Predicting media bias of news articles using deep-learning

Master's Thesis

for acquiring the degree of Master of Science (M.Sc.)

in Economics

at the School of Business and Economics
of Humboldt-Universität zu Berlin

submitted by:

Tobias Krebs

Student no. 585428

Examiner: Prof. Dr. Stefan Lessmann

Berlin, September 21, 2020

Contents

1 Introduction

There is evidence for a high political polarization of media consumption in the US that seems to have even increased over the past five years (Jurkowitz et al., 2020). Directly related to this, is the issue of media bias, i.e., the leaning of a news source towards one political direction. An unbalanced consumption of such biased media can increase polarization and lead to electoral mistakes in form of the election of the wrong candidate (Bernhardt et al., 2008). In order to counteract this problem, this bias first needs to be detected and assessed. This thesis leverages new techniques in machine learning to derive both the prevalence and degree of bias across media outlets.

A model that is able to detect the political bias of news articles could be applied in several ways. First, it could contribute to a more balanced presentation of articles on news aggregator websites, such as Google News (Hamborg et al., 2019). Second, such a model could also be applied by news aggregators or similar to the approach of Munson et al. (2013) in form of a browser widget to asses bias directly and present it to the reader in real time. This might encourage readers to a more balanced news consumption and, due to a higher diversity of views, reduce the probability of producing "filter bubbles"(Flaxman et al., 2016). Third, it could improve media bias related research in the social sciences, that is often relying on simpler methods, e.g., key word searches, to conduct a faster and more precise analysis of a larger text corpus (Hamborg et al., 2019). Last, such a model could be included in the process of news fact-checking, since extreme left and right outlets tend to score low on factual reporting (Baly et al., 2018).

The goal of this paper is to offer insights into the capabilities and limitations of deep-learning models to predict the political bias of news articles. As the main model for the analysis, I choose BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018), as it is a machine learning model that is both comparatively easy to apply and powerful — having been proven to perform well on other text classification tasks. I apply it on data extracted from the dataset of Nørregaard et al. (2019), to predict one of five bias classes, ranging from left to right, and thereby, extend the typically binary or three class categorization used in the literature. Moreover, I shed light on potential issues with source specific labeled datasets, which are neglected in past studies. I also compare BERT to two alternative approaches that are typical for methods applied in the media bias prediction field, regarding performance and computational cost, to allow for a cost-benefit analysis. Another aspect I investigate is whether such a model is able to predict the correct political bias of articles of news outlets it is not trained on. Finally, I present examples of articles and provide insights into why the model categorizes these texts into certain bias classes by applying the software package LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al., 2016). This thesis thus offers a deeper understanding of model decisions and potential reasons for model failures than the current literature does. The code for data preparation and all empirical experiments is provided on GitHub.[1]

[1] https://github.com/Tobias-K93/media-bias-prediction

2 Literature review

Analyzing news articles with the use of machine learning is done in several ways. In this work, the focus lies on political bias prediction which is a rather new and understudied field in machine learning that tries to classify news articles regarding their political leaning. For a comprehensive overview, especially in relation to the work done in the social sciences to the year 2018, see the literature review of Hamborg et al. (2019). A related field that has been studied for a slightly longer time period is sentiment analysis of news articles. The goal is to predict the sentiment, often expressed in the categories positive and negative, of a news article towards a specific topic (Godbole et al., 2007; Balahur et al., 2013). This kind of analysis often focuses on economic and financial terms (Li et al., 2014; Shapiro et al., 2020). Even more closely related, however, is the area of fake news detection that targets the trustworthiness of news articles (Ahmed et al., 2017; Pérez-Rosas et al., 2017). This is why some authors include both, fake news and political bias prediction into their studies, as politically extreme leaning news outlets tend to also score low on factuality (Baly et al., 2018). Thus, many of the following aspects about the media bias literature are also relevant for studies focusing on fake news.

The terms used in the literature sometimes differ from the political media bias terms used in this thesis. While there are fine-grained differences in the meaning of terms like propagandistic content (Barrón-Cedeno et al., 2019), hyperpartisan news (Potthast et al., 2017), or political ideology containing articles (Kulkarni et al., 2018) — the data, as well as the problem settings described in each work suggest that they can be combined into one political bias category. Table 2.1 presents an overview of all relevant political bias literature. It is divided into two major aspects that define and distinguish the work done in this field, data and algorithms. Note that the last four entries are participants of the Semantic Evaluation 2019 competition (Kiesel et al., 2019). To keep a clear overview, I only present the three best performing teams, as well as the submission that is closest related to this work (Mutlu et al., 2019).

Regarding datasets, there are two methods that researchers use to label news articles or other forms of text. One way is to use labels that indicate the political leaning of a news outlet and assign these labels to its corresponding articles. This so called weak labeling requires some entity, e.g., Media Bias/Fact Check[2] (MBFC) or Allsides[3], that offers bias labels for each news source in a dataset. The other prominent way in the literature requires manual labeling, i.e., a person that categorizes an article or text fragment according to a set of rules. This can be done either by crowd sourcing annotators on platforms like Amazon Mechanical Turk[4], or by hiring professional annotators (Da San Martino et al., 2019). Depending on the method that researchers choose, sizes of datasets vary immensely. Due to the high cost of hiring annotators, manually labeled datasets are quite small and range from 300 to 1,627 news articles, excluding

[2] https://mediabiasfactcheck.com
[3] https://www.allsides.com/unbiased-balanced-news
[4] https://www.mturk.com/

Political bias studies	Data						Algorithm			
	dataset size*	weakly labeled	manually labeled	label source	target level	# target classes	main algorithm**	deep-learning	linguistic features	model insights
Iyyer et al. (2014)	(3,412)		X	crowd	sentence	3	RNN	X		
Budak et al. (2016)	10,502	(X)	X	crowd	article	3	LR		X	
Potthast et al. (2017)	1,627		X	BuzzFeed Journalists	article	3	RF		X	
Baly et al. (2018)	94,814 (949)	X		MBFC	source	3/7	SVM		X	
Kulkarni et al. (2018)	120,000	X		Allsides	article	3	CNN/ GRU + Attention	X		attention visualization
Baly et al. (2019)	94,814 (949)	X		MBFC	source	7	COR		X	
Barrón-Cedeno et al. (2019)	22,580/ 51,300	X		TSHP-17/ MBFC	article	2	Max Entropy		X	
Da San Martino et al. (2019)	451 (7,485)		X	professional annotators	text fragment	2	BERT	X		
Fan et al. (2019)	300 (1,727)		X	3 annotators	text span	2	BERT	X		
Horne, Nørregaard, et al. (2019)	158,500	X		NewsGuard/ Open Sources	article	3	RF		X	
Kiesel et al. (2019)	Semantic Evaluation 2019: Hyperpartisan news detection									
Jiang et al. (2019)	750,000/ 645	X	X	MBFC& BuzzFeed/ crowd	article	2	CNN + ELMo	X		
Hanawa et al. (2019)	750,000/ 645	X	X	MBFC& BuzzFeed/ crowd	article	2	linear + BERT Embeddings	(X)	X	
Mutlu et al. (2019)	750,000/ 645	X	X	MBFC& BuzzFeed/ crowd	article	2	BERT	X		
Srivastava et al. (2019)	645		X	crowd	article	2	LR + USE Embeddings	(X)	X	

* Figures in parenthesis present numbers of target items, when targets are not articles.
** RNN: Recurrent Neural Network, LR: Logistic Regression, RF: Random Forest, SVM: Support Vector Machine, CNN: Convolutional Neural Network, GRU: Gated Recurrent Unit, COR: Copula Ordinal Regression, BERT: Bidirectional Encoder Representations from Transformers, ELMo: Embeddings from Language Models, USE: Universal Sentence Encoder

Table 2.1: Work related to political bias detection with machine learning

Budak et al. (2016) that use a mixture of both techniques. This limits the applicability of algorithms that usually require big datasets, like deep neural networks. Weakly labeled datasets, on the other hand, offer comparatively cheap big datasets ranging from 22,580 to 750,000 articles. This, however, comes at the cost of more noise and potentially wrong labels, which I address in section 4.1.

Another potential difference concerning datasets is the target of the analysis. A majority of studies focuses on categorizing articles, however, there also exists work that targets a lower level of the article such as sentences and text fragments, or a higher level such as the sources. It is worth mentioning that all lower level target studies have manual labeling in common, as well as the application of deep-learning algorithms. This is due to the needed precision in labeling, that can only be provided by manual annotators. Similarly, the linguistic features applied in research without deep-learning models cannot deliver the same distinguishing low level information as the embeddings used as inputs for the deep-learning models.

Besides the level, target variables differ also in the number of classes. Most researchers classify political bias into two or three categories. In the binary case, this implies one class that represents little or no bias, and another one that represents high bias. Most three-class cases

use a form of left/right/center categorization in which the center class often coincides with little or no bias, and the other two classes with the (often strong) respective political leaning. Horne, Nørregaard, et al. (2019) diverge from this pattern by using the three classes mainstream-, unreliable-, and biased-sources. Baly et al. (2018) and Baly et al. (2019) are the only examples that use seven bias classes. However, they predict the bias of a source, not an article. Hence, there is a potential for more accurate analysis of news articles by expanding the number of political bias classes, which allows a researcher to utilize all information contained in datasets offered by sources like MBFC or Allsides.

There is no clearly preferred applied algorithm in the literature. Various models based on deep-learning techniques and such that are not based on deep neural networks are almost equally represented. Hanawa et al. (2019) and Srivastava et al. (2019) choose approaches that are somewhere in between, since they use deep-learning based embeddings that are pre-trained. However, they do not train them themselves, but rather use them as static inputs, which combined with linguistic variables form the basis of their simple machine learning algorithms. In two cases the BERT algorithm is used for text fragment based analysis (Da San Martino et al., 2019; Fan et al., 2019), and once, similarly to the approach in this thesis, fine-tuned for article based bias prediction (Mutlu et al., 2019).

The only contribution to the literature that offers model insights that surpass simple ablation studies is the work by Kulkarni et al. (2018). They utilize attention scores to visualize which words and sentences the attention layer of their model focuses on. While this potentially offers interesting insights, it is not clear whether visualizing attentions is a successful way to explain a model (Jain and Wallace, 2019; Serrano and Smith, 2019).

From this review, several aspects of the following study can be inferred. So far, none of the presented literature employed a dataset utilizing five different political bias classes and compared the performance of several typically used algorithms on it. Moreover, there is only little work done on offering insights into why models predict certain labels on certain articles. None of it uses methods that are independent from attention layers, like the LIME model employed in this thesis.

3 Method

3.1 Models

3.1.1 BERT

The main model that I use throughout this study is BERT by Devlin et al. (2018). It is an advancement of the attention based transformer architecture firstly introduced by the "Attention is all you need" paper of Vaswani et al. (2017). Devlin et al. (2018) offer two versions, base and large, that differ in the number of layers and their sizes. I opt for the smaller base version that consists of twelve layers of which each layer includes 768 hidden units and twelve attention heads that sum up to a total of 110 million parameters.

There are several reasons why applying BERT is a reasonable approach to predict media bias in this setting. First, the authors themselves, as well as other researchers (Sun et al., 2019), show that it is well suited to solve text classification tasks similar to the one applied in this work. The Stanford Sentiment Treebank (SST-2) benchmark test in the original paper shows a big improvement in sentiment prediction, which is closely related to media bias prediction (Hamborg et al., 2019). Second, the fact that its architecture was pre-trained on a large corpus of texts, combined with the option to fine-tune parameters for only a few epochs on the specified task, entails a powerful model with comparably short training time for its size. Third, the algorithm introduces mostly simple but effective concepts, like an improved bidirectional structure for transformers using a masked language model. This not only makes the task of understanding the architecture easy, but also simplifies its task specific implementation.

As suggested for classification tasks, I use the final hidden vector output of BERT's CLS token. On top, I apply a typical classification output layer with TanH activation and 10% dropout to classify inputs according to their target labels, as described in the original paper. Further, Devlin et al. (2018) apply word piece tokenization (Wu et al., 2016) to the input texts, which I also implement. Compared to normal word tokenization, this allows for a much smaller vocabulary, as rare words are split up into pieces instead of added as a whole to the vocabulary. The word piece vocabulary of BERT in this case consists of 30,522 tokens.

3.1.2 Benchmark Models

3.1.2.1 Single Headed Attention BiLSTM

While transformer based models like BERT have been successful in classifying texts, they often come at a high computational cost, and especially high memory usage. Moreover, non-transformer based deep-learning architectures like CNNs or forms of RNNs are used in the media bias literature (Kulkarni et al., 2018; Jiang et al., 2019). Thus, it makes sense to compare the results of BERT to those of a non-transformer based deep-learning model, such as the

one developed by Merity (2019). His Single Headed Attention (SHA) RNN is presented as an efficient alternative to transformer models, as the benefits of many multi-head attention heads, such as those used in the BERT architecture might not be worth the high memory requirements.

The main difference between my benchmark model and the one of Merity (2019) is that he developed it for language modeling, while I use it for text classification. Due to that, I adjust some factors to match the application. First, I use bidirectional LSTM layers with 2 x 512 neurons, instead of a one directional layer with 1024. This allows the model to get a better contextual understanding of the input and has shown to improve predictions in classification tasks (Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005). Second, I add a classification layer at the end that fits my task of predicting five classes, instead of predicting the next word in a text. Last, the original paper suggests to use something described as continuous cache (Grave et al., 2016), which stores past hidden states as memory. This is a useful technique for language modeling, where the goal is to predict the next word depending on the past words in a text. In the present setting, however, the classification of one text does not depend on the target classes of past texts. Thus, there is no benefit in including this sort of memory, which leads me to exclude it from the model. Independently of considerations regarding the classification task, I use the same word piece tokens as for the BERT model, as it allows for a better comparison of the two models, while at the same time being a possible improvement suggested by Merity (2019).

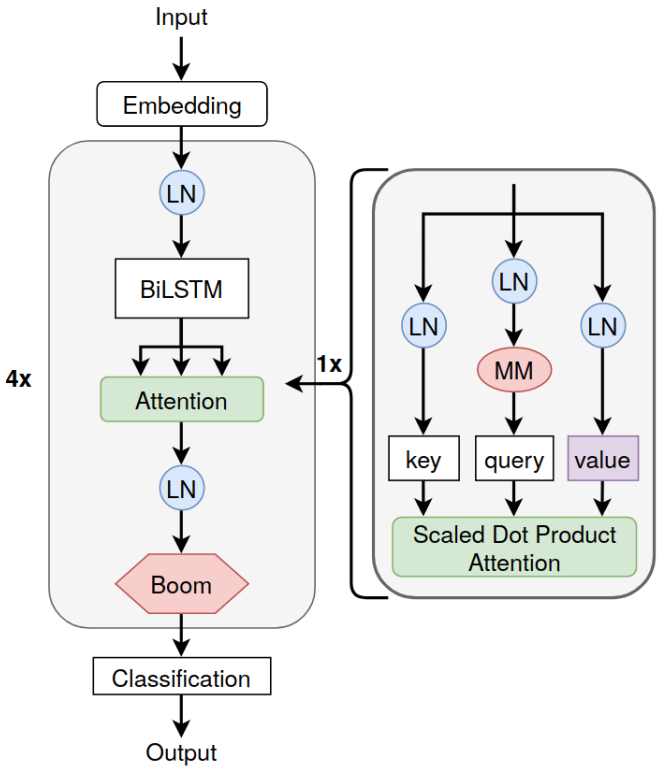


Figure 3.1: Architecture of BiLSTM Benchmark model

Figure 3.1 shows the model architecture. Besides the mentioned exceptions, it closely follows the original model. At the beginning, input embeddings are trained. The main part consists of four stacked blocks, which include a BiLSTM layer with layer normalization (LN) (Ba et al., 2016) before and after, followed by the so called "Boom" layer — a feed forward layer similar to the ones used in Transformers that consists of four times the amount of neurons used in the hidden layer. Only in the third block, is a *single* attention head added between BiLSTM and "Boom" layer. To each input of the attention head, layer normalization is applied. Only the query input is multiplied afterwards with a parameter matrix (MM). Next, each input (key, query, and value) is multiplied with a parameter vector before scaled dot product attention, as described in Vaswani et al. (2017), is applied. Different from the key and query operation, the value operation includes an additional over-parameterized-component (Merity, 2019), i.e. applying the following equation to the parameter vector:

$$output = sigmoid(W^f\, vector) \cdot TanH(W^c\, vector)$$

where W^f and W^c are parameter matrices to produce a forget gate and candidate respectively. In a last step, after all four blocks, the linear classification layer with soft-max activation is put on top. The resulting model consists of approximately 61 million parameters.

3.1.2.2 Random Forest with linguistic features

Despite the vast improvements of deep-learning applications for NLP tasks in recent years, more traditional machine learning algorithms, like logistic regression, support vector machines, or random forest (RF) are still popular among researchers analyzing media bias, as the overview in Table 2.1 shows. In most cases, researchers combine these algorithms with a comprehensive set of manually crafted features. Thus, similar to Horne, Nørregaard, et al. (2019) and Potthast et al. (2017), I use a RF model on top of linguistic variables as a non-deep-learning benchmark for comparison with both deep-learning based approaches described above.

The RF algorithm offers comparatively low computational cost and is easy to apply without much time needed to set it up, which is an obvious advantage. However, selecting and creating the linguistic features is time consuming, and either requires some experience with common practices, or even more time to familiarize oneself with it and the linguistic literature. For this study I applied a combination of features mainly taken from Recasens et al. (2013), Barrón-Cedeno et al. (2019), and Horne, Nørregaard, et al. (2019). The total number of 54 variables can be roughly divided into character based, vocabulary richness, dictionary, and part of speech (POS) features. All features are count variables, i.e., for each article, entities like characters, specific word tokens, or sentences are counted, and in some cases, combined to new variables. A complete list of all features can be found in Table A.1 of the Appendix. I only include variables that can be created without knowledge of the original news source, since the model is supposed to predict the political leaning of an article independently from its source, which is the case for

both deep-learning models as well. Because of this, I do not include whether a source has a Wikipedia page, as Horne, Nørregaard, et al. (2019) do.

3.2 Data

The main dataset I use throughout my analysis was produced by Nørregaard et al. (2019). It originally consists of about 713,000 news articles directly collected from each news producer's website, between February and November 2018. Moreover, it covers 194 news outlets and eight different sources for ground truth labels concerning reliability, bias, transparency, and consumer trust. All ground truth labels are weak labels, in the sense that articles are labeled on a news outlet level not a text level.

Of these labels, three cover political leaning or bias, namely Media Bias/Fact Check (MBFC), Allsides, and BuzzFeed. The BuzzFeed variable seems to be the least appropriate for the task for several reasons. First, it offers only two categories, left and right, which limits the insights of an analysis. Second, in contradiction to the other two sources, BuzzFeed itself is a typical news website that is labeled as moderately biased by both other sources, whereas Allsides and MBFC offer strictly methodological bias categorization as a main focus of their product. Third, with only 56 outlets it covers the least amount of news sources. Both remaining ground truth label sources are well suited and have been used before for similar tasks (see Table 2.1).

MBFC offers the most articles and news outlets, 498,911 and 108 respectively, where each news outlet is assigned one of seven labels ranging from extreme left to extreme right. Allsides, on the other hand, assigns bias labels to 286,235 articles and 65 outlets in this dataset and offers five different labels, ranging from left to right. After first considerations of using a combination of both labels, I end up only using the labels offered by Allsides. The reason to not use a combination is that their chosen labels in many cases do not coincide, with the lack of overlap often being in cases either where MBFC or Allsides labels one outlet further to the right or left than the other. This is mostly due to different scale sizes, but potentially also because of different methods to assign bias labels. While MBFC uses mostly volunteers to help edit their labels with the help of a defined methodology[5]; Allsides offers a range of methods that includes blind bias surveys, editorial reviews, third party analyses, independent reviews, and community feedback[6]. Furthermore, MBFC includes factual reporting into their bias score calculations, while Allsides solely focuses on political bias, which is the focus of this thesis.

The reasons to choose Allsides over MBFC are the following: Although MBFC's bigger corpus could be beneficial to the analysis, this is partly due to the fact that it includes sources from non-English speaking countries: Sputnik (Russia), France24, Spiegel (Germany), and Telesur TV (Venezuela), which possibly adds noise in the form of different coverage and writing styles across countries. This leads Horne, Nørregaard, et al. (2019) to exclude all non-US sources

[5] https://mediabiasfactcheck.com/methodology/
[6] https://www.allsides.com/media-bias/media-bias-rating-methods

from their analysis. The main advantage of the Allsides corpus, however, is that it is more balanced than MBFC's, and its five category scale — left, lean left, center, lean right, right — is a reasonable middle way between enough insight and a low number of labels. Figure 3.2 shows the relative balance of the Allsides dataset in comparison to the MBFC dataset. Moreover, its size of almost 250,000 articles (after data cleaning) is still big enough for a thorough deep-learning analysis.

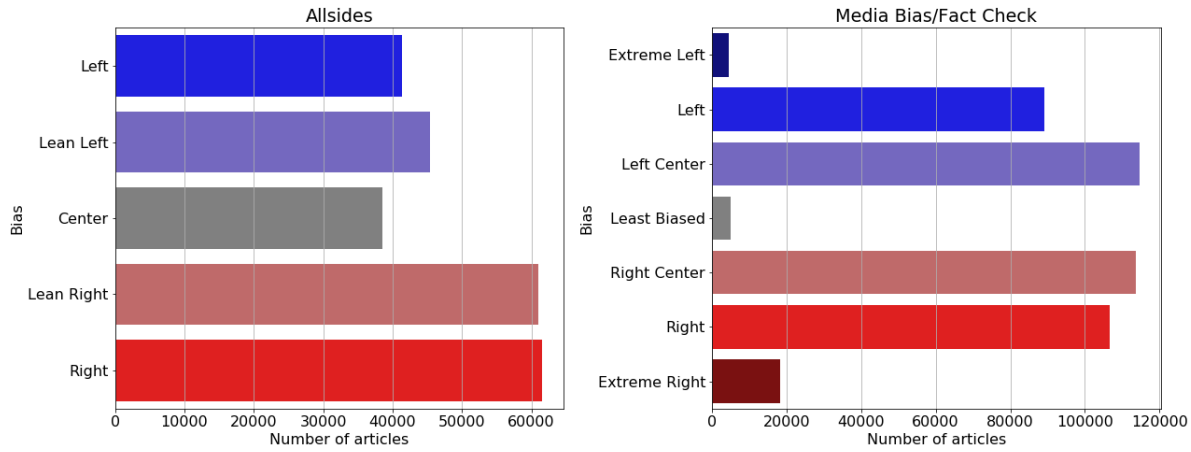


Figure 3.2: Distribution of Allsides and MBFC political bias labels (after data cleaning)

Although Nørregaard et al. (2019) already performed data cleaning, some cleaning tasks remain. In a first step, I remove some articles without content (218) or a title (3). Further, it seems that in the process of scraping news articles from the websites' RSS feeds, not only single whole articles were collected. In some cases, error messages, short descriptions with links to full content, or news unrelated content like commercials were collected. In rarer cases, several articles are combined together resulting in unusually long texts. To reduce this kind of noise in the data, I exclude all particularly short and long entries. After exploring and testing different cutoffs, I remove all articles shorter than 500 characters (~100 words) and longer than 20000 characters (~4000 words) from the corpus, which roughly coincides with typical word counts of news stories[7]. This results in a reduction of 36910 and 1391 entries, respectively. Another aspect is the wrong labeling of one news source. The Allsides labels of the present dataset assigned the outlet "Right Wing Watch" a right bias, while MBFC and BuzzFeed assigned it a left bias in the same dataset. Moreover, Allsides currently does not label this source on their website, which leads me to remove all 793 articles. In a last step, I fix some wrongly coded signs from html to unicode and remove all http- and https-links from the content. This reduces noise, and in some cases avoids giving away the source of the content, e.g. in the case of the source "bearing arms" and the link "https://bearingarms.com/author/davidl/". Altogether, data cleaning leads to a reduction of 39,315 articles, resulting in a final number of 246,920.

[7] https://www.journalism.org/2005/12/12/the-stories-short-wire-copy-versus-original-reporting/

4 Experimental Setup

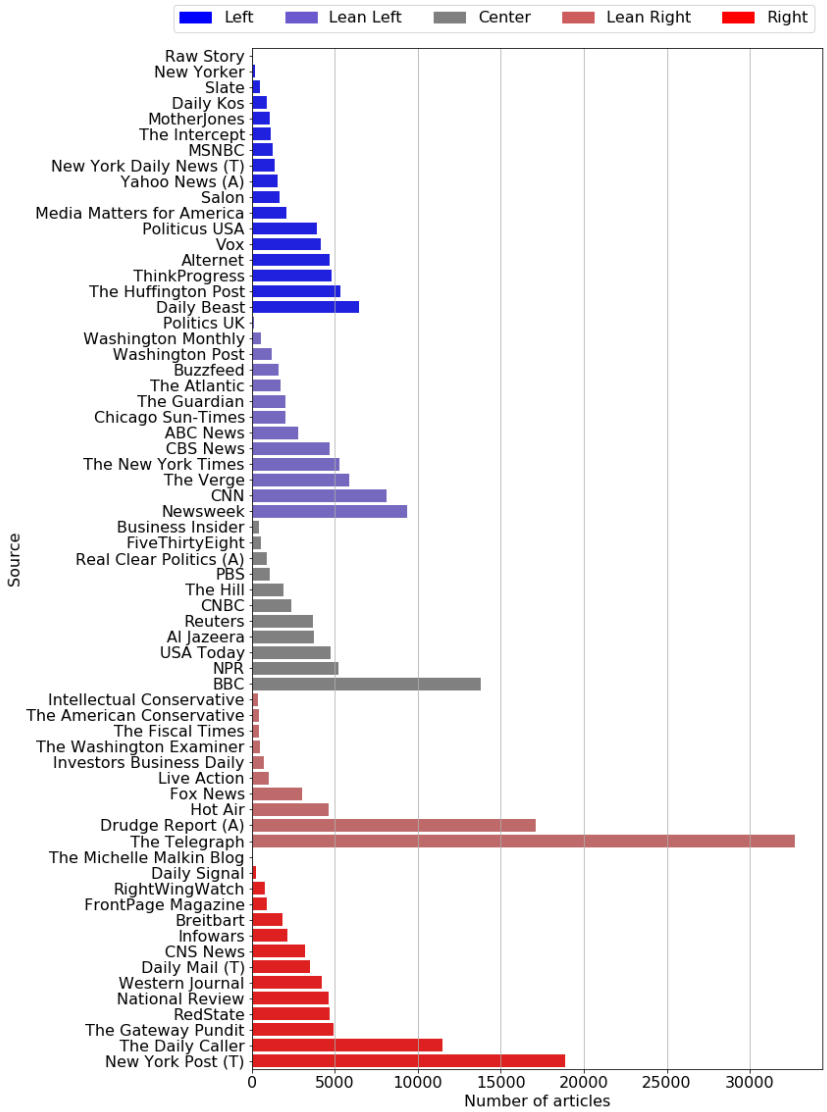
In chapter 5 below I conduct several experiments. For all these experiments accuracy and F1 score are presented. For the multi class case, I choose to calculate the macro F1 score, since it weighs all five possible bias labels equally. This seems appropriate in this case because all labels are of equal importance for the analysis. Moreover, a comparison of accuracy and macro F1 score offers insights into whether an unbalanced dataset has a negative effect on the model's predictions, since accuracy weighs more frequent classes higher than the macro F1 score. Moreover, I report the mean squared error (MSE) due to the fairly ordinal nature of the given problem (Baly et al., 2019). Depending on the goal of the application, it matters whether a right labeled article is categorized as moderate right or as left. Comparing MSE scores can indicate whether a model rather predicts bias labels further away or closer to the true value. To reduce the randomness of my results, i.e., lower the variance of presented metrics, I further conduct all experiments three times and take the average of these three runs for each metric.

4.1 Applied datasets

The data described above in section 3.2, even after general data cleaning, still contains aspects that could confuse the model during the training process in an unintended way. Thus, I remove or alter certain articles and compare results between these alternative datasets and the original dataset. The model I employ on these datasets is the BERT model described in section 3.1.

Figure 4.1 shows all news sources represented in the original dataset, i.e., the dataset after data cleaning and before the removing of certain articles described below. It presents the number of articles of each source as well as each source's bias label. Whether a news source belongs to the news aggregator or tabloid category (explained below) is indicated with an (A) or (T), respectively. What is clearly observable is the wide range of frequencies between news sources, ranging from only 52 articles of "The Michelle Malkin Blog" up to 32,715 of "The Telegraph".

The first issue that potentially affects bias prediction are news aggregators. While none of the news sources in the dataset are pure news aggregators, some offer a lot of articles that originate from other news outlets next to their original content. This leads to a wide diversity of political views presented on these news aggregator websites from sources that are often labeled with opposing biases. The fact that one source consists of articles from not only several different bias categories, but also outlets with differing biases (that further are separately represented in the dataset) makes it impossible for a model to distinguish between bias types within the same source. This means that an article of a news aggregator, if it is originally from another news source with a different bias label than the news aggregator itself, can be seen as a wrongly labeled article. Since there is no indication in the dataset itself, whether an outlet is a news aggregator or not, I looked up the description of each source on Wikipedia and the source's website to categorize them into aggregators and non-aggregators. Following that, I exclude the



(A) = News Aggregators, (T) = Tabloids

Figure 4.1: Distribution and labeling of news sources

news aggregators “Drudge Report”, “Real Clear Politics”, and “Yahoo News” from this separate dataset, which leads to a reduction of 19,578 articles.

Another group of news sources that potentially causes problems for a model to correctly predict political bias are news sources that conduct tabloid journalism[8]. These kind of outlets tend to release fewer articles about political topics when compared to other news outlets, and instead more about sports, TV shows, or celebrities. The high amount of non-political content introduces a lot of noise into training of a model that is supposed to predict political bias. As a consequence, instead of the actually targeted political leaning of an article, a model might connect these non-political topics to a bias label. To quantify the different amount of political content between tabloid and non-tabloid outlets, I randomly select 100 articles, respectively, from each of both types and manually categorize them into political and non-political content. I regard

[8] “type of popular, largely sensationalistic journalism”, https://www.britannica.com/topic/tabloid-journalism

content as political if it mentions at least one of the following: politicians; political institutions like governments, parliaments, or courts; or politically relevant topics like climate change, war, or terrorism. As a result, 70 out of 100 non-tabloid articles contain political content, while for tabloid articles it is only the case for 35 out of 100. This wide gap of 35 percentage points supports the assumption of fewer political articles from tabloid outlets. Whether a news source is considered to produce tabloid journalism or not is determined similarly to the news aggregator case by gathering information from Wikipedia and original news website. Following this procedure, I remove "New York Daily News", "Daily Mail", and "New York Post" from this dataset, which results in a reduction of 23,732 articles.

So far, all changes to the dataset focus on removing specific groups of sources to facilitate model training. Another approach I apply, focuses on altering articles instead of removing them. Many news websites frequently include sentences in their articles that are not necessarily related to the content. Usually found at the end of the article, these sentences may promote the news outlet's social media appearance, give information regarding the author, or present other standardized information. For examples see Table 4.1. Having an identical sentence in many articles of the same source potentially allows the model to learn a connection between this sentence and the bias label of the article's source. Ideally the model should learn typical characteristics of each political bias label independently from the article's source. Thus, to prevent the model from making these undesired connections, I remove sentences that are frequent within articles of individual sources. The removal of sentences is limited to sentences with more than twelve characters, to avoid removing common short phrases like "No.", "Yes.", or "He added.". Moreover, I adjust the cutoff that rates a sentence as frequent depending on the number of total articles per source in the dataset, ranging from 4 to 50 appearances. While this procedure most likely will not increase model accuracy, it should improve confidence in the ability of a model to correctly predict media bias based on bias related characteristics in the text, rather than unrelated frequent sentences.

Sentence	Source	Frequency	Articles
Follow us on Facebook, on Twitter @BBCNewsEnts, or on Instagram at bbcnewsents.	BBC	418	13818
This is a developing story.	BuzzFeed	28	1593
Click to subscribe!	CNN	87	8143
You can get it now on Amazon.	Daily Kos	114	924
The Associated Press contributed to this report.	Fox News	492	3045
Paul Joseph Watson is the editor at large of Infowars.com and Prison Planet.com.	Infowars	262	2171
Click here for more Commentary and Opinion from Investor's Business Daily.	Investors Business Daily	153	729
Todays installment of campaign-related news items from across the country.	MSNBC	126	1240
Reuters has not edited the statements or confirmed their accuracy.	Reuters	115	3670
If you have an idea for a piece, pitch us at thebigidea@vox.com.	Vox	98	4183

Table 4.1: Examples of sentences that are frequently appearing in articles

Each of the three aforementioned alterations (removing news aggregators, tabloids, and frequent sentences) is applied individually to create a dataset, and then all three are combined to

create a fourth. This allows for the effects of the methods to be compared against one another. Since this fourth dataset most likely contains the least amount of noisy and wrongly labeled data, I use it for all further experiments that compare models or other relevant aspects.

All the above described datasets have in common that they are weakly labeled. Since manually labeled sets are also common in the literature, testing whether a model trained on a weakly labeled dataset is able to predict manually labeled articles could offer some interesting insights. Generally, it could help answering the question whether interchanging models trained with one type of labeling can be applied to another type. Further, it serves as an additional test set to investigate the validity of a model. Unfortunately, there is no available dataset that uses the same five bias labels to assign political leaning to news articles that I use for my analysis. However, Kiesel et al. (2019) allow access to their manually labeled train set with the two classes hyperpartisan and non-hyperpartisan. It was used for their 2019 SemEval competition and consists of 645 articles. By grouping the categories left and right as hyperpartisan, and lean left/right and center as non-hyperpartisan, I can apply a model trained on one of the above mentioned weakly labeled datasets to this dataset as well.

Besides the manually labeled SemEval dataset, which is not used for training, all datasets are randomly split into training, validation, and testing set with a 80/10/10 percent ratio, respectively. Only the validation set is used for hyper-parameter search and other forms of testing, while the test set results are only once determined at the end. This is also the case for the benchmark studies described further below. Note that in all relevant cases, aggregators and tabloids are removed from training, validation, and testing set, as wrong labeling and unpolitical content is also undesired for validation. Frequent sentences, on the other hand, are only removed from the training set, because the model should still be able to predict the bias of an article containing these kind of sentences. It cannot, however, predict the correct label of a wrongly labeled article of the news aggregator category.

4.2 Benchmark studies

The experiments described so far are all conducted using the BERT based model. However, this comes with some disadvantages, especially regarding computational and memory related cost. Moreover, the literature review shows that the majority of recent research has employed other types of models to predict media bias in the news. To allow a comparison between BERT and the two representative benchmark models introduced in section 3.1.2, I conduct experiments with all three models and compare their metric scores as well as the duration time and memory consumption of training. Strubell et al. (2019) recommend to state these values to allow researchers to perform a cost-benefit analysis and to put energy use into considerations when choosing a model.

For all three models, I performed a hyperparameter search. Due to the fact that it is already pre-trained, BERT's hyperparameters are easiest to select. Devlin et al. (2018) suggest only

three parameters to choose from in their fine-tuning procedure. Following that, after manually testing all suggestions, I select a batch size of 16 due to memory limits, a learning rate of 2e-5 for the Adam algorithm, and train the model for 3 epochs. The authors also state that especially bigger datasets, containing more than 100,000 labeled inputs, are less sensitive to the choice of hyperparameters, which I find to be true in this case as well. The tokenized input sequence, created from the news articles, I cut off after 500 word piece tokens. This is necessary, since attention is quadratic to the sequence length which makes longer sequences disproportionately expensive (Devlin et al., 2018). Moreover, the standard BERT architecture limits the length of a sequence to the number of hidden units, which makes cutting off longer sequences necessary even when computational limits are irrelevant. With a median length of about 500 tokens, there should be enough bias information contained in the input sequences used. Further show findings of Fan et al. (2019) that especially lexical bias, a form of media bias that arises from linguistic attributes, appears disproportionately often in the first quartile of an article.

The SHA-BiLSTM from section 3.1.2.1 is not pre-trained and has not been applied to this setting before, which makes a hyperparameter search more necessary than in the BERT case. I mostly experimented with the learning rate and ended up applying the same learning rate as in the BERT case (2e-5) for three epochs, followed by two epochs with the learning rate cut in half to 1e-5 with the Adam optimizer. This is similar to the approach chosen by Merity (2019) who also cuts the learning rate into half for the last two epochs of training. However, the author uses a higher learning rate and more epochs for the training of his language model. Moreover, this approach leads to better results than simply using the same learning rate, 2e-5 or 1e-5, for all training epochs. Due to the lower memory demand of the architecture, I was able to use a batch size of 64. However, for a better comparison, I also present duration and memory figures with a batch size of 16. As it is the case for BERT, I apply a sequence length of 500. First, because it allows for a better comparison of both models especially concerning computation time but also regarding prediction performance. Second, although there are no architectural limitations to the sequence length, increasing it still raises computation cost while likely having only a disproportionally small effect on prediction performance.

The RF algorithm is known to work well with its default hyperparameters. Nevertheless, it can profit from hyperparameter search (Probst et al., 2019). Hence, I conducted a random grid search over the following parameters: number of estimators, maximum depth, minimum samples split, maximum of features, and maximum of samples. This was only possible because of the lower computation time necessary for training RF compared to the above mentioned deep learning models. As a result, only changing the maximum of features from $\sqrt{num\ of\ features}$ to $half\ of\ features$ improved predictions compared to using the default values. The final parameters used are shown in Table 4.2. Due to the different inputs that the RF approach uses, utilizing the complete contents of all articles instead of 500 tokens only affects the time to produce inputs. It does not, however, affect the time to produce predictions. Thus, I create the linguistic variables from complete contents.

Parameter	Value
Number of estimators:	500
Maximum depth:	fully developed
Minimum samples split:	2
Maximum of features:	half of features (27)
Maximum of samples:	100%

Table 4.2: Final hyperparameters of RF model

4.3 Additional experiments

As already mentioned above, the media bias prediction setting with five labels ranging from left to right can be seen as an ordinal problem. So far, however, I do not address the ordinal nature. To change that, I alter the loss function to make it cost sensitive, i.e. it increases the loss more when predicted label and true label are further apart on the left-right-scale than when they are close. For the construction of the cost sensitive loss I follow the approach of Khan et al. (2017). Table 4.3 shows the cost matrix ξ I use in the process which I created according to the proposed properties of Khan et al. (2017). It returns the factor 0.2 for all correctly predicted biases and increases the returned factor the further away the prediction is from the true value.

predicted labels		true labels				
		0	1	2	3	4
left	0	0.2	0.4	0.6	0.8	1
lean left	1	0.4	0.2	0.4	0.6	0.8
center	2	0.6	0.4	0.2	0.4	0.6
lean right	3	0.8	0.6	0.4	0.2	0.4
right	4	1	0.8	0.6	0.4	0.2

Table 4.3: Cost matrix ξ

This factor $\xi_{p,n}$ I then include into the soft-max function of the cross entropy loss in the following way:

$$\ell(\mathbf{d}, \mathbf{y}) = -\sum_{n}(d_n log(y_n)) \qquad y_n = \frac{\xi_{p,n} exp(o_n)}{\sum_k \xi_{p,k} exp(o_k)}$$

with p indicating the true label, k the number of classes, n the number of observations, d_n the desired output, and y_n the soft-max output, which depends on the model output o_n and cost ξ. Implementing cost sensitivity this way does not change the calculation of the gradient, and thus, does not influence the back-propagation process. The loss is only affected via the soft-max output y_n (Khan et al., 2017). Since the goal is not to improve classification accuracy but to adjust training according to the proximity of different target classes to one another, i.e., the ordinal nature, I do not expect the accuracy or F1-score to improve. However, if successful,

introducing a loss function that includes distances between classes into the calculation should improve the mean squared error, a measure of distances.

Another aspect I want to test is whether the model mostly predicts political bias due to patterns it learns from each news source individually within the classes, or whether it is able to find common patterns for each class that allow to generalize. This would determine whether it is able to predict the correct bias class of a news source it was not trained on. To test this, I remove all articles of one source per class from the training set and compare the individual test results of each source with the results when they are included in the training set. In this way, I am able to get a direct comparison of the effect that including articles of a source into the training set has on test results. For a more robust analysis, I choose to use two groups of sources, i.e. conducting two experiments, once with a group of sources that have a comparably low representation in the dataset, ranging from 368 to 745 articles, and once with a group that is more often represented, ranging from 2200 to 3123 articles. All removed sources, their frequencies, and bias labels can be found in Table A.2 of the appendix.

4.4 Prediction approximation with LIME

Although the general methods used in the BERT architecture are known and mostly easy to understand, it is still too complex for a human to interpret how exactly it derives a specific prediction. Thus, it can be seen as a black box model that predicts the political bias of an article without giving an explanation on why it predicts a certain class. One way to gain more insights into why a model predicts a certain outcome is LIME (Ribeiro et al., 2016).

LIME offers information on which inputs, i.e., in this case word piece tokens, are important for the decision of a model. For that, inputs are converted to human interpretable representations and fed to an interpretable linear model. To allow local approximation, random samples, that are close to original inputs and weighted accordingly are drawn and labeled using the original model, i.e. using BERT. Close in this setting means that a subsample of an article consists of mostly the same words as the original article — specifically that the more that are dropped, the further away the input, and the lower its weight. After applying the linear model on the sampled set, one receives the K most relevant words with their respective weights. I set $K = 10$ to get enough information from the mostly 500 token long texts and draw 5000 samples as suggested in the original paper. For multi-class predictions, like the five bias labels applied in this thesis, results for each class are predicted separately.

4.5 Deployed software and hardware

Both deep-learning models, BERT and BiLSTM, are applied using the deep-learning library PyTorch. While I construct the benchmark model from scratch, I use the transformer package by

Wolf et al. (2019) to construct the workflow with BERT and import its pre-trained parameters. For the RF benchmark I used the scikit-learn library. To accelerate model training and validation, I run all deep-learning experiments on Google Cloud Platform with a single NVIDIA T4 GPU[9]. For training the RF model, I employ an Intel Core i5-10210U CPU and run experiments parallel on eight workers.

Another way to speed up model training even further is the Pytorch extension AMP [10] (Automated Mixed Precision) by NVIDIA. The concept of mixed precision utilizes the differences between FP32 and FP16 tensors (Micikevicius et al., 2017). Using 16-bit floating point tensors offers faster calculations and lower memory usage, however, this comes at the cost of lower precision compared to FP32 tensors. Thus, with mixed precision, FP32 tensors are just used when a high degree of precision is necessary, such as for loss functions, weight updates, and norms; while matrix-matrix multiplications and most pointwise operations are done using FP16 tensors. This way, I approximately double the speed of training without losing any accuracy.

[9] https://www.nvidia.com/en-us/data-center/tesla-t4/
[10] https://nvidia.github.io/apex/index.html

5 Results

Results are divided into experiments that apply different datasets, algorithms, and other approaches. Insights drawn from applying LIME to three example articles are also discussed.

5.1 Experiments

Table 5.1 shows empirical results of training BERT on differently augmented datasets that are introduced in section 4.1. As it is also the case for benchmark models results further below, I present accuracy-, F1-, and MSE-scores for training, validation and testing sets. All values are averages over three runs, each conducted with a different random initialization with standard deviations given in parentheses.

Datasets	Training			Validation			Testing		
	Acc	F1	MSE	Acc	F1	MSE	Acc	F1	MSE
Unchanged	0.9122	0.9114	0.4440	0.8819	0.8809	0.6433	0.8817	0.8807	0.6140
	(0.0010)	(0.0011)	(0.0071)	(0.0027)	(0.0026)	(0.0179)	(0.0031)	(0.0034)	(0.0205)
Removed aggregators	**0.9379**	**0.9371**	0.3585	**0.9099**	**0.9099**	0.5774	**0.9080**	**0.9078**	0.5745
	(0.0014)	(0.0014)	(0.0131)	(0.0043)	(0.0041)	(0.0379)	(0.0024)	(0.0023)	(0.0172)
Removed tabloids	0.9212	0.9209	0.3569	0.8906	0.8900	0.5560	0.8884	0.8876	0.5709
	(0.0007)	(0.0007)	(0.0002)	(0.0029)	(0.0031)	(0.0249)	(0.0036)	(0.0038)	(0.0231)
Removed frequent sentences	0.9169	0.9160	0.4056	0.8780	0.8780	0.6356	0.8746	0.8746	0.6356
	(0.0008)	(0.0008)	(0.0051)	(0.0037)	(0.0032)	(0.0253)	(0.0047)	(0.0042)	(0.0239)
Combined changes	0.9384	0.9383	**0.3178**	0.9060	0.9061	**0.5202**	0.9026	0.9028	**0.5265**
	(0.0012)	(0.0012)	(0.0054)	(0.0048)	(0.0050)	(0.0328)	(0.0040)	(0.0041)	(0.0377)

Values are means over results of three runs with standard deviations in parentheses.

Table 5.1: Results of applied datasets

Removing news aggregators from the dataset leads to the best test results regarding the classification metrics accuracy and F1-score. This is not surprising, since, as mentioned above, removing news aggregators likely reduces possible confusion of the model caused through wrongly labeled articles. Removing tabloid newspapers, on the other hand, leads to a comparatively small improvement of accuracy and F1 score of approximately 0.7 percentage points, compared to more than 2.6 percentage points in the case of removed aggregators. The reason for this might be the above explained issue of a higher share of non-political content in tabloids. While this might counteract the purpose of detecting political bias, it does not necessarily stop the model from categorizing articles into the correct bias category. The model might still be able to use unwanted non-political aspects of an article to label it correctly. Since tabloids, at least in this dataset, are not evenly distributed over all bias labels, it could allow the model to connect tabloid contents to specific bias labels. This would imply that removing them might not lead to major improvements concerning classification metric scores.

As expected, only removing frequent sentences from articles leads to lower scores than not doing so. Accuracy and F1-score drop about 0.7 and 0.6 percentage points, respectively. This is most likely the case because repeating sentences, like the examples in Table 4.1, help the algorithm to easily connect articles to news outlets, and following that, to bias labels. Removing these kinds of sentences forces the algorithm to focus on other parts of articles and makes it more difficult to categorize them correctly. The last row of Table 5.1 shows results for the dataset to which all before mentioned changes are applied. With an accuracy and F1-score improvement of about 2 percentage points, it is the second best performing dataset in this list. A potential reason for the lower classification scores when compared to only removing aggregators could be that this dataset was trained on fewer articles than the best performing dataset. Assuming that removing tabloids and frequent sentences have similar effects on the combined dataset, as they have in the single cases, they might cancel each other out and what is left is the effect of a smaller training set. However, one still needs to take randomness into account, since the difference of about 0.5 percentage points is rather small given the standard deviations of both datasets' scores ranging from 0.23 to 0.41 percentage points.

Although presented here for completion, the MSE score gives only limited insights into the performance of models trained on the respective datasets. Changes in MSE scores here are highly influenced by changes in the distribution of bias labels within the augmented datasets. As shown in Figure 4.1 above, especially tabloids are only present in the two most extreme categories left and right, numerically 0 and 4, respectively. Assuming that, in case of wrong classification, the probability for each label is the same, then an article of one of these two categories has a higher expected error outcome than for example the center label with a numerical value of 2. Removing these articles then automatically improves the MSE, even when there is no actual improvement in performance. Thus, it is not surprising anymore that the combined changes lead to the lowest MSE, or that the dataset without tabloids has a lower MSE than the dataset without aggregators. This is because the datasets with a lower MSE mostly contain fewer extreme labels compared to the ones with a higher MSE.

In Table 5.2, I compare the metric scores of BERT with the ones of the two benchmark models. All three models are applied on the dataset that combines all suggested changes. It is immediately clear that BERT outperforms both benchmark models in all given metrics. The difference between BERT and the other deep-learning model SHA-BiLSTM is more than 6 percentage points regarding test accuracy and F1-score, and also the MSE scores have a difference of about 0.35. The gap between these deep-learning based models and the RF model is even wider. About 25 percentage points is the difference in accuracy and F1-score between BERT and RF, and about 1.6 between the MSE scores.

The discrepancy between the two neural network based models has two potential reasons. The first is the difference in model architecture. BERT utilizes multi-head-attention in every layer, while the SHA-BiLSTM only uses a single attention head in one layer. The big difference between performances of the two models could be an indication that the attention heavy

Models	Training			Validation			Testing		
	Acc	F1	MSE	Acc	F1	MSE	Acc	F1	MSE
BERT	0.9384	0.9383	0.3178	**0.9060**	**0.9061**	**0.5202**	**0.9026**	**0.9028**	**0.5265**
	(0.0012)	(0.0012)	(0.0054)	(0.0048)	(0.005)	(0.0328)	(0.0040)	(0.0041)	(0.0377)
SHA-BiLSTM	0.9124	0.9124	0.4929	0.8429	0.8423	0.8834	0.8416	0.8410	0.8752
	(0.0015)	(0.0014)	(0.0017)	(0.0045)	(0.0048)	(0.0837)	(0.0030)	(0.0032)	(0.0779)
RF	**0.9999**	**0.9999**	**0.0004**	0.6576	0.6559	2.0227	0.6541	0.6528	2.0956
	(0.0000)	(0.0000)	(0.000)	(0.0001)	(0.0001)	(0.0043)	(0.0007)	(0.0007)	(0.0029)

Values are means over results of three runs with standard deviations in parentheses.

Table 5.2: Results of benchmark models

transformer architecture outperforms models that do not rely as much on attention layers. Also architecture related is the size of both models. With 110M to 61M, BERT consists of almost double the amount of parameters that the benchmark model consists of. However, increasing the number of neurons per layer has only a small effect on the SHA-BiLSTM metric scores, which suggests that this is not the major reason for the difference in performance. The second difference, besides the architecture, involves the embeddings used within both models. While BERT is pre-trained as language model on a big text corpus, the embedding layer of the SHA-BiLSTM is randomly initialized, i.e., it was not pre-trained in any way.

Since the RF approach is different from both deep-learning approaches in many ways, it is difficult to pinpoint what exactly the reason is for the comparably bad result. One difference between the presented setting here and settings in media bias studies with similar approaches is the number of target labels. It might be that the deep-learning approaches handle the five labels better than the RF. Similarly, the linguistic count variables might not contain enough information to distinguish five classes, compared to mostly two classes used in the literature. Moreover, compared to the mostly automatically created deep-learning inputs, I created and selected the linguistic inputs manually. Although I do that in line with procedures and chosen variables in the literature, it could still be the case that my approach lacks specific variables or adjustments needed to accomplish better results. One such issue is that I do not have access to the proprietary LIWC[11] software, often used to create some of the dictionary based variables. Lastly, since big gaps between training and validation results are often a sign of overfitting, I want to mention that the use of different hyperparameter values that reduce training scores, such as minimum samples split, do not increase validation scores.

Similarly to the benchmark performance results discussed so far, the computational cost of presented models show big differences in computation time and memory usage. Table 5.3 shows the average computation time and the maximum memory usage of each model. The times displayed are the single run training and validation intervals, as well as the final prediction on the test-set. To allow for a better comparison, I present the SHA-BiLSTM values for a batch size of 16, as is the case for BERT, as well as for a size of 64, which is the actually applied size.

[11] https://liwc.wpengine.com/

Models	Time (min)	Memory (MiB)
BERT (batch=16)	430.13	12606
SHA-BiLSTM (batch=16)	337.58	4712
SHA-BiLSTM (batch=64)	187.23	10798
RF	9.32	5685

Table 5.3: Computational cost of models

Unsurprisingly, BERT, the biggest model, is the most expensive one. It takes more than double the time that it takes to train the SHA-BiLSTM. Even when the batch sizes are set to the same value, BERT still takes about 1.5h or 27% more time. Memory usage exhibits a similar discrepancy. Even with a four times bigger batch size, the benchmark model utilizes less memory than BERT. This allows for an easier adjustment of the SHA-BiLSTM to a given hardware depending on the available memory. Comparing the figures of the two deep-learning models to the ones of RF is not entirely valid, since former originate from a GPU, while letter originate from a CPU. One thus needs to consider the 1800 MiB that are allocated to the operating system in the RF case. Moreover, the employment of a GPU is more expensive, since it comes on top of the cost for CPU and dedicated memory. Nevertheless, it is evident that the RF application is computationally far cheaper than both deep-learning applications, taking only about 5% of the time the fastest deep-learning example takes. As shown in table 5.2 however, the savings in time leads clearly to a worse prediction performance.

Table 5.4 reports the results of predictions on the manually labeled SemEval 2019 dataset. Predictions of BERT trained on the original dataset without augmentations and with all applied changes are presented, as well as results of both benchmark models trained on the augmented dataset. Additionally, to put results into perspective, I also present scores of random predictions that I produce with class probabilities according to the true class distribution of the SemEval dataset, 36.9% hyperpartisan and 63.1% non-hyperpartisan. In this case, however, the presented F1 score is the standard binary version instead of the macro version for multiclass prediction that is used in other experiments.

Models + Datasets	Acc		F1	
Random	0.5333	(0.0165)	0.3795	(0.0178)
BERT Unchanged	0.6734	(0.0176)	0.5375	(0.0208)
BERT Combined changes	**0.6739**	(0.0155)	**0.5783**	(0.0280)
SHA-BiLSTM	0.5953	(0.0122)	0.4985	(0.0337)
RF	0.4036	(0.0029)	0.2574	(0.0067)

Values are means over results of three runs with standard deviations in parentheses.

Table 5.4: Results of SemEval dataset

Comparing the model outcomes to each other, the order from best to worst performing model resembles the results of Tables 5.1 and 5.2, with the BERT model trained on the dataset that includes combined changes performing best and RF performing worst. Furthermore, the absolute

differences between metric scores are also similar to Table 5.2, with exception of the two BERT versions, which further supports the findings regarding the performance differences between presented models. However, the level of performance is considerably lower than in the original test set case, especially considering that this is a binary class problem, rather than five classes as in the main dataset. Concerning the overall lower scores, it might be that the categorization of articles into left and right within the training set, and the categorization of hyperpartisan in the SemEval test set, do not coincide enough to obtain better results. Another reason could be that models do not generalize well enough to predict on articles from news sources they are not trained on. This aspect, I examine further below. Additionally, standard deviations are larger than in experiments before, most likely due to the small dataset, meaning findings are less reliable.

Results in Table 5.5 compare the cost sensitive to the standard cross entropy loss when applied within the training process of BERT. The cost sensitive loss performs worse on all calculated metric values. While I expect this result for accuracy and F1-score, I do not for the MSE. Apparently, the cost sensitive transformation of the loss function does not help the model to classify articles in an ordinal way that reduces distances between political bias labels. Although not affecting the gradient, augmenting the loss function this way might disturb the general loss minimization procedure more than it helps reducing the error term between true and predicted class. Given the range of standard deviations being 0.0228 to 0.0377 in the MSE case, the difference of 0.0224 between both losses is too small to make any credible judgment about a performance difference. Hence, the sensitive loss function applied here can be seen as ineffective to improve the MSE score in this setting.

Loss	Training			Validation			Testing		
	Acc	F1	MSE	Acc	F1	MSE	Acc	F1	MSE
Standard	**0.9384**	**0.9383**	**0.3178**	**0.9060**	**0.9061**	**0.5202**	**0.9026**	**0.9028**	**0.5265**
	(0.0012)	(0.0012)	(0.0054)	(0.0048)	(0.005)	(0.0328)	(0.004)	(0.0041)	(0.0377)
Cost sensitive	0.9341	0.9341	0.3485	0.9007	0.9008	0.5527	0.8994	0.8996	0.5489
	(0.0016)	(0.0016)	(0.0106)	(0.0038)	(0.0041)	(0.0131)	(0.0032)	(0.0035)	(0.0228)

Values are means over results of three runs with standard deviations in parentheses.

Table 5.5: Results of cost sensitive loss

Table 5.6 shows the effects that the removal of a given source has on the test prediction accuracy. Divided into a small and a large group, the first row presents the accuracy for each source when included in training, while the second row presents the accuracy for the same source when excluded from training. Generally, there is a significant drop in prediction performance for all news sources presented. However, there are differences in magnitude. Center is the worst performing class, regardless of whether one looks at the difference between first and second row, or the absolute accuracy. Articles of left sources, on the other hand, are predicted best in both cases. Their accuracy is clearly above the one of random predictions (0.2). This is also the case

for articles of both right sources, although scores are distinctly lower, and standard deviations larger than for the two left sources. Left and right leaning sources seem to be in between center and extreme classes, with each category having one source with lower and one with higher accuracy than random. As expected, accuracy scores of the remaining sources that are always included in training barely change. There is only a slight increase due to the fact that sources that on average do not improve the performance of predicting remaining sources are removed from the training set.

Group of sources	Sources used in training	Bias type of excluded source					
		Left	Lean Left	Center	Lean Right	Right	Remaining sources
Small group	yes	0.6629	0.4752	0.7547	0.5163	**0.8551**	0.9063
		(0.0386)	(0.0362)	(0.0556)	(0.0092)	(0.0489)	(0.0041)
	no	**0.4356**	0.0780	0.0063	0.3595	0.3768	0.9106
		(0.0671)	(0.0100)	(0.0089)	(0.1212)	(0.1824)	(0.0019)
Large group	yes	0.9286	0.8260	**0.9981**	0.7926	0.8189	0.9048
		(0.0150)	(0.1153)	(0.0014)	(0.0198)	(0.0202)	(0.0034)
	no	**0.5034**	0.4588	0.0010	0.1302	0.2801	0.9150
		(0.0382)	(0.043)	(0.0014)	(0.0518)	(0.1207)	(0.0014)

Values are means of accuracy scores from three runs with standard deviations in parentheses.

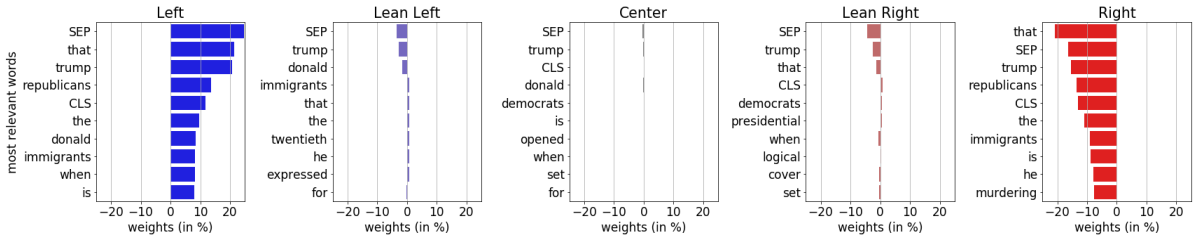
Table 5.6: Results of individual news sources removed from training

Although findings of this experiment cannot be entirely generalized, since only ten out of 58 sources were included, a few trends can nevertheless be observed. With the given dataset, the BERT model is not able to predict articles of the center class independent from their sources. This is most likely due to the more heterogeneous character of that group compared to its more biased counterparts. The special issue with Reuters, the center source of the large group, is described in the next section with LIME. Moreover, it seems that the model predicts the correct bias better the more partisan an article is, which is a pattern that cannot be seen from results of experiments that include all sources in training. Overall, Table 5.6 indicates that the model cannot predict political bias of articles sufficiently, unless it is trained on articles of the same source. As a side effect, it also shows that articles of sources that are less frequent in the training set tend to be correctly predicted less often. This follows from only one out of five small sources performing better than the large counter part, and only one small source having an accuracy above 80%, compared to four large sources with such an accuracy.

5.2 LIME insights

In the following, I present the insights offered by LIME for several news articles. These insights are always accompanied by two figures. The first figure shows the ten most relevant words with respective weights for each bias category. The higher positive weights are, the more important

LIME estimates a certain word to be for the respective class. Generally, the two functional tokens “CLS” and “SEP” that mark the beginning and ending of a text, get assigned high weights. This seems logical, since they are a crucial part of BERT’s inner workings, however, they do not offer any explanation into why the model predicts certain classes. Thus, I omit both tokens from the interpretation of the following plots. The second type of figure shows the actual article with the most relevant words highlighted. Words are highlighted in the color of the corresponding class. For articles that are shorter than 500 tokens, I remove the padding tokens at the end to allow for a more concise presentation.



Source: Politicus USA | True class: Left | Predicted class: Left

Figure 5.1: LIME - most relevant words - example 1

Figures 5.1 and 5.2 present results of an article that was correctly predicted to be left. The weights of left and right categories are almost symmetric, in the sense that most words that are important for a left classification have similar weights for the right class, only on the negative scale. At the same time, the other three classes seem irrelevant for the prediction of this article. This could indicate that the model detects in left and right articles a partisan language and then decides, depending on certain words and context, which class to predict. Regarding the presented words, it seems that aside from the frequent words without obvious standalone meaning such as “that” and “the”, the model makes its decision based on words that are representative for the content of the article, such as “trump”, “republicans”, and “immigrants”.

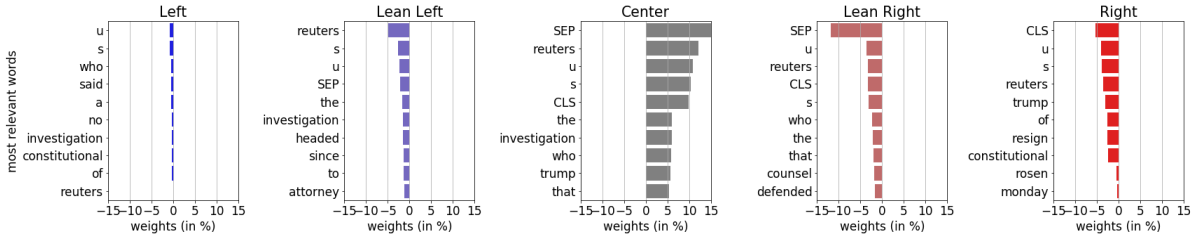
[CLS] donald trump opened his mouth and set the republicans up for a world of problems when he proclaimed that he would love to have another government shut ##down . trump said , if we don ##t change this , get rid of the loop ##holes , where killers and gang members can come into our country , if we don ##t change it , well have a shut ##down . well do a shut ##down , and its worth it for our country . i would love to see a shut down if we don ##t get this taken care of . trump is suggesting that legal immigrants are in gangs and murdering people . this isn ##t the first , second , or even twentieth that trump has expressed the view that immigrants are murderers . he launched his presidential campaign by calling rap ##ists and murderers . the es ##cala ##tion is logical from the point of view that trump is sinking under the weight of the russia investigation . the president is doing all that he can to hold to his base of supporters . donald trump is never going to sign any legislation to keep the dreamer ##s in the united states because he wants black and brown - skinned immigrants out of the country . trump has once again opened his mouth and screwed over the republican party . there is no pressure on democrats to vote for any cr that they don ##t like . trump wants a government shut ##down , so he will be blamed if the government shut ##s down . the only political cover that republicans had was blown to smith ##ere ##ens because trump can ##t hide his hate for non - white immigrants . [SEP]

Figure 5.2: LIME - article with most relevant words highlighted - example 1

Words that are split up during word piece tokenization tend to be regarded as less relevant,

most likely because splitting up a word into several tokens weakens the signal sent by that word, compared to one-token words. This might explain why the word “shutdown”, which is split up into “shut” and “##down”, is not among the ten most relevant words, despite its frequency and importance for the message of the article. Another aspect that should be considered when interpreting the results of LIME is that BERT is supposed to learn words not independently but in context of surrounding words. Hence, a specific word might indicate one class in a certain article and another class in a different article. Further, certain word tokens potentially signal a specific class when combined, but not when interpreted individually. A special case of this are the aforementioned split up words.

The next example is typical for an article of the news source Reuters. As seen in Table 5.6 of the section above, articles of this source are almost always correctly predicted to be center. Figure 5.3 shows that, ignoring the two functional tokens, “reuters” is the most important word to explain the center prediction. This is most likely the case because Reuters articles typically have the name stated at the beginning, as shown in Figure 5.4. Thus, the model learns a strong connection between the first word being “reuters” and the center category. Often the “reuters” token is accompanied by the location that the content of the text takes place, which might explain why the tokens “u” and “s” are considered similarly relevant, although having a different meaning in this text. Other words that might seem more relevant from a reader’s point of view to understand the content, like “investigation” and “trump”, are considered to be less relevant by the model, in comparison. Probably because of the strong signal of the first tokens of the article, other classes seem to be irrelevant for the bias prediction of the model.



Source: Reuters | True class: Center | Predicted class: Center

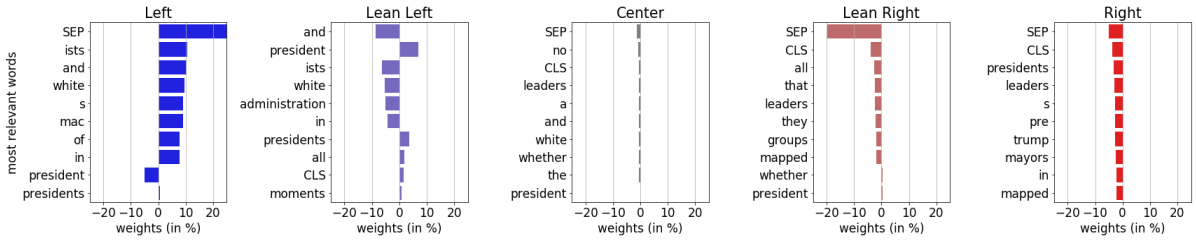
Figure 5.3: LIME - most relevant words - example 2

There is no easy fix to this problem. Simply removing all “reuters” tokens from the beginning of articles seems like a trivial solution to force the model to learn different connections between Reuters articles and the center class. However, this ignores that other news sources reference content of the news agency in their own articles and state Reuters as source, which could lead the model to learn connections between other bias categories and the “reuters” token instead. This is also one additional explanation why removing Reuters from the training set in Table 5.6 decreases the prediction performance by such an immense amount.

Figure 5.5 shows the most relevant words of an article that is categorized as lean left and predicted to be left. By simply looking at the relevant tokens of the left class, it is not obvious

[CLS] (reuters) - u . s . deputy attorney general rod rosen ##stein , who oversees the special counsel investigation into russia ##s role in the 2016 presidential election , headed to the white house on monday amid reports he would be leaving the post . a source told reuters that rosen ##stein had not been fired but had spent the weekend contemplating whether he should resign after a new york times report said in 2017 he had suggested secretly recording president donald trump . if rosen ##stein resign ##s , trump has more lee ##way on replacing him while firing him would make it harder for trump to designate a successor . the ax ##ios news website cited an unidentified source with knowledge of the matter as saying rosen ##stein , the no . 2 justice department official and a frequent target of trump ##s anger , had verbal ##ly resigned to white house chief of staff john kelly . another ax ##ios source said rosen ##stein is expecting to be fired so he plans to step down . nbc news reported rosen ##stein said he would not resign and the white house would have to fire him . trump has faced mounting pressure from the investigation by special counsel robert mueller , who is looking into russia ##s role in the 2016 presidential election . there had been widespread speculation that trump would fire rosen ##stein since friday when a new york times report said that in 2017 rosen ##stein had suggested secretly recording the president and recruiting cabinet members to in ##vo ##ke a constitutional amendment to remove him from office . the times said none of those proposals came to fruit ##ion . rosen ##stein denied the report as inaccurate and factual ##ly incorrect . the justice department made no comment on rosen ##stein . trump was at the united nations on monday cnn reported rosen ##stein was summoned for a meeting at the white house with kelly . shortly after the times story , trump told supporters at a rally in missouri that there is a lingering stench at the justice department and that were going to get rid of that , too . rosen ##stein has defended mueller and been a target of trump since he assumed supervision of the russia investigation after his boss , attorney general jeff sessions , rec ##used himself because of his own contacts with russia ##s ambassador to washington while serving as a trump campaign adviser became public . rosen ##stein ##s departure would prompt questions about whether trump , who has called the russia investigation a witch hunt , would seek to remove mueller . the move comes just six weeks ahead of the nov . 6 congressional elections , and could become an explosive political issue as trump ##s fellow republicans try to keep control of congress . [SEP]

Figure 5.4: LIME - article with most relevant words highlighted - example 2



Source: ABC News | True class: Lean Left | Predicted class: Left

Figure 5.5: LIME - most relevant words - example 3

how the model arrives at its conclusion. However, looking at Figure 5.6, one can see that the tokens “white”, “mac”, and “ists”; are three of the five tokens that make up the term “white supremacists”. The frequent use of this term, it is mentioned seven times in the article, leads the model to categorize the article as left. This probably follows from the fact that the term is nearly three times as frequent in left articles as it is in lean left articles of the training set. The singular version “white supremacist” is four times as frequent. Both can be inferred from Table A.3 in the Appendix. Following that the frequent use of the term “white supremacists” should lead to a left classification could be a decision made by a human annotator as well, since borders between political bias classes are rather vague. Taking this into consideration, the given article can be seen as an example of how BERT is able to make reasonable article level predictions that are missed by applying the same label to all articles from one source. Furthermore, it is also an example of the capability of BERT to detect the combined meaning of tokens. Lastly, the

only indication for a lean left prediction among the relevant words are the more formal phrases “president” and “presidents”.

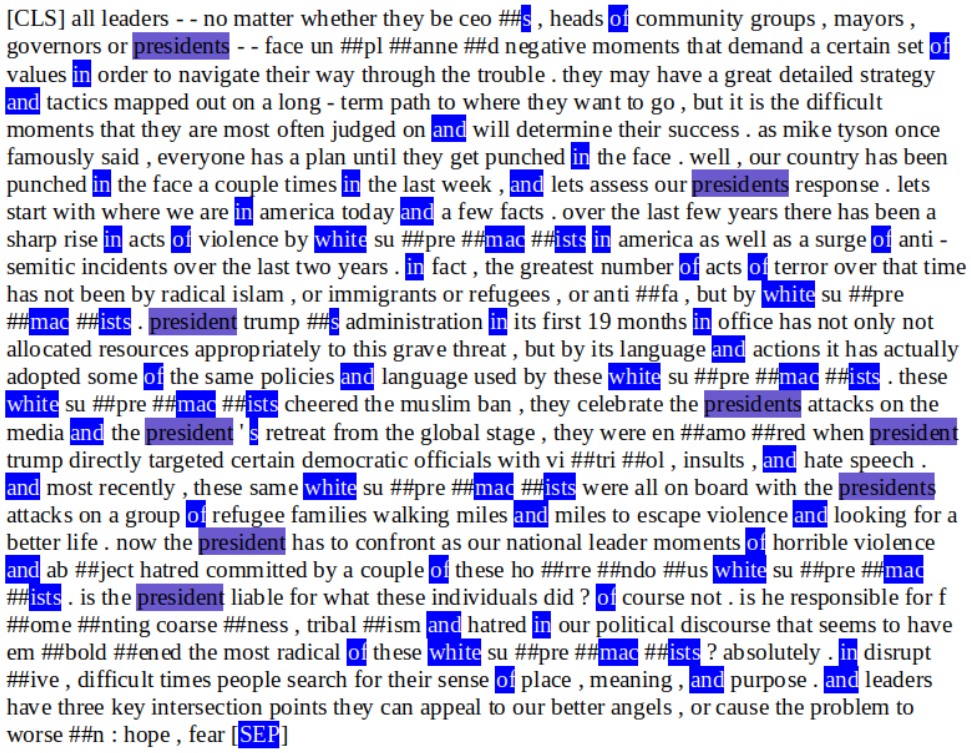


Figure 5.6: LIME - article with most relevant words highlighted - example 3

6 Discussion and Conclusion

In this thesis, I conduct several empirical experiments to test the capability of the deep-learning model BERT to predict the media bias of news articles and further compare its performance to two alternatives, a smaller and less computational expensive deep-learning model, and an approach that does not rely on neural networks. Moreover, I point out potential issues of datasets that contain articles that are labeled according to the news sources that publish them, and present their impact on the prediction performance. Additionally, I test the main model on a small manually labeled dataset, show that introducing a cost-sensitive loss does not improve the model's ability to integrate the proximity between similar bias classes (e.g. Left and Lean Left) into its predictions, and test whether the model is able to predict the bias of sources it is not trained on. Lastly, I apply LIME to offer a better understanding of example model predictions.

The empirical results show that news aggregators significantly worsen bias predictions, and thus, researchers using weakly labeled datasets, like in this study, should remove articles of these sources from the data. They could consider the same for tabloids, although the effect of this type of news outlet on predictions is not as clear. A method that successfully selects only political articles of tabloids is likely to be more effective. Even after applying these alterations, as well as removing frequent sentences that allow the model to make undesired connections during training, weakly labeled datasets still seem rather unsuited for the task of media bias prediction. The way these datasets are used in the literature and in this work, there still seems to be too much noise, such as the news outlet's name being mentioned in the article. Until there are better methods found to circumvent these issues, manually labeled datasets, despite their own disadvantages, might be the better option (Kiesel et al., 2019).

Following up on that, a model of the kind presented here has only a limited area of application. It could be used to predict media bias of articles of unknown sources such as on social media when only the text of a news article or parts of it are posted without indicating its source. The more sources are included in model training, the more likely the model predicts the correct bias then. Besides that, the benefits of such a model are limited, since it is not able to generalize enough to predict the bias of an article that has a source that was not included in the model training. Especially center articles, those considered to have little bias, are routinely very difficult to correctly predict by the model. Thus, at the current state, I cannot recommend that news consumers are exposed to such an application.

So far, most work in machine learning based media bias research neglected to offer insights into and explanations of the model's decisions. The application of LIME in this thesis shows how using explanatory methods can help better understand model predictions. Examples include that BERT seems to understand the opposing nature of left- and right-biased articles, and further LIME finds potential issues of predictions, including that the model connects news outlets mentioned in articles to bias classes. Thus, future research should consider applying more explanatory methods like LIME to better explain their model predictions and potentially

reveal undesired behavior. Moreover, this could help to increase the credibility of predictions presented to consumers, as is done in Horne, Nevo, et al. (2019) in the case of news reliability.

Regarding model choice, BERT is able to reach accuracy scores of more than 90%, and thus, should be considered for tasks similar to the above outlined limited area, and as a benchmark for future research in the field of media bias prediction. However, the empirical results also show that this kind of architecture comes at a significantly higher cost than an alternative deep-learning model. Hence, researchers need to asses whether these costs meet the benefits it offers to their specific application, and take findings of this study regarding computation time and memory usage into consideration. Furthermore, results suggest that the use of a random forest model in combination with linguistic variables cannot be recommended to predict a wide range of political biases in news articles.

Future work on the field of media bias prediction most likely cannot succeed by only using larger and more advanced models. Focus should be put on the improvement of data quality, and further on the development of methods that are able to cope with noise and other issues. Another issue that is not included in this thesis is that the change of topics, inclusion of certain important words, or even the writing patterns of news over time likely have an influence on predictions (Horne, Nørregaard, et al., 2019).

One thing that should not be understated is the importance of continued work in the field of machine learning driven media bias detection. The polarized political landscape of today is reflected in a similarly divided news media — the filter through which citizens receive needed information. This polarization is more and more present in the every day lives of people — only intensifying sociopolitical division that blocks needed structural reforms and action on urgent issues. Hence, research that helps explain media bias and offers solutions to present readers with more diverse views needs to be pursued.

References

Ahmed, H., Traore, I., & Saad, S. (2017): Detection of online fake news using n-gram analysis and machine learning techniques, In *First international conference on intelligent, secure, and dependable systems in distributed and cloud environments*, Vancouver, Canada. https://doi.org/10.1007/978-3-319-69155-8_9

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016): Layer normalization. *arXiv preprint arXiv:1607.06450*. https://arxiv.org/abs/1607.06450

Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., Van Der Goot, E., Halkia, M., Pouliquen, B., & Belyaeva, J. (2013): Sentiment analysis in the news. *arXiv preprint arXiv:1309.6202*. https://arxiv.org/abs/1309.6202

Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., & Nakov, P. (2018): Predicting factuality of reporting and bias of news media sources. *arXiv preprint arXiv:1810.01765*. https://arxiv.org/abs/1810.01765

Baly, R., Karadzhov, G., Saleh, A., Glass, J., & Nakov, P. (2019): Multi-task ordinal regression for jointly predicting the trustworthiness and the leading political ideology of news media. *arXiv preprint arXiv:1904.00542*. https://arxiv.org/abs/1904.00542

Barrón-Cedeno, A., Jaradat, I., Da San Martino, G., & Nakov, P. (2019): Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, *56*(5), 1849–1864. https://doi.org/10.1016/j.ipm.2019.03.005

Bernhardt, D., Krasa, S., & Polborn, M. (2008): Political polarization and the electoral effects of media bias. *Journal of Public Economics*, *92*(5-6), 1092–1104. https://doi.org/10.1016/j.jpubeco.2008.01.006

Budak, C., Goel, S., & Rao, J. M. (2016): Fair and Balanced? Quantifying Media Bias through Crowdsourced Content Analysis. *Public Opinion Quarterly*, *80*(S1), 250–271. https://doi.org/10.1093/poq/nfw007

Da San Martino, G., Yu, S., Barrón-Cedeño, A., Petrov, R., & Nakov, P. (2019): Fine-grained analysis of propaganda in news article, In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, Hong Kong, China, Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1565

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018): Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://arxiv.org/abs/1810.04805

Fan, L., White, M., Sharma, E., Su, R., Choubey, P. K., Huang, R., & Wang, L. (2019): In plain sight: Media bias through the lens of factual reporting. *arXiv preprint arXiv:1909.02670*. https://arxiv.org/abs/1909.02670

Flaxman, S., Goel, S., & Rao, J. M. (2016): Filter Bubbles, Echo Chambers, and Online News Consumption. *Public Opinion Quarterly*, *80*(S1), 298–320. https://doi.org/10.1093/poq/nfw006

Godbole, N., Srinivasaiah, M., & Skiena, S. (2007): Large-scale sentiment analysis for news and blogs. *Icwsm*, *7*(21), 219–222. http://www.uvm.edu/pdodds/files/papers/others/2007/godbole2007a.pdf

Graham, J., Haidt, J., & Nosek, B. A. (2009): Liberals and conservatives rely on different sets of moral foundations. *Journal of personality and social psychology*, *96*(5), 1029. https://doi.org/10.1037/a0015141

Grave, E., Joulin, A., & Usunier, N. (2016): Improving neural language models with a continuous cache. *arXiv preprint arXiv:1612.04426*. https://arxiv.org/abs/1612.04426

Graves, A., & Schmidhuber, J. (2005): Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, *18*(5-6), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

Hamborg, F., Donnay, K., & Gipp, B. (2019): Automated identification of media bias in news articles: An interdisciplinary literature review. *International Journal on Digital Libraries*, *20*(4), 391–415. https://doi.org/10.1007/s00799-018-0261-y

Hanawa, K., Sasaki, S., Ouchi, H., Suzuki, J., & Inui, K. (2019): The sally smedley hyperpartisan news detector at SemEval-2019 task 4, In *Proceedings of the 13th international workshop on semantic evaluation*, Minneapolis, Minnesota, USA, Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-2185

Hooper, J. B. (1975): On assertive predicates. In *Syntax and semantics volume 4* (pp. 91–124). Brill. https://doi.org/10.1163/9789004368828_005

Horne, B. D., Nevo, D., O'Donovan, J., Cho, J.-H., & Adali, S. (2019): Rating reliability and bias in news articles: Does AI assistance help everyone? *arXive preprint arXiv:1904.01531*. http://arxiv.org/abs/1904.01531

Horne, B. D., Nørregaard, J., & Adali, S. (2019): Robust fake news detection over time and attack. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *11*(1), 1–23. https://doi.org/10.1145/3363818

Hu, M., & Liu, B. (2004): Mining and summarizing customer reviews, In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining*, Seattle, WA, USA, Association for Computing Machinery. https://doi.org/10.1145/1014052.1014073

Hyland, K. (2005): *Metadiscourse: Exploring interaction in writing*. Continuum.

Iyyer, M., Enns, P., Boyd-Graber, J., & Resnik, P. (2014): Political ideology detection using recursive neural networks, In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, Baltimore, Maryland, Association for Computational Linguistics. https://doi.org/10.3115/v1/P14-1105

Jain, S., & Wallace, B. C. (2019): Attention is not explanation. *arXiv preprint arXiv:1902.10186*. https://arxiv.org/abs/1902.10186

Jiang, Y., Petrak, J., Song, X., Bontcheva, K., & Maynard, D. (2019): Team bertha von suttner at SemEval-2019 task 4: Hyperpartisan news detection using ELMo sentence representation convolutional network, In *Proceedings of the 13th international workshop on semantic evaluation*, Minneapolis, Minnesota, USA, Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-2146

Jurkowitz, M., Mitchell, A., Shearer, E., & Walker, M. (2020): *U.s. media polarization and the 2020 election: A nation divided* (tech. rep.). Pew Research Center. https://www.journalism.org/2020/01/24/u-s-media-polarization-and-the-2020-election-a-nation-divided/

Karttunen, L. (1971): Implicative verbs. *Language*, *47*(2), 340–358. http://www.jstor.org/stable/412084

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2017): Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(8), 3573–3587. https://doi.org/10.1109/TNNLS.2017.2732482

Kiesel, J., Mestre, M., Shukla, R., Vincent, E., Adineh, P., Corney, D., Stein, B., & Potthast, M. (2019): SemEval-2019 task 4: Hyperpartisan news detection, In *Proceedings of the 13th international workshop on semantic evaluation*, Minneapolis, Minnesota, USA, Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-2145

Kulkarni, V., Ye, J., Skiena, S., & Wang, W. Y. (2018): Multi-view models for political ideology detection of news articles. *arXiv preprint arXiv:1809.03485*. https://arxiv.org/abs/1809.03485

Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2014): News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, *69*, 14–23. https://doi.org/10.1016/j.knosys.2014.04.022

Merity, S. (2019): Single headed attention rnn: Stop thinking with your head. *arXiv preprint arXiv: 1911.11423*. https://arxiv.org/abs/1911.11423

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., Et al. (2017): Mixed precision training. *arXiv preprint arXiv: 1710.03740*. http://arxiv.org/abs/1710.03740

Munson, S. A., Lee, S. Y., & Resnick, P. (2013): Encouraging reading of diverse political viewpoints with a browser widget, In *Seventh international aaai conference on weblogs and social media*, Cambridge, Massachusetts, USA, The AAAI Press. https://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6119

Mutlu, O., Can, O. A., & Dayanik, E. (2019): Team howard Beale at SemEval-2019 task 4: Hyperpartisan news detection with BERT, In *Proceedings of the 13th international workshop on semantic evaluation*, Minneapolis, Minnesota, USA, Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-2175

Nørregaard, J., Horne, B. D., & Adalı, S. (2019): Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles, In *Proceedings of the thirteenth international aaai conference on web and social media*, Munich, Germany. https://doi.org/10.7910/DVN/ULHLCB

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017): Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*. https://arxiv.org/abs/1708.07104

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2017): A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638*. http://arxiv.org/abs/1702.05638

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019): Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, *9*(3), e1301. https://doi.org/10.1002/widm.1301

Recasens, M., Danescu-Niculescu-Mizil, C., & Jurafsky, D. (2013): Linguistic models for analyzing and detecting biased language, In *Proceedings of the 51st annual meeting of the association for com-*

putational linguistics (volume 1: Long papers), Sofia, Bulgaria, Association for Computational Linguistics. https://www.aclweb.org/anthology/P13-1162

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016): ‘‘why should i trust you?”: Explaining the predictions of any classifier, In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, San Francisco, California, USA, Association for Computing Machinery. https://doi.org/10.1145/2939672.2939778

Schuster, M., & Paliwal, K. K. (1997): Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681. https://doi.org/10.1109/78.650093

Serrano, S., & Smith, N. A. (2019): Is attention interpretable? *arXiv preprint arXiv:1906.03731*. https://arxiv.org/abs/1906.03731

Shapiro, A. H., Sudhof, M., & Wilson, D. (2020): Measuring news sentiment. Federal Reserve Bank of San Francisco Working Paper 2017-01. https://doi.org/10.24148/wp2017-01

Srivastava, V., Gupta, A., Prakash, D., Sahoo, S. K., R.R, R., & Kim, Y. H. (2019): Vernon-fenwick at SemEval-2019 task 4: Hyperpartisan news detection using lexical and semantic features, In *Proceedings of the 13th international workshop on semantic evaluation*, Minneapolis, Minnesota, USA, Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-2189

Strubell, E., Ganesh, A., & McCallum, A. (2019): Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*. http://arxiv.org/abs/1906.02243

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019): How to fine-tune bert for text classification?, In *China national conference on chinese computational linguistics*, Cham, Springer International Publishing. https://doi.org/10.1007/978-3-030-32381-3_16

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017): Attention is all you need, In *Advances in neural information processing systems*, Long Beach, California, USA, Curran Associates, Inc. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

Wilson, T., Wiebe, J., & Hoffmann, P. (2005): Recognizing contextual polarity in phrase-level sentiment analysis, In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, Vancouver, British Columbia, Canada, Association for Computational Linguistics. https://www.aclweb.org/anthology/H05-1044

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019): Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*. https://arxiv.org/abs/1910.03771

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Et al. (2016): Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. http://arxiv.org/abs/1609.08144

Appendix

Variable	Explanation	Source
Number of quotes	-	Horne, Nørregaard, et al. (2019)
Number of uppercase letters	-	Horne, Nørregaard, et al. (2019)
Total number of characters	-	own
Number of periods	-	Horne, Nørregaard, et al. (2019)
Number of question marks	-	Horne, Nørregaard, et al. (2019)
Number of exclamation marks	-	Horne, Nørregaard, et al. (2019)
Number of digits	-	own
Average sentence length	Number of word tokens divided by number of sentences	Horne, Nørregaard, et al. (2019)
Number of word tokens	-	Horne, Nørregaard, et al. (2019)
Average length of word tokens	Number of characters divided by number of word tokens	Horne, Nørregaard, et al. (2019)
Type-token ratio	Amount of different word token types divided by total amount of tokens	Barrón-Cedeno et al. (2019)
Hapax legomena	Number of word token types appearing only once	Barrón-Cedeno et al. (2019)
Hapax dislegomena	Number of word token types appearing only twice	Barrón-Cedeno et al. (2019)
Honore's R	$\frac{100 * log(number\ of\ tokens)}{1-(hapax\ legomena)/(number\ of\ types)}$	Barrón-Cedeno et al. (2019)
Yule's characteristic K	$10^4 \times \frac{\sum_i i^2 \times types_i - tokens}{tokens^2}$	Barrón-Cedeno et al. (2019)
Assertive verbs	List of verbs whose complement clauses assert a proposition	Hooper (1975), Recasens et al. (2013)
Bias lexicon	List of 654 bias-inducing lemmas	Recasens et al. (2013)
Factive verbs	List of verbs that presuppose the truth of their complement clause	Recasens et al. (2013)
Hedges	Used to reduce one's commitment to the truth of a proposition	Hyland (2005), Recasens et al. (2013)
Implicative Verbs	Imply the truth or untruth of their complement	Karttunen (1971), Recasens et al. (2013)
Negative opinion words	-	Hu and Liu (2004), Recasens et al. (2013)
Positive opinion words	-	Hu and Liu (2004), Recasens et al. (2013)
Report verbs	-	Recasens et al. (2013)
Authority dictionary	Taken from list with words connected to morality	Graham et al. (2009), Barrón-Cedeno et al. (2019)
Fairness dictionary	Taken from list with words connected to morality	Graham et al. (2009), Barrón-Cedeno et al. (2019)
Harm dictionary	Taken from list with words connected to morality	Graham et al. (2009), Barrón-Cedeno et al. (2019)
Ingroup dictionary	Taken from list with words connected to morality	Graham et al. (2009), Barrón-Cedeno et al. (2019)
Purity dictionary	Taken from list with words connected to morality	Graham et al. (2009), Barrón-Cedeno et al. (2019)
6 subjectivity dictionaries	List of subjective words divided into positive/neutral/negative and strong/weak respectively	Wilson et al. (2005), Barrón-Cedeno et al. (2019)
First person singular	First person singular pronouns: My, I, me, myself	Barrón-Cedeno et al. (2019), own
Second person	Second person pronouns: You, your, yourself	Barrón-Cedeno et al. (2019), own
18 Part of speech (POS)	Number of words of each POS category divided by total token count	Horne, Nørregaard, et al. (2019)

Table A.1: List of all linguistic variables

Group 1 (small)			Group 2 (large)		
Source	Frequency	Bias label	Source	Frequency	Bias label
Daily Kos	745	left	Politicus USA	3123	left
Washington Monthly	436	lean left	ABC News	2200	lean left
FiveThirtyEight	419	center	Reuters	2946	center
The Washington Examiner	368	lean right	Fox News	2444	lean right
FrontPage Magazine	722	right	CNS News	2603	right

Table A.2: Sources removed for generalization experiment

white supremacist(s) term	Left	Lean Left	Center	Lean Right	Right
singular	777	193	114	103	223
plural	499	180	55	81	139

Table A.3: Frequency of white supremacist(s) term per class in training set

Declaration of Academic Honesty

"I, Tobias Krebs, hereby declare that I have not previously submitted the present work for other examinations. I wrote this work independently. All sources, including sources from the Internet, that I have reproduced in either an unaltered or modified form (particularly sources for texts, graphs, tables and images), have been acknowledged by me as such. I understand that violations of these principles will result in proceedings regarding deception or attempted deception."

Tobias Krebs
Berlin, September 21, 2020