

Detecting Media Bias in News Articles using Gaussian Bias Distributions

Wei-Fan Chen

Paderborn University
Department of Computer Science
cwf@mail.upb.de

Khalid Al-Khatib

Bauhaus-Universität Weimar
Faculty of Media, Webis Group
khalid.alkhatib@uni-weimar.de

Benno Stein

Bauhaus-Universität Weimar
Faculty of Media, Webis Group
benno.stein@uni-weimar.de

Henning Wachsmuth

Paderborn University
Department of Computer Science
henningw@upb.de

Abstract

Media plays an important role in shaping public opinion. Biased media can influence people in undesirable directions and hence should be unmasked as such. We observe that feature-based and neural text classification approaches which rely only on the distribution of low-level lexical information fail to detect media bias. This weakness becomes most noticeable for articles on new events, where words appear in new contexts and hence their “bias predictiveness” is unclear. In this paper, we therefore study how second-order information about biased statements in an article helps to improve detection effectiveness. In particular, we utilize the probability distributions of the frequency, positions, and sequential order of lexical and informational sentence-level bias in a Gaussian Mixture Model. On an existing media bias dataset, we find that the frequency and positions of biased statements strongly impact article-level bias, whereas their exact sequential order is secondary. Using a standard model for sentence-level bias detection, we provide empirical evidence that article-level bias detectors that use second-order information clearly outperform those without.

1 Introduction

Media bias is discussed and analyzed in journalism research (Groseclose and Milyo, 2005; DellaVigna and Kaplan, 2007; Iyengar and Hahn, 2009) and NLP research (Gerrish and Blei, 2011; Iyyer et al., 2014; Chen et al., 2018). According to the study of Groseclose and Milyo (2005), bias “has nothing to do with the honesty or accuracy”, but it means “taste or preference”. In fact, journalists may (1) report facts only in favor of one particular political side and thus (2) conclude with their own opinion. As an example, the following sentences from *allsides.com* reporting on the event “Trump asks if disinfectant, sunlight can treat coronavirus”

demonstrate media bias on the sentence level:

The activists falsely claimed that Trump “urged Americans to inject themselves with disinfectant” and “told people to drink bleach.”

— The Daily Wire, right-oriented

Lysol maker issues warning against injections of disinfectant after Trump comments

— The Hill, center-oriented

“This notion of injecting or ingesting any type of cleansing product into the body is irresponsible and it’s dangerous,” said Gupta.

— NBC News, left-oriented

From an NLP perspective, bias in the example sentences could be detected by capturing sentiment words, such as “falsely” or “irresponsible”. Without the background knowledge of the political side of Trump or the event itself, however, predicting which side these sentences are slanted to is difficult.

Bias detection even becomes harder at the article level. For illustration, Figure 1 shows two articles and their sentence-level bias from the used dataset. It becomes clear that the actual words in the biased sentences are not always indicative to distinguish biased from neutral articles, nor is the count of the biased sentences: Bias assessments on sentence level do not “add up”. In this regard, the *position* of biased sentences appears to be a better feature.

The existing approaches to bias detection are transferred from other, less intricate text classification tasks. They largely model *low-level lexical information*, either explicitly, e.g. by using bag-of-words (Gerrish and Blei, 2011), or implicitly via neural networks (Gangula et al., 2019). Such approaches tend to fail at the article level, particularly for articles on events not covered in the training data. The reason is that bias clues are subtle and rare in articles, especially event-*independent* clues. Altogether, modeling low-level information at the

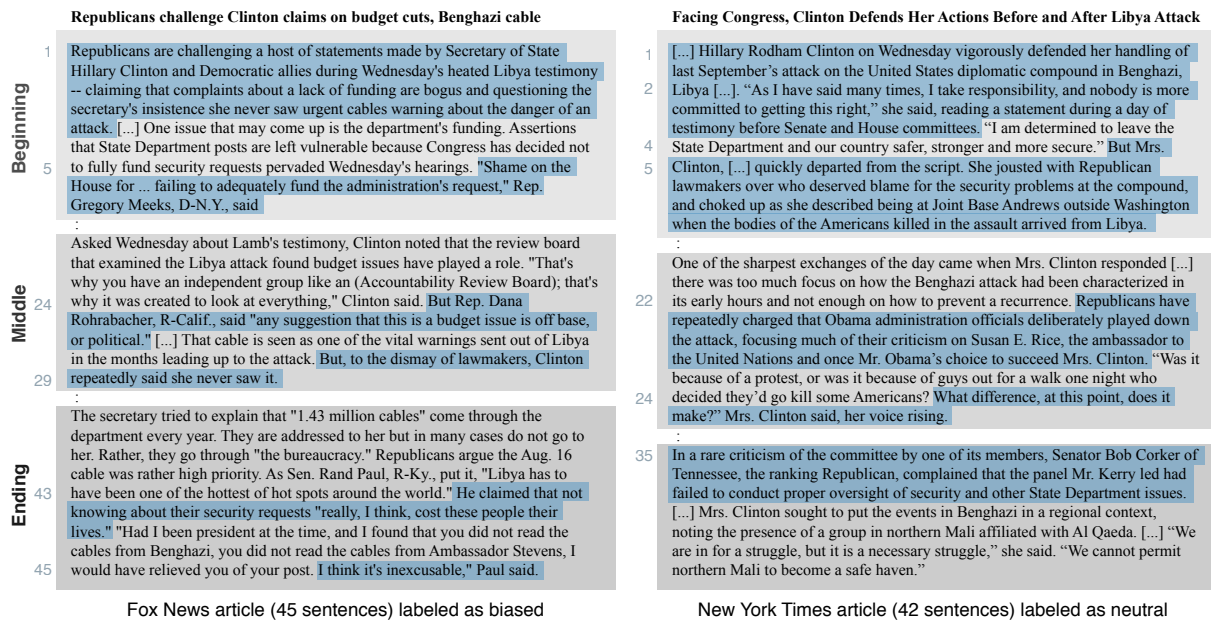


Figure 1: Excerpts of a biased article (left) and a neutral article (right) from the used dataset. All sentences labeled as having lexical or informational bias are highlighted; their position can be read from the numbers next to them.

article level is insufficient to detect article-level bias, as we will later stress in experiments.

We study article-level bias detection both with and without allowing to learn event-specific information. The latter scenario is more challenging, but it is closer to the real world, because we cannot expect that the information in future articles always relates to their past events. Inspired by ideas from modeling local and global polarities in sentiment analysis (Wachsmuth et al., 2015), we hypothesize that using *second-order bias information* in terms of lexical and informational bias at the sentence level is key to detecting article-level bias. To the best of our knowledge, no bias detection approach so far uses such information. We investigate this hypothesis in light of three research questions:

- Q1. How effective are standard classification approaches in article-level bias detection, with and without exploiting event information?
- Q2. How does sentence-level bias impact article-level bias in general?
- Q3. To what extent can sentence-level bias detection be utilized for article-level bias detection?

To study Q1–Q3, we employ the BASIL dataset, which includes manually annotated bias labels at article level as well as lexical and informational bias labels at sentence level (Fan et al., 2019). While the dataset contains only 300 articles, it provides the best basis for understanding the interaction of bias at both levels available so far.

For Q1, we evaluate an n -gram-based SVM and a BERT-based neural network in article-level bias detection. To assess the impact of event-related information, we split the dataset in two ways, once with event overlap in the training set and test set, and once without. As expected, we observe that the effectiveness of both approaches is generally low, especially when event information cannot be exploited. The results indicate that the concept of sentence-level bias is too subtle and rare to be utilized by these approaches.

For Q2, we study multiple types of correlations between sentence-level and article-level bias on the ground-truth annotations, covering (a) the frequency of biased sentences, (b) their position in an article, and (c) their sequential order. For each type, we model the bias distribution in a new way through a Gaussian Mixture Model (GMM), in order to then exploit it as features of an SVM (for frequency), Naïve Bayes (for positions), and a first-order Markov model (for sequential order). The results show strong correlations between the two levels for frequency and position information, whereas sequential order seems less correlated.

For Q3, finally, we propose a new approach applicable in realistic settings. In particular, we retrain the bias detectors from the Q1 experiments on the sentence level and then exploit the GMM as above to predict to article level bias. In our evaluation, the approach significantly outperforms the article-level approaches analyzed for Q1. Counter-

ing intuition, it even achieves higher effectiveness than what we observed on the ground truth for Q2. We explain this result by the fact that the sentence-level detector creates more deterministic sentence bias features, allowing our approach to learn from them in a more robust way.

Altogether, the contribution of this paper is three-fold: (1) We provide evidence that standard approaches fail in detecting article-level bias. (2) We develop a new approach utilizing second-order bias information, i.e., sentence-level bias. (3) We show that second-order bias information is an effective means to build better article-level bias classifiers.

2 Related Work

Media bias detection has been studied with computers since the work of [Lin et al. \(2006\)](#). As of then, media bias has been investigated in slight variations under different names, including *perspective* ([Lin et al., 2006](#)), *ideology* ([Iyyer et al., 2014](#)), *truthfulness* ([Rashkin et al., 2017](#)), and *hyperpartisanship* ([Kiesel et al., 2019](#)). To detect bias, early approaches relied on low-level lexical information. For example, [Greene and Resnik \(2009\)](#) used *kill* verbs and *domain-relevant* verbs to detect articles being pro Israeli or Palestinian perspectives. [Recasens et al. \(2013\)](#) relied on linguistic cues, such as factoid verbs and implicatives, in order to assess whether a Wikipedia sentence conveys a neutral point of view or not. Besides the NLP community, also researchers in journalism have approached the measurement of media bias. E.g., [Gentzkow and Shapiro \(2010\)](#) used the preferences of phrases at each side (such as “war on terror” for Republican but “war in Iraq” for Democratic). [Groseclose and Milyo \(2005\)](#) used the counts of think-tank citations to estimate the bias.

With the rise of deep learning, NLP researchers have also used neural-based approaches for bias detection. [Iyyer et al. \(2014\)](#) used RNNs to aggregate the polarity of each word to predict sentence-level bias based on parse trees. [Gangula et al. \(2019\)](#) made use of headline attention to classify article bias. [Li and Goldwasser \(2019\)](#) encoded social information in their Graph-CNN. While deep learning is believed to capture deeper relations among its inputs, we show that extending a neural network from sentence-level to article-level bias detection does not “just work”.

One point of variation in media bias detection is the level of text being analyzed, which varies from

tokens ([Fan et al., 2019](#)) and sentences ([Bhatia and Deepak, 2018](#)) to articles ([Kulkarni et al., 2018](#)), sources ([Baly et al., 2019](#)), and users ([Preoțiuc-Pietro et al., 2017](#)). While the effectiveness of machine learning models on different levels helps understanding how media bias becomes manifest at different levels, [Lin et al. \(2006\)](#) are to our knowledge the only to discuss the difference between sentence-level and article-level bias detection.

Source-level and user-level bias can be seen as directly emerging from summing up bias in the associated texts. For example, [Baly et al. \(2019\)](#) averaged the feature vectors of articles as the feature vectors of a source. The relation between sentence-level and article-level bias remains unstudied so far. The goal of this paper is not to discuss the difference between these levels. Rather, we examine how to aggregate the sentence-level bias to generate second-order features, and then use these features to predict article-level bias.

The use of low-level information to generate second-order features was studied in the context of product reviews by modeling patterns in the reviews’ sentiment flow ([Wachsmuth et al., 2015](#)), by tuning neural network to capture important sentences ([Xu et al., 2016](#)), and by routing in aggregating sentence embeddings into document embedding ([Gong et al., 2018](#)). In particular, our usage of low-level information is inspired by [Wachsmuth et al. \(2015\)](#), where we hypothesize that such flows exist in media bias as well. However, we do not limit our approach to entire sequences of sentence-level information, but we also consider frequency, position, or only two to three continuous sentences.

3 Standard Bias Detection Approaches

Standard approaches for bias detection, on both article and sentence level, mainly exploit the low-level lexical features to classify the texts as biased or not, neglecting bias-specific features. The two main low-level lexical feature types that are employed in such approach are: (1) n -gram features, where n is typically one to three (i.e., unigram, bigram, or trigram), and (2) word embeddings, especially within pre-trained language models (i.e., transformers) such as BERT.

We propose two classification settings to answer research question Q1, which addresses the importance of event information: In the first setting, called *event overlapping*, we form the training and test sets by randomly assigning examples to

them, more specifically, without looking at event information. The setting allows texts of the same event to occur in both the training and the test set. The second setting is called *event non-overlapping* since the texts to be classified are first categorized according to the main event that they address. During the splitting in training set and test set we then ensure for each event that all its related texts are in exactly one of these sets.

The difference in the effectiveness of the standard approaches on the two settings indicates whether and to what extent standard bias detection approaches rely on event information.

4 Second-Order Bias Information

For research question Q2, we study the correlation between sentence-level and article-level bias. Specifically, we examine whether article-level bias correlates with (a) the frequency of biased sentences, (b) their position in an article, and (c) their sequential order. For each correlation, we extract features and then train a respective machine learning model. The code is available at <https://github.com/webis-de/EMNLP-20>.

4.1 Bias Frequency

A straightforward way of leveraging sentence-level bias information is counting. Let an article with sentence-level bias labels $\{b_1, b_2, \dots, b_n\}$ be given, where n is the number of sentences in the article and b_i the label of the i -th sentence. Assuming that b_i is binary with $b_i = 1$ being bias, the *absolute bias frequency*, f_{abs} , is defined as:

$$f_{abs} = \sum_{i=1}^n b_i \quad (1)$$

Accordingly, the *relative bias frequency*, f_{rel} , is defined based on the length of the article as:

$$f_{rel} = \frac{\sum_{i=1}^n b_i}{n} \quad (2)$$

4.2 Bias Position

We consider the positions of biased sentences as second-order features. Given a target number of positions, k , we first normalize the sentence-level bias annotations $\{b_1, b_2, \dots, b_n\}$ into $\{\bar{b}_1, \bar{b}_2, \dots, \bar{b}_k\}$, with $\bar{b}_i \in [0, 1]$. The higher \bar{b}_i , the more likely position i is biased. In detail, we first normalize $\{b_1, b_2, \dots, b_n\}$ to $\{b'_1, b'_2, \dots, b'_m\}$ by linear interpolation, where m (here set to 100) is larger than the

largest n (and also larger than k). After the interpolation, b'_i is in the range of $[0, 1]$. Secondly, we “sample” from the b'_i to make the final sentence-level bias having length k . There are three “sampling” methods we explore: (1) average (take the average of the datapoints), (2) maximum (take the maximum value in the range), and (3) last (take the last datapoints). We treat this as a hyperparameter and find the best one by the validation set. We use this two-step normalization (upsampling and then downsampling) to avoid the instability during sampling when n/k is not an integer.

Our goal is to predict the most likely article-level bias label, a^* , given the sentence-level bias. Formally, assuming that an article can be seen as a combination of its sentences, we have

$$a^* = \arg \max_a p(a \mid \bar{b}_1, \bar{b}_2, \dots, \bar{b}_k), \quad (3)$$

where a is any possible bias label (0 for neutral and 1 for bias), and $p(a \mid \cdot)$ is the conditional probability of a , given a sentence-level bias sequence. According to Bayes’ rule and given that $p(\bar{b}_1, \bar{b}_2, \dots, \bar{b}_k)$ is irrelevant to the $\arg \max$, we can rewrite it as:

$$a^* = \arg \max_a p(\bar{b}_1, \bar{b}_2, \dots, \bar{b}_k \mid a) \cdot p(a) \quad (4)$$

Assuming that each \bar{b}_i is independent from other positions, we further simplify this as

$$a^* = \arg \max_a \prod_{i=1}^k p(\bar{b}_i \mid a) \cdot p(a), \quad (5)$$

which is a Naïve Bayes classifier, and each $p(\bar{b}_i \mid a)$ is the bias position feature we are interested in.

In the remainder, we simplify the notation $p(\bar{b}_i \mid a)$ to $p(\bar{b} \mid a)$. Estimating $p(\bar{b} \mid a)$ in each position for each a is difficult, since $\bar{b} \in [0, 1]$ and we cannot observe enough data points in that range on realistic text corpora. Instead, we therefore estimate $p(a \mid \bar{b})/p(a)$, where $p(a)$ can be properly estimated by the distribution of the labels, and $p(a \mid \bar{b})$ can be estimated well using a Gaussian Mixture Model.

Gaussian Mixture Model Given a set of m articles along with their bias labels, $\{a_1, a_2, \dots, a_m\}$, we first retrieve the interpolated bias value in each position $b_{i,j}$ where i is the index of the position and j is the index of the article. $b_{i,j}$, $1 \leq j \leq m$ can be seen as a distribution of the bias strength in one position i . For example, the distribution in

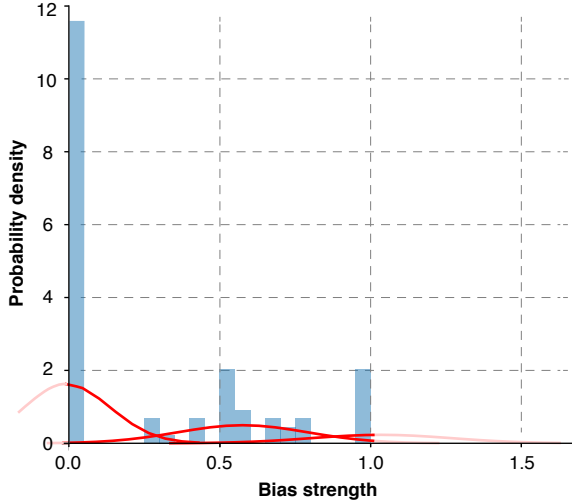


Figure 2: Bias strength in one position and the fitted Gaussian mixtures of it. The bias strength is the value of \bar{b}_i . Note that the y-axis is the probability density, i.e., the sum of all area in bins or sum of all area under Gaussian mixtures is one.

Figure 2 shows the bias in the second position if we normalize the articles into 10 positions.

To model the distribution, we employ a Gaussian mixture model (GMM) (Reynolds, 2009). The assumption behind GMMs is that a distribution can be seen as a combination of Gaussian distributions, where each distribution is represented by its mean μ , its variance σ^2 , and a weight w , the sum of all weights being 1. Modeling a GMM is unsupervised; we only need to set the number of mixtures we would like to have.

After applying GMM on $b_{i,j}, 1 \leq j \leq m$, the distribution of a bias position i is represented by a set of Gaussian mixtures, $\mathcal{N}_l(\mu_l, \sigma_l^2, w_l)$, where l is the index of mixtures. For each mixture, we can then learn its bias distribution by:

$$p(a = 1 | \mathcal{N}_l) = \frac{\text{occur}(\bar{b}_{i,j} \in \mathcal{N}_l, a_j = 1)}{\text{occur}(b_{i,j} \in \mathcal{N}_l)} \quad (6)$$

To avoid zero probability in some mixtures, we also apply add-one smoothing. Then, the bias probability $p(\bar{b} | a = 1)$ in one position is:

$$p(\bar{b} | a = 1) \propto \frac{p(a = 1 | \bar{b})}{p(a = 1)} \sim \frac{p(a = 1 | \mathcal{N}_{\bar{b}})}{p(a = 1)}, \quad (7)$$

where $\mathcal{N}_{\bar{b}}$ is the mixture most likely generating \bar{b} .

4.3 Bias Sequence

The Naïve Bayes classifier in Equation 5 assumes that each position is independent from other positions. We can also consider a position to depend

on the previous positions. For example, under the assumption that each position depends on the one before, we can rewrite Equation 5 as:

$$a = \arg \max_a \prod_{i=1}^k p(\bar{b}_i | \bar{b}_{i-1}, a) \cdot p(a) \quad (8)$$

Then, we can further rewrite $p(\bar{b}_i | \bar{b}_{i-1}, a)$ as:

$$p(\bar{b}_i | \bar{b}_{i-1}, a) = \frac{p(a | \bar{b}_i, \bar{b}_{i-1})}{p(\bar{b}_{i-1} | a) \cdot p(a)} \quad (9)$$

In this equation, $p(\bar{b}_{i-1} | a)$ can be approached by the GMM as described, and the numerator of the equation can be seen as the transition probability in a Markov process. In particular, after finding the mixtures most likely generating \bar{b}_i , and \bar{b}_{i-1} , we estimate the transition probability $p(a | \bar{b}_i, \bar{b}_{i-1})$ as:

$$p(a | \bar{b}_i, \bar{b}_{i-1}) \sim p(a | \mathcal{N}_i, \mathcal{N}_{i-1}), \quad (10)$$

where \mathcal{N}_i and \mathcal{N}_{i-1} are the mixtures most likely generating \bar{b}_i and \bar{b}_{i-1} respectively. Again, we apply add-one smoothing when estimating the transition probabilities.

The previous equations can be easily extended to the case that each position is dependent on more than one position. However, longer dependencies imply fewer observations of each possible transition. As a result, we only test the first and the second-order Markov process below (i.e., dependence on the previous one or two positions).

5 Experiments

This section presents the experiments that we designed to study research questions Q1–Q3 based on the media bias dataset BASIL.

5.1 Dataset

To test the hypothesis that sentence-level bias is an important feature for article-level bias detection, we need data that is annotated for both bias levels. Recently, Fan et al. (2019) released a dataset on media bias, *Bias Annotation Spans on the Informational Level (BASIL)*. The dataset contains 300 news articles on 100 events, three each per event. These three articles were taken from Fox News, New York Times, and Huffington Post, which have been selected as a representative of right-oriented, neutral, and left-oriented portals respectively.

On the article level, the dataset comes with manually annotated media bias labels (right, center, or

	Training		Validation		Test	
	Neutral	Bias	Neutral	Bias	Neutral	Bias
w/ Event	85	95	26	34	33	27
w/o Event	84	96	31	29	29	31

Table 1: Bias distribution of articles in the two experiment settings for research question Q1: *w/ event* indicates that there is event overlap in the training, validation, and test set (random split), while *w/o event* refers to an event-controlled split.

left). While we noticed that more Fox news articles are right (50) than Huffington post articles (10), the labels do not only rely on the source of the articles. Since we target bias in general rather than a specific orientation, we merged right and left to the label *bias*, and see center as *neutral*. Because both bias and unbiased articles include all three portals, we can be confident that the task is not detecting the source, but detecting the bias.

On the sentence level, each sentence has been manually labeled as having *lexical bias*, *informational bias*, or *none*. According to Fan et al. (2019), lexical bias refers to “how things are said”, i.e., the author used polarized or otherwise sentimental words showing bias. On the other hand, sentences with informational bias “convey information tangential or speculative”. In our experiments, we consider both settings where we separate the two types of bias and settings where we merge them.

5.2 Experiment Settings

In light of our three research questions, we consider the following experiments:

Q1. To study Q1, we compare two experiment settings of article-level bias detection: (1) with event information being available, and (2) with event information not being available. In both settings, the size of the training set (180 articles), validation set (60 articles) and test set (60 articles) are identical. The distribution of labels in each set and setting can be found in Table 1. As can be seen, the article-level labels are almost balanced, with some more biased than neutral articles. According to the distribution in the training set, we choose all-bias as the majority baseline in the later experiments.

As standard feature-based approaches, we employ an SVM and a logistic regression classifier based on word n -grams with $n \in \{1, 2, 3\}$. The considered n -grams are learned on the training set and lowercased. Hyperparameters such as cost and class balance are optimized on the validation set.

As a standard neural approach, we employ a pre-trained uncased BERT model using word embeddings as “features”.¹ We fine-tuned the approach and optimize the number of epochs for fine-tuning on the training and validation set. Only the first 256 and the last 256 words of an article are used for bias prediction, because the maximum sequence length of the BERT model is 512 tokens.

Q2. To study Q2, we use the same splitting of articles as used for the *w/o event* setting above. In the experiments of this research question, we use the ground-truth sentence-level bias from the dataset. Thereby, we investigate the ideal case where the sentence-level bias can be detected perfectly (assuming the manual annotations are correct). The different types of sentence-level bias are also tested to understand if article-level bias is more correlated to a certain type.

We prepare three types of sentence-level bias features, according to the descriptions in Section 4: For *bias frequency*, we consider a single feature SVM. We use linear kernel and optimize its cost hyperparameter on the validation set. For *bias positions*, we compute the bias probability in each position and then apply either Naïve Bayes, in line with Equation 5, or an SVM. For *bias sequences*, we use the Markov process from Equation 8 to predict an article-level bias label. Besides, we use the probabilities $p(\bar{b}_i | \bar{b}_{i-1}, a)$ as features for an SVM. Finally, we also test *stacking* models. To test the effectiveness of each feature, we stack all three SVMs of each bias feature, as well as any two of the three SVMs as an ablation test.

Q3. To study Q3, we test our approach in a real-world scenario. We first employ the same features and models as in Q1 for sentence-level bias classification. The only difference between article-level and sentence-level setting is that we do not trim sentences for the BERT model. The best classifier is later used in subsequent experiments. The splitting of sentences follows the *w/o event* splitting in the article-level bias detection, i.e., the sentences in the training set represent are used for training, and accordingly for validation and test. The distribution of the different types of sentence-level bias in each set can be found in Table 2.

Given the predicted sentence-level bias from Q1, we test our approaches as in Q2. Also, we test a scenario where the event information is available.

¹Cased and uncased BERT performed similarly in tests.

	Training		Validation		Test	
	Neutral	Bias	Neutral	Bias	Neutral	Bias
Lexical bias	4 611	263	1 558	85	1 382	78
Informational bias	4 102	772	1 404	239	1 272	188
Any bias	3 839	1035	1 319	324	1 194	266

Table 2: Distribution of the different types of sentence-level bias in the settings for research question Q1. In the *Any bias* setting, a sentence is considered biased if it contains lexical and/or informational bias.

Feature	Classifier	Accuracy
–	All-bias baseline	0.45
n -grams (1–3)	SVM	0.55 (+0.10)
n -grams (1–3)	Logistic Regression	0.46 (+0.01)
Word embeddings	BERT	0.52 (+0.07)

Table 3: Accuracy of the three standard approaches and the all-bias baseline in article-level bias detection on the dataset split *w/ event*. The numbers in parentheses indicate the difference compared to the baseline.

Similar to the setting in Q1, we randomly split the articles and then split the sentences according to their article-level splitting. We then train the sentence-level bias classifiers and use the best one for our approach.

6 Results and Discussion

To answer the three research questions of this paper, we report and discuss the results of the experiments described in Section 5.

6.1 Standard Approaches to Bias Detection

Tables 3 and 4 show the results of the experiments for Q1, which address the effectiveness of standard classification approaches in article-level bias detection. With a maximum of 0.55, the accuracy of all classifiers is generally low for a two-class classification task. When event information is available, accuracy improves at least up to 10 percentage points over the baseline, though. When not available, the classifiers seem to learn almost nothing: In the absence of event features, the classifiers are more forced to learn style or structural features. Yet, they turn out not to be able to do so without a proper design of such features. These results suggest that standard approaches are insufficient for article-level bias detection.

6.2 Impact of Sentence-Level Bias in General

As regards Q2, the column $Acc(GT)$ of Table 5 shows the accuracy of employing ground-truth

Feature	Classifier	Accuracy
–	All-bias baseline	0.52
n -grams (1–3)	SVM	0.52 (+0.00)
n -grams (1–3)	Logistic Regression	0.53 (+0.01)
Word embeddings	BERT	0.53 (+0.01)

Table 4: Accuracy of the three standard approaches and the all-bias baseline in article-level bias detection on the dataset split *w/o event*. The numbers in parentheses indicate the difference compared to the baseline.

Bias	Feature	Classifier	Acc (GT)	Acc (Pr)	
Lex.	f_{abs}	SVM	0.65	0.52	
	f_{rel}	SVM	0.63	0.48	
	Bias Position	Naïve Bayes	0.55	0.48	
		SVM	0.57	0.48	
	Bias Sequence	Markov Process	0.50	0.50	
		SVM	0.53	0.50	
	F + P	SVM Stacking	0.65	0.52	
		F + S	SVM Stacking	0.65	0.52
		P + S	SVM Stacking	0.52	0.52
		F + P + S	SVM Stacking	0.65	0.52
Info.	f_{abs}	SVM	0.57	0.52	
	f_{rel}	SVM	0.52	0.52	
	Bias Position	Naïve Bayes	0.55	0.50	
		SVM	0.55	0.50	
	Bias Sequence	Markov Process	0.48	0.48	
		SVM	0.47	0.48	
	F + P	SVM Stacking	0.55	0.52	
		F + S	SVM Stacking	0.58	0.52
		P + S	SVM Stacking	0.58	0.52
		F + P + S	SVM Stacking	0.58	0.57
Any	f_{abs}	SVM	0.65	*0.67	
	f_{rel}	SVM	0.65	0.65	
	Bias Position	Naïve Bayes	0.57	0.58	
		SVM	0.52	0.52	
	Bias Sequence	Markov Process	0.58	0.58	
		SVM	0.42	0.42	
	F + P	SVM Stacking	0.63	0.65	
		F + S	SVM Stacking	*0.67	0.62
		P + S	SVM Stacking	0.50	0.50
		F + P + S	SVM Stacking	*0.67	0.62

Table 5: Accuracy of all evaluated combinations of features and classifiers in article-level bias detection based on ground-truth (GT) and predicted (Pr) sentence-level bias. F combines absolute (f_{abs}) and relative (f_{rel}) bias frequency, P stands for bias position, and S for bias sequence. The best value for each bias type is marked bold. The best values overall are marked with *.

sentence-level bias features in predicting article-level bias. The SVM stacking classifier with bias frequency and sequence (F+S) performs best with an accuracy of 0.67. Stacking all features (F+P+S) achieves the same accuracy. In general, all feature and classifier combinations outperform all approaches found in Table 4.

Among the features for sentence-level bias, bias frequency and bias position can be exploited best by the SVM. While bias sequence does not perform as well as the others, the stacking classifier using it yields the highest effectiveness. The bias sequence appears to be weakest and sometimes brings negative impact to the performance. However, there may be several reasons behind it. For example, the sequential features may be too subtle, such that our models (SVM and Markov process) are too sensitive to the tiny changes in the features. But, it may also be that a smarter combination strategy for the three different types of feature is required; to keep the models simple, we tested only stacking. On the single features, the results show that an SVM is not always the best choice to utilize the features. In particular, Naïve Bayes and Markov process work better when dealing with informational bias and any bias.

Next, we take a closer look at the stacking part of Table 5, to analyze the feature’s effectiveness. While using lexically biased sentences as features, the frequency features contribute more (combinations in stacking with F achieve the best results). On the other hand, while using informationally biased sentences as features, the sequential features are more important. In other words, to detect article bias, it is important to know the number of lexically biased sentences as well as the order of informationally biased sentences. Our interpretation is that, the existence of lexical bias is already a strong clue for presenting bias, whereas informational bias has to be conveyed in a certain order or writing strategy (and thus is more difficult to be captured).

Regarding the two types of sentence-level bias, the best results are observed for *any* bias. Using only informational bias leads to the lowest effectiveness. While there is more informational than lexical bias, as shown in Table 2, the classifiers seem to rely more on lexical bias. The reason could be that lexical bias is easier to capture (by the word usage), while informational bias clues, if any, are subtle. Still, including both types of bias (but not distinguishing them) works best.

6.3 Impact of Predicted Sentence-Level Bias

Regarding Q3, we first present the results of applying the standard approaches to sentence-level bias detection in Table 6. Besides accuracy, we also show precision, since a high precision boosts the confidence in predicting sentence-level bias. We

Bias	Feature	Classifier	Acc.	Prec.
Lex.	–	All-bias baseline	0.05	0.05
	<i>n</i> -grams (1–3)	SVM	0.13	0.13
	<i>n</i> -grams (1–3)	Logistic Regression	0.07	0.05
	Word embeddings	BERT	0.95	0.38
Info.	–	All-bias baseline	0.13	0.13
	<i>n</i> -grams (1–3)	SVM	0.13	0.13
	<i>n</i> -grams (1–3)	Logistic Regression	0.47	0.14
	Word embeddings	BERT	0.86	0.37
Any	–	All-bias baseline	0.18	0.18
	<i>n</i> -grams (1–3)	SVM	0.38	0.18
	<i>n</i> -grams (1–3)	Logistic Regression	0.69	0.23
	Word embeddings	BERT	0.79	0.58

Table 6: Accuracy (Acc.) and precision (Prec.) of the three standard approaches and the all-bias baseline in sentence-level bias detection. The highest accuracy and precision values for each bias type are marked bold.

Bias	Classifier	Precision	Recall	F ₁
Lex.	Fan et al. (2019)	29.13	38.57	31.49
	Reimplementation	37.50	13.64	20.00
Info.	Fan et al. (2019)	43.87	42.19	43.27
	Reimplementation	58.62	32.08	41.46

Table 7: Classification results of Fan et al. (2019) and our reimplementation. Both use pre-trained BERT, but the exact dataset split of Fan et al. (2019) is unclear.

expect precision to be more important than recall, since we use the predicted bias for computing the article-level bias features. We find that fine-tuned BERT is strongest in effectiveness. Matching intuition, predicting lexical bias seems much easier than predicting informational bias.

Since Fan et al. (2019) provide their results of using BERT on sentence-level bias classification, we also used BERT for comparison. To this end, we split the dataset into sets of the same *size* as Fan et al. (randomly with 6819 training, 758 validation, and 400 test instances). However, the actual distribution of labels is not provided by the authors. As shown in Table 7, the results of our reimplementation for predicting informational bias is comparable to their results (in terms of F₁-score), but it is much worse for predicting lexical bias. Note that lexical bias in the dataset is rather rare (478/7984 ≈ 6%). We thus assume that the difference between our and the original test set caused the difference.

We used the predictions of the best sentence-level bias classifier (i.e., BERT) to compute the bias features. The resulting effectiveness in article-level bias detection can be found in column *Acc(Pr)* of Table 5. Comparing these results to those obtained for Q2, we see a clear drop in the effec-

tiveness, when using only lexical bias or only informational bias. Interestingly, however, the best configuration—with absolute bias frequency (f_{abs}) and SVM on any bias—is as good as the best one for Q2. This means that using the predicted bias can sometimes be better than using ground-truth bias. We explain this by the fact that sentence-level bias classifiers are deterministic while human annotators may be not, which can help our approaches to learn more stable patterns in the features.

Overall, our approaches with sentence-level bias information clearly outperform the standard approaches, underlining the impact of our approach. With an accuracy of 0.67, we outperform the standard approaches (0.53) by 14 points and the all-bias baseline (0.52) by 15 points. Regarding the different types of bias, the bias frequency is still the best feature, while the bias position and the bias sequence are weaker. The stacking model is the most effective in general.

Finally, we also considered the case where event information is available, as in Table 3. We followed the same process by selecting the best sentence-level bias classifier, which is again BERT with 0.83 accuracy and 0.58 precision, and use it to generate the article-level bias features. Similar to the results in Table 5, the best classifier is an SVM on absolute bias frequency. We achieve 0.60 accuracy outperforming the baseline (0.45), which is again around 15 points higher in accuracy. These results demonstrate that our approach can achieve high effectiveness robustly, regardless of whether it can exploit event information or not.

6.4 Hyperparameters

To deepen insights and to simplify reproducibility, this section discusses important hyperparameters used in the experiments.

Bias Normalization In the bias position and bias sequence features, the first step is to normalize the length of the bias annotations. Interestingly, the best sampling methods vary in different settings. Specifically, *last* is best for bias position with Naïve Bayes, *average* for bias position with SVM, *maximum* for bias sequence with Markov process; and *last* for bias sequence with Naïve Bayes.

Number of Normalized Positions We tested the number of positions needed in the bias position and bias sequence features. This number of positions roughly refers to how many bias clues are in an article. We find that the best value according to

the validation set is different in each setting. In summary we determine 10 for bias position with Naïve Bayes, 3 for bias position with SVM, 10 for bias position with Markov process, and 8 for bias position with SVM.

Number of Gaussian Mixtures The number of Gaussian mixtures indicates the variability of the bias distribution in a single position. We find that the best number of mixtures is 3 for bias position with SVM, and 5 for other settings. While this value depends also on the number of datapoints, it shows that setting it to 5 mixtures is reasonable in general.

Number of Markov’s Order We tested the order of the Markov process in Equation 8. We find that first-order Markov (a position depends on the previous position only) is best. As discussed, longer dependencies require more datapoints to estimate a better transition probability. Due to the size of our dataset (300 articles with 180 of them as the training set), the second or higher order of Markov does not make sense in our case.

7 Conclusion

In this paper we have given evidence that the exploitation of low-level lexical information is insufficient to detect article-level bias — especially, if the dataset is small. To provide a complete picture, we have formulated three research questions related to article-level bias detection, in order (1) to assess the state of the art of event-dependent and event-independent bias prediction, (2) to learn about the relation between sentence-level and article-level bias, and (3) to study whether sentence-level bias can be leveraged to predict article-level bias.

To tackle the detection of article-level bias, we have proposed and analyzed derived (second-order) bias features, including bias frequency, bias position, and bias sequence. As a main result of our research, we have shown that this new approach clearly outperforms the best approaches existing so far.

If bias detection can be done sufficiently robust on article level, we envisage, as a line of future research, the development of “reformulation” strategies and algorithms for the task of neutralizing biased articles (Pryzant et al., 2020).

Acknowledgments

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center “On-The-Fly Computing” (SFB 901/3) under the project number 160364472.

References

- Ramy Baly, Georgi Karadzhov, Abdelrhman Saleh, James Glass, and Preslav Nakov. 2019. Multi-task ordinal regression for jointly predicting the trustworthiness and the leading political ideology of news media. *arXiv preprint arXiv:1904.00542*.
- Sumit Bhatia and P Deepak. 2018. Topic-specific sentiment analysis can help identify political ideology. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 79–84.
- Wei-Fan Chen, Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. 2018. Learning to flip the bias of news headlines. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 79–88.
- Stefano DellaVigna and Ethan Kaplan. 2007. The fox news effect: Media bias and voting. *The Quarterly Journal of Economics*, 122(3):1187–1234.
- Lisa Fan, Marshall White, Eva Sharma, Ruisi Su, Prafulla Kumar Choubey, Ruihong Huang, and Lu Wang. 2019. In plain sight: Media bias through the lens of factual reporting. 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)*, pages 6344–6350.
- Rama Rohit Reddy Gangula, Suma Reddy Duggenpudi, and Radhika Mamidi. 2019. Detecting political bias in news articles using headline attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 77–84.
- Matthew Gentzkow and Jesse M Shapiro. 2010. What drives media slant? evidence from us daily newspapers. *Econometrica*, 78(1):35–71.
- Sean Gerrish and David M Blei. 2011. Predicting legislative roll calls from text. In *Proceedings of the 28th international conference on machine learning*, pages 489–496.
- Jingjing Gong, Xipeng Qiu, Shaojing Wang, and Xuan-Jing Huang. 2018. Information aggregation via dynamic routing for sequence encoding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2742–2752.
- Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics*, pages 503–511. Association for Computational Linguistics.
- Tim Groseclose and Jeffrey Milyo. 2005. A measure of media bias. *The Quarterly Journal of Economics*, 120(4):1191–1237.
- Shanto Iyengar and Kyu S Hahn. 2009. Red media, blue media: Evidence of ideological selectivity in media use. *Journal of communication*, 59(1):19–39.
- Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1113–1122.
- Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839.
- Vivek Kulkarni, Junting Ye, Steve Skiena, and William Yang Wang. 2018. Multi-view models for political ideology detection of news articles. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3518–3527.
- Chang Li and Dan Goldwasser. 2019. Encoding social information with graph convolutional networks for political perspective detection in news media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2594–2604.
- Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander G Hauptmann. 2006. Which side are you on? identifying perspectives at the document and sentence levels. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 109–116.
- Daniel PreoŃiu-Pietro, Ye Liu, Daniel Hopkins, and Lyle Ungar. 2017. Beyond binary labels: political ideology prediction of twitter users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 729–740.
- Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Thirty-Forth AAAI Conference on Artificial Intelligence*.
- Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference*

on Empirical Methods in Natural Language Processing, pages 2931–2937.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1650–1659.

Douglas A Reynolds. 2009. Gaussian mixture models. *Encyclopedia of Biometric Recognition*.

Henning Wachsmuth, Johannes Kiesel, and Benno Stein. 2015. [Sentiment flow - A general model of web review argumentation](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 601–611. Association for Computational Linguistics.

Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuan-Jing Huang. 2016. Cached long short-term memory neural networks for document-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1660–1669.