is a hidden state h^t and cell state c^t. The h^t and c^t are both vectors of length n. The cell stores the long-term information. Thus, providing the LSTM with the ability to erase, write and read information from the cell. The selection of the information which needs to be erased, written and read is controlled by three corresponding gates. The gates are also vector of length n. For every step in timestamp t the gates can be open, closed or in between the two. The values of the gates are dependent on the current action and thus possess a dynamic nature. Every step t has a input $x^{(t)}$ and the LSTM model will compute sequence of hidden state $h^{(t)}$ and the cell states $c^{(t)}$[10].

Basic Diagrammatic Representation of LSTM is given as below:

Every step t has three gates: a) Forget State f^t: This gate is responsible to decide if the information received from the previous cell needs to be preserved or needs to be forgotten. The equation for the forgot state is given as below:
$f^t = \sigma(W_f h^{(t-1)} + U_f x^{(t)} + b_f)$

b) Input Gate i^t: This gate is responsible to decide which

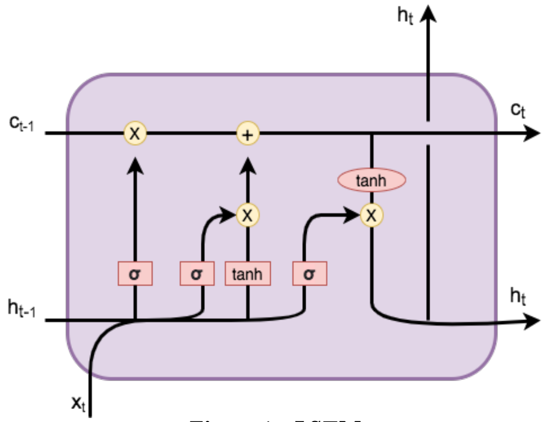

Figure 7: LSTM

portion of the new cell content has to be written to the cell. The equation for the Input Gate is given as below:
$i^t = \sigma(W_i h^{(t-1)} + U_i x^{(t)} + b_i)$

c) Output Gate o^t This gate is responsible to decide which part of the cell will serve as the output to the hidden state. The equation for the output gate is given as below:
$o^t = \sigma(W_o h^{(t-1)} + U_o x^{(t)} + b_o)$

For every timestep t there are three states associated:
a) New cell content c^{-t} : In this state new content is written to the cell. The equation of the state is given as below:
$c^{-t} = tanh(W_c h^{(t-1)} + U_c x^{(t)} + b_c)$

b) Cell State c^t This state is responsible for erasing the content from the last cell state and update the cell with the new cell content. The equation for cell state c^t is given as below:
$c^t = f^{(t)} o c^{(t-1)} + i^{(t)} o c^{(-t)}$

c) Hidden State h^t: The hidden state The equation for hidden state for every timestep t is given below:
$o^t = \sigma(W_o h^{(t-1)} + U_o x^{(t)} + b_o)$

5.2.2 *LSTM Model with GloVe Embeddings*

The Glove Embeddings is used as a pretrained model to learn the semantics of the news articles. The GloVe pretrained model generates 100 high dimensional vectors and these vectors reflect the semantics of the sentences. These 100 high dimensional vectors are then trained using a single layer of LSTM to generate a Softmax function. In case of Outlet Classification, the Softmax function contains 4 neurons and every neuron resembles a unique publication. In case of Media Classification, the Softmax function contains 2 neurons which indicates either left or right.

5.3 Bidirectional LSTM

5.3.1 *Introduction to Bidirectional LSTM*

They are an extension of LSTM Model. The main idea of Bidirectional LSTM Model is trains two LSTM instead of one LSTM on the input sequence. The first on the input sequence and the second on the reverse copy of the input sequence. This provides the Bidirectional LSTM Model with additional context and result in faster learning. The first LSTM is trained in forward direction and the same LSTM is trained in backward direction for better performance[2].

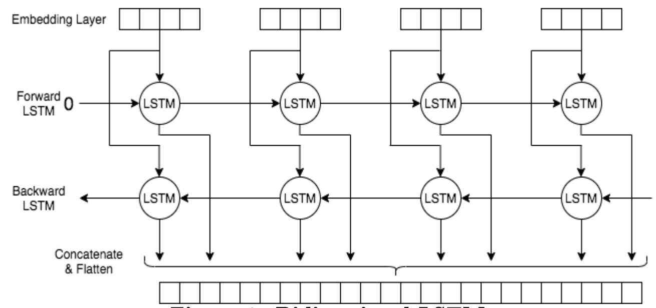

Figure 8: Bidirectional LSTM

5.3.2 *Bidirectional LSTM with GloVe Embedding*

The 100-dimensional vectors generated through GloVe pretrained model are provided as the input to the Bidirectional LSTM. The Bidirectional LSTM model trains the LSTM in forward direction and backward direction and generate Softmax Function. In case of Outlet Classification, the Softmax function contains 4 neurons, and every neuron resembles a unique publication. In case of Media Classification, the Softmax function contains 2 neurons which indicates either left or right.

6. RESULTS AND OBSERVATIONS

6.1 Natural Language Processing Models

Accuracy: A highly accurate navigation system can have measurements that are very similar to the normal, true, or known values. Accuracy is described as the closeness of a measurement to the standard or true value. The standard deviation of errors is commonly used to represent accuracy.
The Accuracy score of LSTM for Media Bias Classification is 86.16 and Media Outlet Classification is 77. The Accuracy score of Bidirectional LSTM for Media Bias Classification is 77.61 and Media Outlet Classification is 79. The Accuracy score of ANN for Media Bias Classification is 0.87 and Media Outlet Classification is 0.63.

Table 1: NLP Model Accuracy

	ANN	LSTM	Bi-LSTM
Media Bias Classification	87	86.16	77.61
Media Outlet Classification	63	77	79

Refer table 1 For the Media Bias Classification, ANN model gives the highest accuracy which is trained using the pre-trained model - Universal Sentence Encoder.
For the Media Outlet Classification, Bidirectional LSTM gives the highest accuracy score.

To get better results we try balancing the classes with an oversampler. We will use Synthetic Minority Oversampling

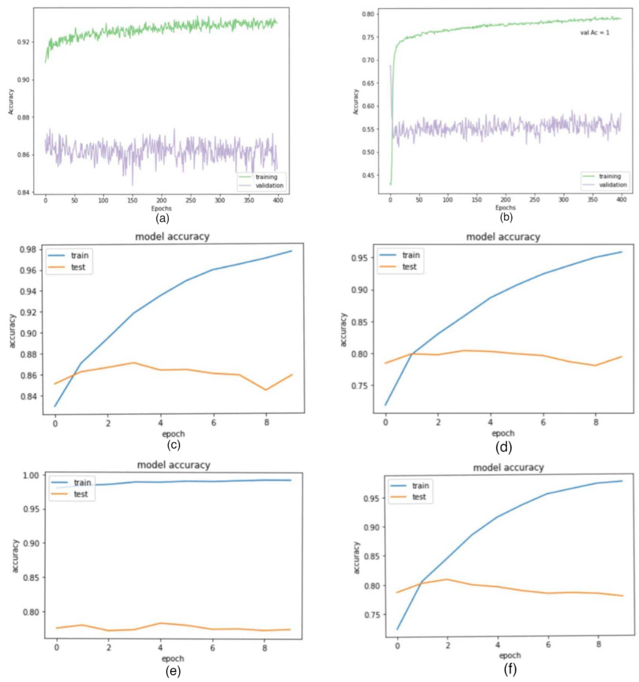

Figure 9: Accuracy of (a)ANN Media Bias Classification; (b)ANN Media Outlet Classification; (c)LSTM Media Bias Classification; (d)LSTM Media Outlet Classification; (e)Bidirectional LSTM Media Bias Classification; (f)Bidirectional LSTM Media Outlet Classification

Technique (SMOTE)from the Python library imblearn. The confusion matrix after training with balanced data[9].

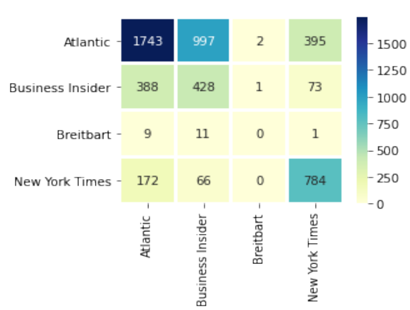

Figure 10: Confusion Matrix(Outlet)

It depicts the four articles fed, with the accuracy values. As the data is randomly selected the accuracy depends on the sample selected and may vary for other samples.For instance, if the number of articles from New York Times or BreitBart is greater than there will be greater then the accuracy for these two publications will be higher. Thus, we will resample the data for better accuracy score.

In figure 11 it depicts the articles that are classified as left or right lean and the accuracy depends on the sample selected. We used F1 score metrics which is helpful for uneven class distributions and computed it with sklearn's built-in function and tried balancing the classes in the training set

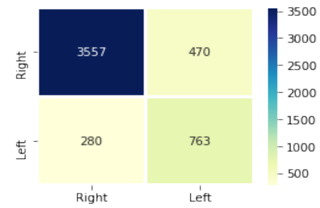

Figure 11: Confusion Matrix(Bias)

with an oversampler to get better results and used SMOTE from the Python library imblearn.

Loss Function

When it comes to neural networks, the aim is usually to reduce the amount of error. The loss function determines the distance between current output and expected output. In our project, we observed that the loss was minimal for the data points. The overfitting of the data points was also avoided with respect to our experiment.

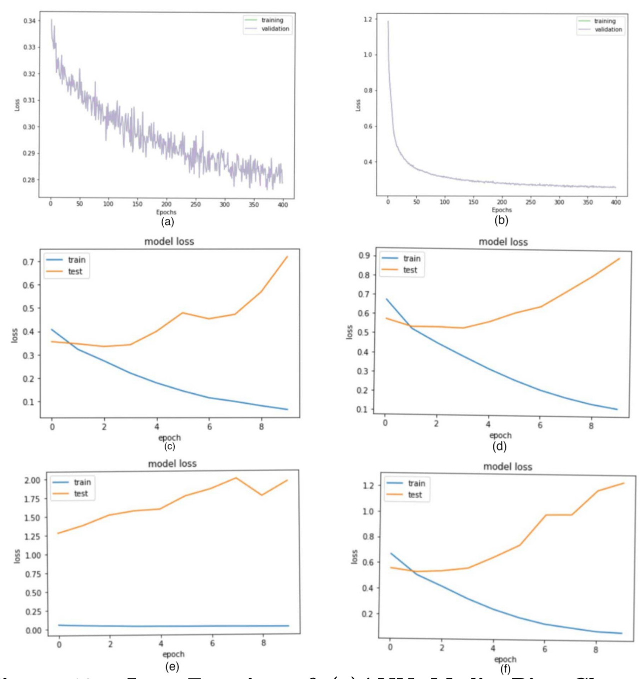

Figure 12: Loss Function of (a)ANN Media Bias Classification; (b)ANN Media Outlet Classification; (c)LSTM Media Bias Classification; (d)LSTM Media Outlet Classification; (e)Bidirectional LSTM Media Bias Classification; (f)Bidirectional LSTM Media Outlet Classification

6.2 Machine Learning Models

We have trained our data on various Machine Learning models that is KNN, SVM, Logistic Regression, Naive Bayes, Stochastic Gradient Decent, Linear SVC.

Table 2: LSTM -ML Model Accuracy(Outlet)

	Model	Score
1	KNN	71.41
0	Support Vector Machines	64.64
2	Logistic Regression	61.66
3	Naive Bayes	48.64
4	Stochastic Gradient Decent	41.86
5	Linear SVC	41.33

Table 3: LSTM -ML Model Accuracy(Bias)

	Model	Score
1	KNN	84.24
0	Support Vector Machines	80.36
2	Logistic Regression	79.25
3	Naive Bayes	69.69
5	Linear SVC	62.04
4	Stochastic Gradient Decent	60.63

- Applying the Machine Learning models on the dataset trained via LSTM model, we observe KNN best suits our model.

Table 4: Bidirectional LSTM -ML Model Accuracy(Outlet)

	Model	Score
1	KNN	71.41
0	Support Vector Machines	64.64
2	Logistic Regression	61.66
4	Stochastic Gradient Decent	51.69
3	Naive Bayes	48.64
5	Linear SVC	45.57

Table 5: Bidirectional LSTM -ML Model Accuracy(Bias)

	Model	Score
1	KNN	84.24
0	Support Vector Machines	80.36
2	Logistic Regression	79.25
3	Naive Bayes	69.69
4	Stochastic Gradient Decent	63.95
5	Linear SVC	52.39

- Applying the Machine Learning models on the dataset trained via Bidirectional LSTM model, we observe KNN best suits our model.

Table 6: ANN - ML Model Accuracy(Outlet)

	Model	Score
1	KNN	78.45
0	Support Vector Machines	77.03
5	Linear SVC	73.88
2	Logistic Regression	73.23
4	Stochastic Gradient Decent	68.22
6	Decision Tree	66.25
3	Gaussian	34.39

Table 7: ANN -ML Model Accuracy(Bias)

	Model	Score
0	Support Vector Machines	93.33
1	KNN	90.69
5	Linear SVC	90.49
2	Logistic Regression	89.96
4	Stochastic Gradient Decent	86.82
6	Decision Tree	83.18
3	Naive Bayes	68.29

- Applying the Machine Learning models on the dataset trained via ANN model, we observe KNN model best suits Media Outlet Classification and SVM best suits Media Bias Classification model.

7. CONCLUSION

Our model is capable of classifying the publications based on respective publication and detect if the publication is left biased or right biased. ANN trained using Universal Sentence Encoder pretrained model yields an accuracy score of 0.87 result in classifying the media bias. However, Bidirectional LSTM trained using GloVE pretrained model yields an accuracy score of 79 as compared to ANN and LSTM. It is difficult to conclude with a huge sample dataset and the number of news articles from every publication is different. The important observation that semantic rules of the news articles are politically inclined and contain bias. Although we already knew the political inclination of the news articles but through tools, we were able to classify the articles based on their political inclination.

8. FUTURE SCOPE

We were able to detect the political inclination of the news publications. Our project lays the foundation to further detect different types of bias in news articles. Detecting news articles which are gender bias, biased based on color. In our project we trained the model to learn the semantics of the news articles and classify the concatenated content of all the news publication into the respective news publication. We can further classify these news articles into their different categories like Entertainment News, Criminal News, Political News and so on.

References

[1] Daniel Cer et al. *Universal Sentence Encoder*. 2018. arXiv: 1803.11175 `[cs.CL]`.

[2] Zixiang Ding et al. *Densely Connected Bidirectional LSTM with Applications to Sentence Classification*. arXiv: 1802.00889 `[cs.CL]`.

[3] Rama Rohit Reddy Gangula, Suma Reddy Duggenpudi, and Radhika Mamidi. "Detecting Political Bias in News Articles Using Headline Attention". In: DOI: 10.18653/v1/W19-4809.

[4] Felix Hamborg. "Media Bias, the Social Sciences, and NLP: Automating Frame Analyses to Identify Bias by Word Choice and Labeling". In: DOI: 10.18653/v1/2020.acl-srw.12.

[5] Manish Mishra and Monika Srivastava. "A view of Artificial Neural Network". In: DOI: 10.1109/ICAETR.2014.7012785.

[6] Arkajyoti Misra. "Political Bias Analysis". In: 2016.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: DOI: 10.3115/v1/D14-1162.

[8] "Universal Sentence Encoder for English". In: DOI: 10.18653/v1/D18-2029.

[9] Jerry Wei. *NewB: 200,000+ Sentences for Political Bias Detection*. arXiv: 2006.03051 `[cs.CL]`.

[10] Lirong Yao and Yazhuo Guan. "An Improved LSTM Structure for Natural Language Processing". In: DOI: 10.1109/IICSPI.2018.8690387.