MEDIA LEGITIMACY DETECTION: A DATA SCIENCE APPROACH TO LOCATE FALSEHOODS AND BIAS USING SUPERVISED MACHINE LEARNING AND NATURAL-LANGUAGE PROCESSING

Nathan Ji[1] and Yu Sun[2]

[1]Portola High School, Irvine, CA, 92618
[2]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Media sources, primarily of the political variation, have a hastening grip on narratives that can easily be constructed using biased views and false information. Unfortunately, many people in modern society are unable to differentiate these false narratives from real events. Utilizing natural language processing, sentiment analysis, and various other computer science techniques, models can be generated to help users immediately detect bias and falsehoods in political media. The models created in this experiment were able to detect up to 70% accuracy on political bias and 73% accuracy on falsehoods by utilizing datasets from a variety of collections of both political media and other mediums of information. Overall, the models were successful as the standard for most natural language processing models achieved only about 75% accuracy.

KEYWORDS

Data Science, Political Bias, Fake News, Supervised Machine Learning, and natural-language processing.

1. INTRODUCTION

Political bias and fake news is often a challenging task to tackle, especially due to the volatile nature of modern media. The problem is often so engrained, especially in a polarized American society, that it greatly distorts many voters' grip on reality [1] and has been argued to create dubious practices that can often harm society and other people [2]. Although there have been some approaches to tackling the problem, like the Bipartisan press who uses AI to determine bias in their own news articles [3], research in this field is too scarce to offer significant help. Therefore, other scientific methods should be employed to effectively help combat these problems that have begun to encroach on the global order.

The primary goal of the paper is to explore whether data science can be adequately applied to a social science field of identifying political bias and falsehoods. Data science can be broadly understood to be the practice of filtering noisy and unstructured data through algorithms, models, and other scientific practices to achieve results that can be easily interpreted by human users. In the context of this paper, the data is primarily thousands of sentences that contain varying degrees of bias and truthfulness, which is then fed into a variety of models to attempt to achieve the goal.

David C. Wyld et al. (Eds): CONEDU, CSITA, MLCL, ISPR, NATAP, ARIN - 2022
pp. 31-41, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121003

This methodology often requires techniques to gather large swaths of data, which is incorporated within the procedures later on. The bulk of the data used in this paper has already been gleaned from political sources like senate debates and headlines in popular media.

The various scientific methods mentioned in the data science approach is where the largest variation of outcomes occurs in the paper. There are two primary differences in how the data can be interpreted using natural language processing (NLP) and supervised machine learning, which is the augmentation of a dataset and the kind of machine learning, using a variety of approaches often used in data science for different kinds of data.

Augmenting the dataset is often the first step in this procedure. Most of the time, the models only required a simple augmentation in which special characters and punctuation were removed if needed and the data was changed into a binary numerical format in order for our regression models to have any effectiveness [4]. Since logistic regression and linear regression both require binary outputs of non correlated, individual outcomes, it was important to not have a gradient of falsehoods or bias in the data to ensure this data could work. However, for our initial bigram model, the augmentation was far more complicated since the convote dataset came from senate speeches, which very often contained highly menial sentences including sentences like “Mr. Chairman, I thank the gentlewoman for yielding me this time” or “I yield to the gentleman from illinois” [5]. This meant that there needed to be a way to distinguish what sentences reflected Republican and Democratic senators’ ideology with those that contained only procedural and rather irrelevant information. A method offered by the University of Maryland was to split the dataset into bigrams, a string of two words, and rank them in usage to feed into the model [6]. This augmentation would allow the isolation of popular political phrases like “illegal alien,” which would help identify republican and democratic bias. Further augmentation was used in the actual model, in which the sentences went through pipelines to numericalize the data [7].

Afterwards, choosing the most accurate models and algorithms for interpreting the data was often highly tricky given that there was no knowledge of whether the data was linearly separable or if they were clustered into various groups. The approach was simply to try all the models available to get a better understanding of what the data looked like, which would then allow more specified tweaking of the algorithms. The initial tests only produced very low accuracy scores of 50-60%, meaning that further augmentation and testing was needed. After adding more robust methodology, the final result was that logistic regression and random forest classifiers were both effective and returned relatively similar accuracy of around 80%, which meant that the data was in some way separable in terms of frequency of different word vectors [7]. These types of models are all supervised machine learning, meaning that they were split in train and test sets to develop the model using incorrect and correct answers. Techniques like regression use a loss function to determine the validity of the current model and make necessary changes in order to increase the amount of data points correctly assessed [8]. This paper utilizes both types of supervised machine learning, classification and regression, to determine which best suits the data. In the end, both proved to be nearly equally useful with varying augmentation, meaning that a multitude of future techniques are still open to be able to be used. The final prototype was considered successful in that it proved that data science could be a viable method in identifying political bias and falsehoods purely from analysis of linguistic patterns.

The rest of the paper is organized as follows: challenges of creating an NLP model for fake news, the solution, including the methodology in reference to datasets and algorithms used, the ending results from the algorithms, and then a discussion of the results as well as the implications in the realm of natural language processing.

2. CHALLENGES

In order to create a model that could accurately identify political bias and fake news, a few challenges have been identified as follows.

2.1. Challenge 1

Initial approaches utilized sentiment analysis from various libraries including NLTK using the vader lexicon. However, the primary problem in this usage is that political media can often have highly varying sentiment without actually being false. For instance, sentences like “President Biden signs a new bill” and “a freeway collision has left 6 people dead and 15 injured in a large explosion” are both factually correct, but contain very different sentiments. Furthermore, this meant that the data was not going to be linearly separable nor was it going to be reliable, meaning that as previously stated, regression models would not work on the data and even decision trees could not give a consistent answer due to the vastly varying sentiments.

2.2. Challenge 2

Datasets play a critical factor within the legitimacy and reliability of a data science model, and those for political bias are rather scarce, making it difficult to come to accurate conclusions since identifying political bias needs to mimic human behavior and therefore needs to be generated manually. This presented a challenge as for each dataset found, testing had to be done to each one to determine if the data could be used reliably, and only until the 5th try was a viable dataset found. Additionally, many of the datasets found themselves using non numeric values, which could not be categorized by one hot encoding the data since many times the non numeric values were a gradient that was rather arbitrary. Ultimately, the usage of multiple datasets together would cushion the effect of many inaccuracies the data would have, as well as the usage of converting the non numeric values to a gradient through trial and error.

2.3. Challenge 3

Another challenge that was posed was the effectiveness of the models as well as the metrics generated by Sklearn. It was found that many times the data was extremely overfit or underfit to the dataset, and generated results that do not mimic human behavior. When testing some models, sklearn returned metrics that were rather high, but on manual testing of the model, it was found that the model would determine that sentences like “The sun is 93 million miles away from the Earth” or “The quick brown fox jumped over the lazy dog” were biased or false. Many times, objective sentences like these were flagged incorrectly by a rather high probability, which resulted in either the problems with the pipelines used as well as the datasets. The solution was a multi-faceted approach in which factual and objective sentences would be gleaned from wikipedia to better train the model in identifying these models, as well as changing the parameters that were fed into the model.

3. SOLUTION

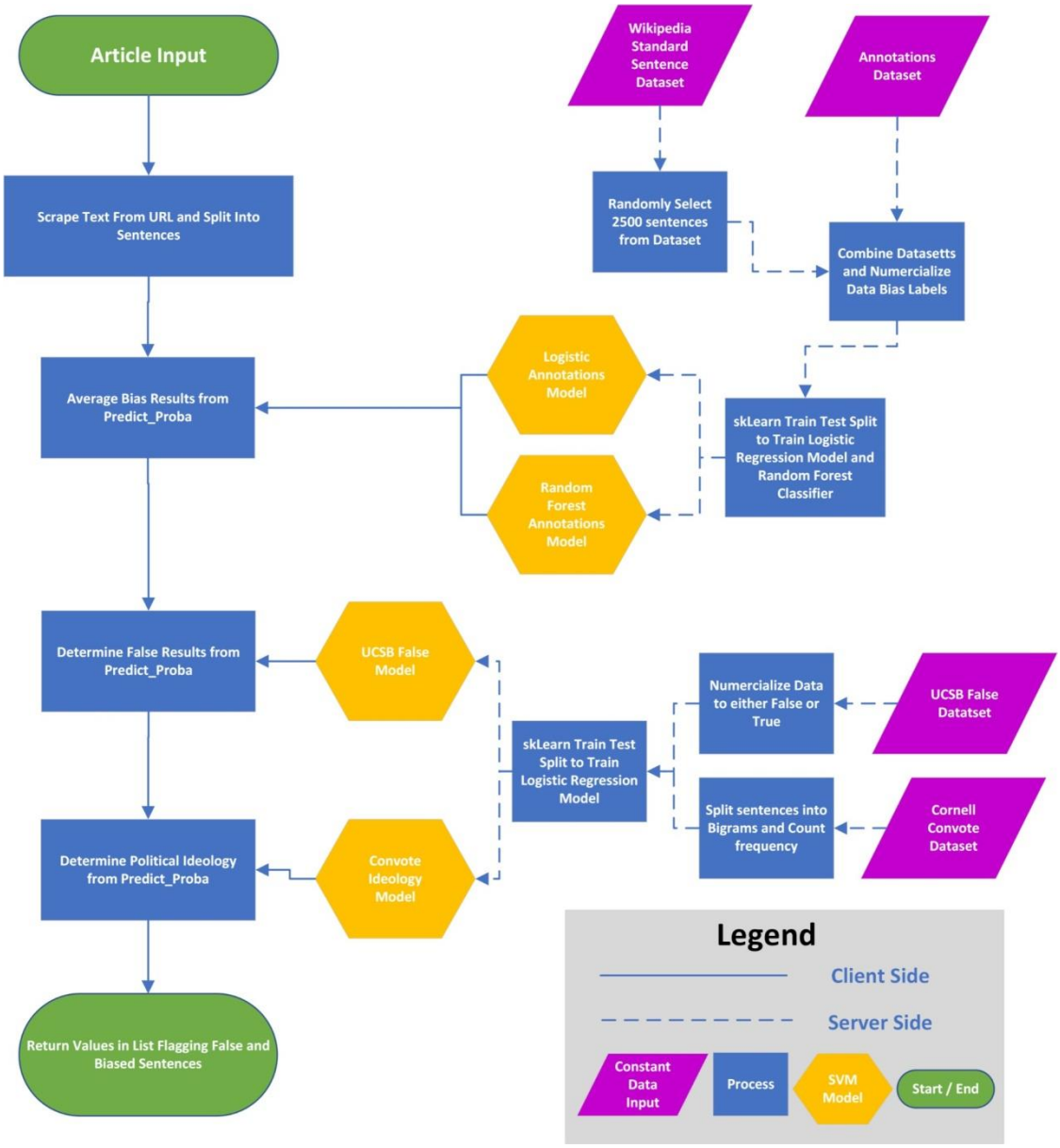

Figure 3.1. The Finalized Prototype

The final solution utilized a variety of techniques that were found to have the highest success rate. The inputs require 4 datasets that will be processed into various models using sklearn's pipelines as well as splitting the data into test and training for the models. Although these models do not need to be regenerated per article using the pickle library, the flowchart displays the first iteration of the prototype.

3.1. Datasets

The Annotations Dataset and Wikipedia Standard Sentence Dataset were both used jointly in determining bias. The Annotations Dataset, developed by MBIC, utilized 10 annotators who were given a set of 17700 sentences and asked to describe various information, including bias, author

information, publication information, and many other characteristics of the data. However, using the dataset alone resulted in data that was overfit since the dataset was

unbalanced as the ratio of biased to unbiased was 10651:7124, which made the model predict bias for factual scientific sentences. The solution was to concatenate the dataset with a split version of the Wikipedia Standard Sentence Dataset, which was used with the assumption that all sentences in the dataset are factual. By using a random number generator, the dataset was reduced from 7 million sentences to only 2500, which would balance the dataset and also include linguistically unique sentences that the annotation dataset might not be including. Afterwards, the data was numerated where biased was replaced by 1 and unbiased was replaced by 0.

The Cornell Convote Dataset gathered the dialogue from Senate speeches in 2006 where senators marked as either democratic or republican had individual files containing all of their dialogue in senate debates [5]. An important consideration is that these files included all dialogue in senate debates including sentences like yielding to other senators, thanking the chairman, and other rather nonsensical sentences that weren't relevant to the task of understanding ideological bias in linguistic patterns. The solution for augmenting this dataset was much different than the other two and did not use full sentences as inputs to a supervised machine learning model. Rather, the sentences were processed by splitting them into a list of bigrams, evaluating the frequency of the bigrams and using those as the weight, which also utilized the count vectorizer and the TFIDF transformer to evaluate sentences based on those bigrams. The other major difference is that this model dataset does not evaluate false or biased media but if they are right or left leaning sentences.

The UCSB False Dataset labeled political sentences as one of six labels on a gradient of truthfulness. Obtained from multiple mediums like social media, the data points can be from both conservatives and liberals and was cross validated manually through human analysis [9]. The dataset also was balanced beforehand, only containing a slight excess of "pants-fire" labels. However, the solution requires a binary output, which is not really compatible with a six label dataset, so rather than balancing the dataset by removing pants-fire label, the solution simply groups the first four true labels into a true output and the next 2 false labels into a false label, thereby balancing the dataset for Supervised machine learning models.

The novel dataset used in this paper utilizes web scraping and models created previously to identify bias to create a larger corpus to train a neural network. The dataset is created by scrapping multiple political news sites that have been evaluated by other papers using human annotators to determine the ideological bias as well as the presence of fake and misleading information. Each label is generated by the

3.2. Models

All of the models utilized in the experiment were Supervised machine learning models that provided a binary output that varied based on the dataset. These models were all provided by sklearn and focused on linguistic analysis as the sole benchmark for finding falsehoods and bias [7]. Logistic regression was used for all three datasets and a random forest classifier was also used in addition to logistic regression in the UCSB False Dataset. Using train test split to evaluate the model, the data was split 80:20 respectively and fed into a pipeline to numericalize the data, which in its raw form is a string.

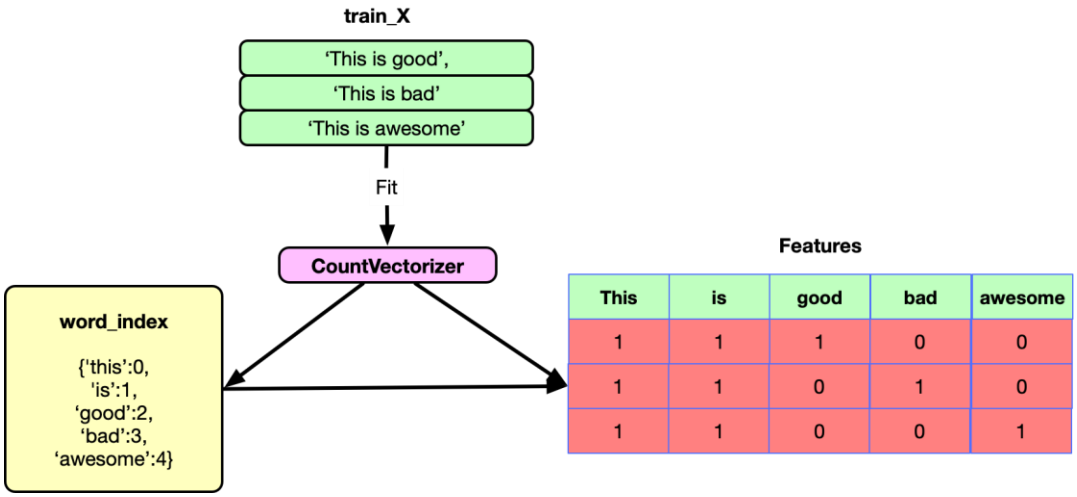

Figure 1. A flowchart of the Counter Vectorizer pipeline [7].

The first step of the pipeline is to pass the sentences as documents into the CountVectorizer pipeline provided by sklearn, which counts the frequency of the occurrences within each sentence and uses that as the word's weight. This is an extension of the bag of words linguistic technique that evaluates linguistic patterns through the usage of frequency. Then, after evaluating the frequency of each word, the solution evaluates its TF-IDF score utilizing two mathematical terms. Given N number of documents, d is the given document, D is the collection of all documents and w is the given word.

$$tf(w,\ d)\ =\ log(1\ +\ f(w,d))$$

Equation 1: The term frequency (tf) calculation [7]

The solution finds term frequency by taking the natural logarithm of the frequency of the word in the document calculated in the CountVectorizer pipeline added to one. In order to isolate key words within individual documents, the solution rewards a greater frequency for words that occur often in an individual document to emphasize the importance of the word.

$$idf(w,\ D)\ =\ log(\frac{N}{f(w,\ D)})$$

Equation 2: The inverse document frequency (idf) calculation [7]

The solution finds inverse document frequency by once again taking the natural logarithm of the number of documents divided by the frequency of a word in all the documents combined. The intention of this formula is to isolate words that may not be helpful in linguistic analysis if it is consistently present in all documents like articles “the” or “is” and help emphasize words that appear prevalently in either one document or a small group of documents.

$$tfidf(w,\ d,\ D)\ =\ tf(w,\ d)\ *\ idf(w,\ D)$$

Equation 3: TF-IDF calculation multiplying term frequency and inverse document frequency [7]

To fully transform the weights from the counter vectorizer into TF-IDF scores we multiply both

the term frequency and the inverse document frequency to find keywords that prove to be useful for further linguistic analysis. Afterwards, the list of binary labels and TF-IDF scores can be fed into supervised machine learning models that can be separable using logistic regression or classified from Random Forest Classifiers. The solution then calls predict_proba on the model that has been fit to the new data to find both the output of the Supervised machine learning models as well as the probability of the output to identify if there is a gradient.

4. EXPERIMENT

The final solution uses multiple supervised machine learning models that are able to discern frequencies within the linguistics of media to reach a conclusion about media bias and falsehoods. The solution's data can be seen below.

4.1. Results and Calculations

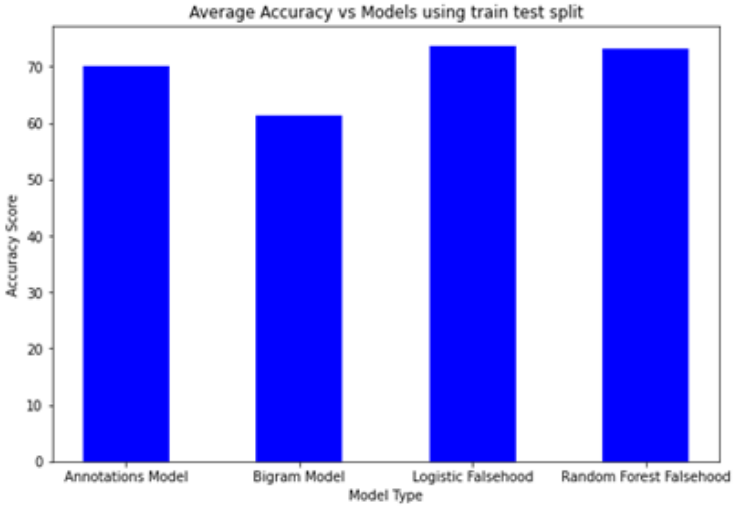

Figure 2. Provides results of Accuracy of all four models at identifying either falsehoods or biasover 10 fold repetitions where the model is regenerated and retrained each time.

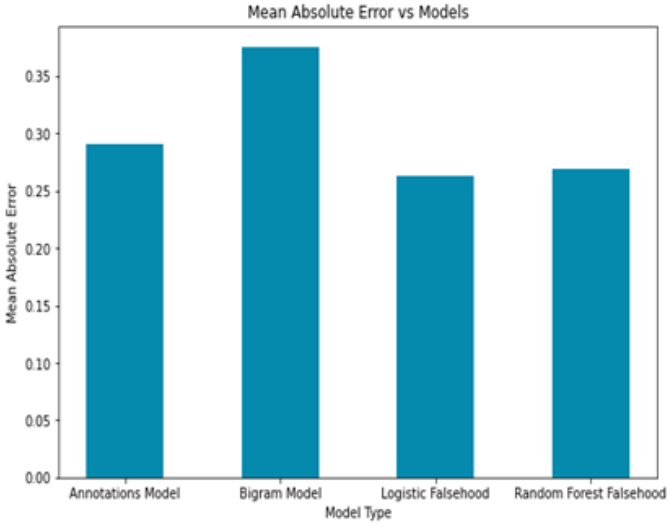

Figure 3. Provides the Mean Absolute Error, evaluated by equation 4 of all four models at identifying either falsehoods or bias over 10 fold repetitions where the model is regenerated and retrained each time.

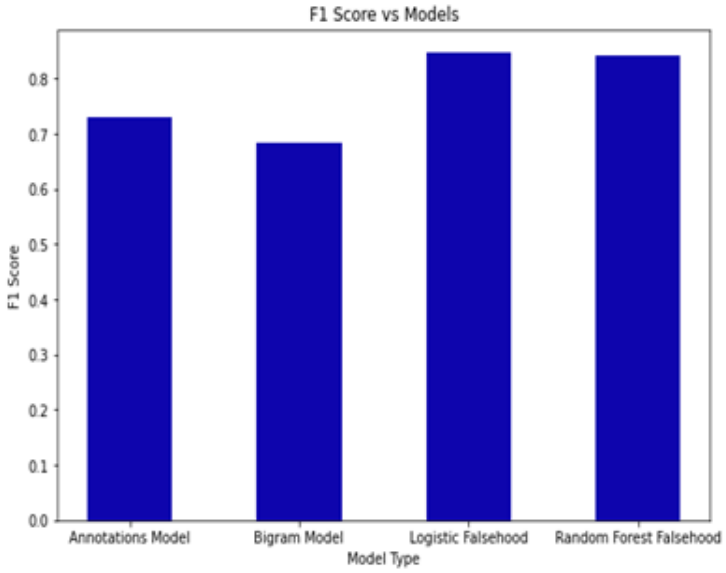

Figure 4. Provides the F1 scores, evaluated by equation 5, of all four models at identifying either falsehoods or bias over 10 fold repetitions where the model is regenerated and retrained each time.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y_i}|$$

Equation 4: Mean Absolute Error Calculation where n is the number of terms and y_i is the expected value and $\widehat{y_i}$ is the observed value.

$$F_1 = 2 * \frac{tp}{tp + \frac{1}{2}(fp + fn)}$$

Equation 5: Derived version of the F1 score calculation obtained from substituting the definition of recall and precision where tp, fp, and fn stand for true positives, false positives, and false negatives respectively.

4.2. Discussion

The results of the model's outcomes for the Annotations and two falsehoods are far more accurate than the Bigram Model and are more supportable for real world application. Understanding that for all the models, the two binary outputs are 1 and 0, any Mean Absolute Error (MAE) that is under 0.5 is essentially all relatively the same in terms of accuracy. This is since one can imagine a cluster around both these endpoints, noting that a MAE that is exactly 0.5 has no overlap between these two clusters. Given that the model's MAE is around 0.2-0.3 aside from the Bigram model, the clusters of results are fairly accurate. The other primarily significant data point is the F1 score of each model, where 1 is absolutely perfect and 0 is entirely incorrect. Understanding the formulas below, the f1 score can be primarily thought of as the harmonic mean between precision and recall. These two levy the correct answers with the two undesired results, effectively creating a numerical representation of how many false positives or false negatives there are in relation to our true positives. Therefore, the higher F1 score displays that the models

are accurate and do not exhibit a worrying amount of incorrect answers. Overall, the statistical interpretation of the model's data proves that the three out of the

four models are applicable for other datasets and their accuracies are justified to be able to be used for real world scenarios.

5. RELATED WORK

This research paper standardizes and compares various sentiment analysis lexicons, being Textblob, W-WSD, and SentiWordNet, resulting in the conclusion that using Support vector models and a Naive Bayes classifier. Using an API, researchers were able to grab tweets and preprocess them to remove URLs and special characters in order to assess their sentiments using multiple models. In relation to the present paper, which originally also used a twitter dataset, the usage of sentiment analysis for twitter proved to be inconsistent and therefore disregarded in the research due to the problems mentioned in this study. Additionally, for political bias, datasets that use tweets are often too inaccurate due to the potential that the tweets do not actually carry political information, which would make scraping them much more difficult and tedious. This study proves that the decision not to utilize sentiment analysis in the present paper was ultimately a useful way to prevent inaccuracies that sentiment lexicons contain to further test the legitimacy of other data science models [10].

This research paper explores media bias on gas drilling the Netherlands, using machine learning models as well as analyzing other factors than linguistic analysis such as media attention by referring to four different newspapers over a long course of time, resulting in findings that not only is media bias present, but is also that risk and the dramatization of the media bias are disproportionately related. Similar to the present paper, the flowchart of preprocessing data and then feeding into a model was utilized. However, this paper used manual validation of the final model in the active learning phase, as well as found different types of media bias over the course of multiple years rather than just analyzing media bias in the present day [11].

This research paper uses a recurrent neural network (RNN) to identify conservative and liberal political bias using a similar dataset as the bigram model in the present paper. The key differentiating aspect is that rather than using bigrams to get over the challenges of gleaning through a dataset containing nonsensical sentences that don't have any inherent bias, the authors handpicked over 4000 data points to use in their neural network. Additionally, by using a long short term memory variant of a recurrent neural network, the researchers were capable of generating a F1 score of around 0.7, which is fairly standard for linguistic models. This variation was also a key difference as it was mentioned that the researchers also were unable to accurately distinguish liberal and conserative bias without the model remembering long term dependencies [12].

This research paper uses a multilayer perceptron model (MLP) to tackle the problem of identifying the political ideology that aligns with possible bias in the articles. They found that the MLP produced about a 9% higher F1 score than RNN and claimed that MLP does in fact perform linguistic analysis better than an RNN model. Rather than using vectorizer pipelines like in the current paper, this paper uses word vectors to numericalized the data and the training also utilized stochastic gradient descent to train the method. Additionally, this model was adapted to a google chrome extension, but the results were fairly inaccurate resulting in almost all articles returning 100% neutrality as well as sometimes misidentifying the bias ideology showing that although the models work in practice with the dataset, live usage of the model is fairly inaccurate [13].

This research paper analyzes whether tweets are opinionated and which political ideology they

fall under if they are. However, this research paper does not physically make use of supervised machine learning models and only uses sentiment analysis corpuses and a usage of trigrams to find results of specific types of opinions. The research paper, however, does mention the future work of using a multilayer perception but notes the hardships the researchers may face since there is no easily accessible corpus for analyzing tweets, especially since so many of them are completely irrelevant to training data science models. Much like the current paper, the usage of tweets and other personal information was very quickly discarded in earlier prototypes because the usage of them for data-driven techniques is rather ineffective [14].

This research paper provides a full faceted approach to identifying political bias using a large majority of datasets online as well as using TF-IDF pipelines to preprocess the data into supervised machine learning models as well as using neural networks. Additionally, the researchers also utilize linguistic theories and use word vectors to find similarities across political bias. An interesting application of the results is the standardization of datasets within the field of work, where the LIAR dataset used in the current paper achieved one of these highest accuracy scores along with another dataset from Fake News Net. This paper finds that across all the recent studies, the best approach is to use a model that is capable of long term learning if one is using a neural network and using hand crafted, manual methods for datasets actually does drastically improve the accuracy of models [15].

6. CONCLUSION AND FUTURE WORK

The scientific findings coupled with the final product does present convincing evidence that linguistic analysis of political news can be used in conjunction with supervised machine learning to identify political bias and falsehoods, but can not identify what ideology those bias and falsehoods may represent. Given that the two falsehood models and the annotations model both did significantly better in statistically supported accuracy than the bigram model, it can be drawn that in order to identify ideological bias, there needs to be other factors incorporated like the media outlet a sentence is from. These findings also standardize supervised machine learning models across each other, showing that both logistic regression and random forest classifiers are well fit for linguistic analysis on possibly biased data. Overall, the data from the experiment does prove the possibility that data science can be adequately applied to fake and biased news, a very prevalent problem in an era where media has become an integral part of people's lives.

6.1. Limitations and Future Work

The current models proposed in the paper suffer from some level of inaccuracy due to the absence of a more robust dataset. Datasets for falsehoods and bias, although present, still are not only too minimal in size to fully overcome the problems of overfitting, but also present only a qualitative estimate of falsehood and bias and not a quantitative estimate of both metrics. All datasets used in this paper use labels such as “false” or “barely true,” which is ultimately subjected to arbitrary brightlines and can be interpreted differently in different experiences. A fully robust dataset that could be produced through continuous web scraping through news aggregators using a mathematical formula to determine bias is a possible solution to the lack of Datasets.

Another limitation faced was that accuracy plateaued at around 70-80%. Therefore, more robust models like the usage of a long term learning for a multilayer perceptron could compensate for weaknesses found in supervised machine learning models. However, the usage of a TF-IDF pipeline still proves to be a very simple but effective method for numericalizing data to find linguistic patterns, but the major difference in the future work is the model doesn’t respond to punishment as a way of learning mistakes but remembers patterns useful later on. The more

robust model and a newer dataset may be the most effective way of analyzing linguistic patterns in bias, but in order to fully replicate human behavior, models would need to be trained also with other data independent from the text and may need to consider the publication, author, date, political climate, and ideologies during the current time.

Additionally, it may prove beneficial to focus on specific events that multiple news agencies of different political alignment report on, such as entire presidency administrations, COVID-19, or other major events that could provide a more standardized view of reporting as the base facts would be consistent and therefore do not need to be evaluated. This observation would therefore open doors to pure semantic analysis as a means of determining political leanings especially when events are correlated with actions done by a specific US administration.

REFERENCES

[1] A. Alesina, A. Miano, and S. Stantcheva, "The polarization of Reality," NBER Working Paper Series No. 26675, 2020

[2] Levy, Neil. "The Bad News About Fake News." Social Epistemology Review and Reply Collective 6, no. 8, pp .20-36, 2017

[3] S. Chandler, "This website is using AI to combat political bias," Forbes, 17-Mar-2020. [Online]. Available: https://www.forbes.com[Accessed: 04-May-2022].

[4] Zach, "The 6 assumptions of logistic regression ," Statology, 13-Oct-2020. [Online]. Available: https://www.statology.org/assumptions-of-logistic-regression/. [Accessed: 04-May-2022].

[5] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts." Proceedings of EMNLP, pp. 327--335, 2006

[6] M. Iyyer, P. Enns, J. Boyd-Graber, and P. Resnik, "Political ideology detection using recursive neural networks," Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014.

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[8] IBM, "What is supervised learning?," IBM, 19-Aug-2020. [Online]. Available: https://www.ibm.com/cloud/learn/supervised-learning. [Accessed: 06-Jun-2022].

[9] W. Y. Wang, "'Liar, Liar Pants On fire': A new benchmark dataset for fake news detection," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017.

[10] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine Learning-Based Sentiment Analysis for Twitter Accounts," Mathematical and Computational Applications, vol. 23, no. 1, p. 11, Feb. 2018, doi: 10.3390/mca23010011.

[11] L. Guo, C. Su, Sejin Paik, V. Bhatia, V. P. Akavoor, G. Gao, M. Betke, D. Wijaya. "Proposing an Open-Sourced Tool for Computational Framing Analysis of Multilingual Data." Digital Journalism 0:0, pp 1-22, 2022.

[12] M. Arkajyoti. "Political Bias Analysis." Stanford University Computer Science, 2016.

[13] M. Vu. "Political News Bias Detection using Machine Learning." Department of Computer Science at Earlham College, 2016.

[14] D. Maynard and A. Funk, "Automatic detection of political opinions in Tweets," Lecture Notes in Computer Science, pp. 88–99, 2012.

[15] R. Oshikawa , J. Qian , W. Y. Wang, "A Survey on Natural Language Processing for Fake News Detection," Proceedings of the 12th Conference on Language Resources and Evaluation, pages,pp 6086–6093, 2020.