Leipzig University
Institute of Computer Science
Degree Programme Computer Science, B.Sc.

Entity Based Bias in News Articles

Bachelor's Thesis

Lukas Felix Göhlich

1. Referee: Prof. Dr. Martin Potthast

Submission date: January 31, 2022

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Leipzig, January 31, 2022

.....
Lukas Felix Göhlich

Acknowledgements

I would like to express my special thanks of gratitude to my supervisors Prof. Khalid Al Khatib and Shahbaz Syed, as well as my referee Prof. Dr. Martin Potthast. Computations for this work were done (in part) using resources of the Leipzig University Computing Centre.

Abstract

The internet allows people to collect a wide range of information on everyday events. Still, the critical judgment of such information lays upon people's responsibility. An automatic tool to classify a document's ideology (and pinpoint the hyperpartisan content) could protect readers from unconscious one-sided news consumption and aid authors to produce unbias news coverage.

In this thesis, we investigate how political bias manifests in a certain part of a news article, i.e., *the text given to introduce people*. To do so, we leveraged a linguistic pattern of interpolated text we call a 'descriptive statement', studying to what extent such statements encode political bias. Within an existing corpus of news articles labelled with political bias, we determine descriptive statements, engineer features that represent bias there (according to our view of bias), and use these features to train a collection of machine learning models to classify a given article into *left, center, or right*.

Inspecting the effectiveness of utilizing descriptive statements for bias identification, we compare our best-performing models to a 'lower bound' approach that guesses the article's bias. Besides, we compare the models against two 'upper bound' approaches that utilize the content of the *entire* given article. The results demonstrate that identifying bias in descriptive statements can be used to classify article bias (it outperforms the lower bound), suggesting that descriptive statements are used to a wide extent by authors to encode bias. Still, the entire article supposedly contains several linguistic parts (besides descriptive statement) that encode bias, and thus, can be used for more effective bias classification (upper bound).

Contents

Chapter 1

Introduction

1.1 Motivation

Opinion-free and objective news coverage is vital for autonomous judgment, decisions, and the resulting actions. Therefore, only unbias information can assist individuals to act in their best interests, resulting in true self-determination.

The web allows open access to information and thus expands the ability of individuals to obtain information independently and shape their views on current events [Hamborg et al., 2019]. The ubiquitous online and social media progressed into a big news source with a significant impact on people's beliefs and behaviours. Furthermore, news coverage has evolved away from complex and detailed reporting towards a personal and subjective style [Blake et al., 2019]. This can be ascribed to the efforts of news outlets to go viral on the web and social media, at least to some degree. Media channels discovered the potency of content that stimulates strong emotions, often induced by one-sided coverage [Kiesel et al., 2019].

These trends suggest an increasing exposure to polarized reporting, while leaving critical consumption to the individual's responsibility. This is illustrated in 2016 United States presidential elections [Kiesel et al., 2019]. Understanding of an article's agenda and ideology is indispensable to make an adequate judgment of the presented information. However, in our world where social media is omnipresent, constantly identifying an author's intent demands skill and persistence. Assisting automatic tools that help people decide what news articles to consume and what to discard evolved into a pressing issue.

An individual's beliefs are influenced by their experiences, education, and cultural background. Despite an author's intention of delivering an objective perspective on a subject, creating an opinion-free and unbiased report is not a trivial task, as the perception of bias is linked to ones' political views [Yano et al., 2010]. Understanding the mechanisms by which bias is introduced is

vital to detect and avoid it. Thus, it is the first step to debiasing of documents, manually and automatically.

1.2 Bias

In general, bias is described as a disproportionate weight that influences judgment. The term encompasses various forms and concepts across many disciplines [Wikipedia, 2021].

In this thesis, we investigate bias from the perspective of natural language processing. In a text, bias can manifest itself in the (1) lexical level i.e. word choice and positive and negative connotations, or the (2) semantic level, i.e. the choice of presented or neglected information [Fan et al., 2019]. Since the term “bias" is used in abroad fashion, we focus on certain types of text spans with extraneous information merely given by an author to influence the reader's opinion.

In this thesis, we view *bias* as a text:

1. affiliated with an ideology, at the lexical or semantic levels.
2. incompatible with the context of the document

1.3 Our Approach

One possible source of bias is the selected way to introduce entities. When mentioning named entities, authors tend to give background information alongside. Depending on the information itself, its relationship with the context, and its degree of generality, such information may convey a sentiment towards the entity. It is the author's decision what information to include and what words to use. Therefore, it is a powerful opportunity to frame the entity and introduce bias, consciously or unconsciously. We call such bias "entity-based bias".

In this thesis, we will explore to what extent bias is encoded in the information presented when introducing named entities. We will focus on news articles and consider bias affiliated with the left-right political spectrum. The named entities investigated will be restricted to *people*.

To gain a tangible notion of what information we want to consider, we will need a rigorous delimitation from the rest of the source document. Since we want to work with brief information that is closely tied to an entity, we opted to leverage a linguistic pattern known as an *appositive*; a type of *relative clause* (**Figure 1.1**) [Radford, 2019].

These **allegations**, *which Trump made during his campaign,* turned turned out to be fake.

Figure 1.1: Example of an appositive. The appositive (cursive) is surrounded by commas, it is preceded by its head (bold), and is prefixed by its relativizer (underlined).

Since we want to work with appositives that state information on named entities, namely people, we will constrain the appositives' heads and relativizers. We define a *descriptive statement* as an appositive where

1. the head is a proper noun (in our case a person)
2. the relativizer is *a/an/who/whose/the*

The election of **Trump**, *a candidate who made fear and xenophobia a central part of his campaign*, has spurred advocates to pledge to fight for the dignity of all families.

Figure 1.2: Example of a descriptive statement from a Truthdig article, a publisher labelled as *left* by BuzzFeed journalists or MediaBiasFactCheck.com

All eyes are on the markets as Wall Street could welcome **Trump**, *a businessman at heart*, with open arms.

Figure 1.3: Example of a descriptive statement from a Fox News article, a publisher labelled as *right* by BuzzFeed journalists or MediaBiasFactCheck.com

With these definitions established, we can proceed to describe our research question as follows: *To what extent does a descriptive statement encode bias?* To investigate this research question, we create a corpus that comprises descriptive statements and use it to train and test a machine learning model that predicts article-level bias on the left-right political spectrum. The effectiveness of the model determines the degree of bias encoded in a descriptive statement. The hypothesis that descriptive statements entail bias to an extent that allows them to be leveraged to classify an article's ideological bias is tested, and the results of our experiments confirm this hypothesis.

The thesis at hand is structured as follows; Chapter 2 gives a brief overview of related work in the field of bias classification. In chapter 3 we explain the approach to answer our research question in detail, including how we created and preprocessed a corpus of descriptive statements, and how we designed features to represent bias for individual descriptive statements. Chapter 4 considers the experiments we conducted using the corpus and the features. Ultimately, in chapter 5, we then discuss these experiments and their results, alongside examining the limitations and conversing about future work.

Chapter 2

Related Work

2.1 Token Level Approaches

Prior work in natural language processing on bias detection has primarily focused on leveraging bias arising from content realization; how information is worded. Since semantics are subordinate to such forms of bias, they are independent of outside context but dependent on linguistic attributes, such as polarized words.

Yano et al. [2010] compiled a corpus drawing 1100 sentences from political American blogs. Each sentence given to five annotators to be labelled with the extent of bias, the biases direction on the liberal-conservative spectrum and the words cuing the author's bias. The work suggests that certain tokens recurrently introduce bias into the sentence and that these tokens vary depending on the ideology. It is also shown that bias perception was influenced by annotators' political views, highlighting the complexity of compiling a well annotated corpus.

Recasens et al. [2013] also investigates bias introducing tokens and continues to apply them. The work employs Wikipedia's revision history and specifically edits aiming to eliminate bias. Comparisons between the text before and after the edits were used to gain insights on the biases lexical realisation. Two major types of biases were found. *Framing bias* stemming from word choice that conveys an attitude and *epistemological bias* related to the believability of a proposition. Single word edits were used to train a model, giving all unedited words from the sentence as negative examples and the edited word as a positive example. The trained model could predict the bias introducing word with a 34.35% accuracy and a 58.70% accuracy if returning three possible candidates.

Yano et al. [2010] and Recasens et al. [2013] works imply that single tokens can be indicative of bias at sentence level and associated the ideology. In this thesis, while not fully depending on them, we also employ bias cues on the

token level.

2.2 Richer Context Approaches

Approaches merely considering single tokens or *bag or words* classifiers fail to take into account compositional effects.

Iyyer et al. [2014] used logistic regression and deep learning models to classify bias as *liberal* or *conservative* on the sentence and phrase level. The trained RNN model outperformed logistic regression models that use *bag of words* and word embeddng features. The model achieved 70.2% in accuracy on the *Convote* corpus. Since we plan to examine a very specific concept, the descriptive statement for people, we will not have enough data to train an RNN. However Iyyer et al. [2014] shows that a logistic regression model with word embeddings still performs reasonably well, achieving 66.6% in accuracy. The *Convote* corpus was created from US Congressional floor debate transcripts. The sentences were labeled by propagating down the ideology of the speaker's party. This resembles the labelling of the corpus we plan to use, in which descriptive statements are labeled based on the articles' outlets' ideology they occurred in.

Fan et al. [2019] created *BASIL*, a corpus of 300 news articles covering 100 events. For each event *BASIL* includes the reporting of a liberal, center and conservative outlet. Bias spans in the articles were annotated with the type of bias, falling into either *lexical* or *informational bias*. Fan et al. [2019] calls bias concerning factual content employed merely to sway the reader's opinion *informational bias* and bias that manifests itself in word choice and linguistic attributes *lexical bias*. *BASIL* shows the prevalence of informational bias in new articles with 73.6% falling into that category. It is also suggested that informational bias occurs evenly throughout the article, while lexical bias occurs in the first quarter. A binary classification task was carried out on the token and sentence level, predicting whether bias occurs or not. A *BERT*-model achieved an F1-score of 18.71% for *informational bias* and 25.96% for lexical bias on the token level. On the sentence level the model achieved F1-scores of 43.27% for *informational bias* and 31.49% for *lexical bias*. This shows the importance of context when detecting *informational bias*, which is also relevant to our task, dealing with information given to introduce an entity.

Iyyer et al. [2014] and Fan et al. [2019] illustrate the importance of richer context when detecting bias, leading us to work with representations of the entire descriptive statement in this thesis.

2.3 Article Level Classification

Although previous work has been done in bias detection on token, phrase and sentence level, using these predictions in order to make assumptions on an article's bias is still mostly uninvestigated.

Chen et al. [2020] shows that using low level predictions to generate second order features intending to predict an article's bias is promising. The paper examines the performance of models trained on entire articles of the *BASIL* corpus and concludes that the performance of such models largely depends upon whether the model has been trained on other articles covering the same event, but perform with 55% accuracy at best. Leveraging second-order features improves the performance to 62%. In this thesis we use the descriptive statements as second order features.

[Kiesel et al., 2019] investigated the automated detection of hyperpartisanship, an extreme left- or right-wing affiliation, in news articles. In order to do so, two corpora of news articles were labeled and provided as resources for a shared task. One corpus featured 1273 manually labeled articles and the other contained 754,000 articles labeled by publisher. 42 teams submitted their approaches. Most teams employed convolutional neural networks (CNN). This way Bertha von Suttner managed to achieve a 82.2% accuracy on the manually label corpus. Tintin managed to achieve the best accuracy of 70.6% on the by-publisher corpus by only using n-grams as features. We utilize large parts of the by-publisher corpus for extracting descriptive statements.

Chapter 3

Approach

In this chapter we will explore how we created and preprocessed a corpus of descriptive statements. We will then go into what preliminary analysis we performed on the corpus and how we designed our features to represent bias.

3.1 Task

In order to answer our research question, we trained a classifier to infer a news article's bias using descriptive statements as input. The performance of the classifier indicates the level of bias encoded in descriptive statements. To be able to train such a classifier, we needed a corpus of descriptive statements and features representing bias based on our definition i.e. a representation for the statement's affiliation with an ideology and the statement's relevance to the context. In the following, we explain the requirements the corpus needs to meet, how we created it, and the intuition and technical realization of the features.

3.2 Data

3.2.1 Data Source

Since a descriptive statement is not a common concept, we could not fall back on an existing corpus. Thus, we were challenged to create the data set of descriptive statements ourselves. We chose to derive the descriptive statements from an existing corpus; *Data for PAN at SemEval 2019 Task 4: Hyperpartisan News Detection* (https://doi.org/10.5281/zenodo.1489920). The corpus had originally been created to perform detection of hyperpartisanship, however, the training data of the final version and all the data of the previous versions

contain article-level bias labels on the left-right political spectrum. Since the previous versions are not fully cleaned and have some encoding errors, we opted for the training data to be our source from which we extracted the descriptive statements. The source data contains 600,000 articles. 50% of the articles are labeled to have no bias, 25% are labeled left and 25% are labeled right. The articles are labeled by their publishers' overall bias provided by *BuzzFeed* journalists or *MediaBiasFactCheck.com*.

3.2.2 Descriptive Statement Extraction

Descriptive Statement Detection Due to our strict definition of descriptive statements, we can easily translate parts of the linguistic pattern detection into a string pattern matching problem. Detecting a proper noun i.e. named entity, especially over multiple tokens, is not a trivial task and can not be accomplished with simple pattern matching rules. Everything else, however, can be captured using a *regular expression* (regex).

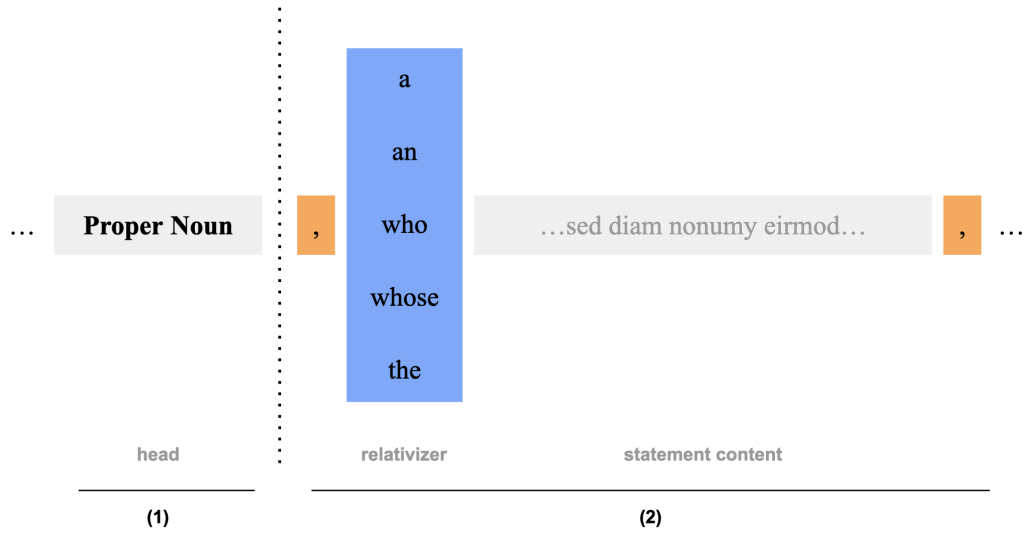

Figure 3.1: Components of a descriptive statement. **(2)** denotes the components of a descriptive statement detectable using a regular expression.

We tested the regular expression in **Figure 3.2** with a fixed head of "Trump" and retrieved 2543 descriptive statements. 26 of the 30 examined statements were of the kind we expected, resulting in a precision of 86.67%. The structure the regex detects is not exclusive to descriptive statements as it occasionally arises in other compositions. In our evaluation all false positives used the relativizer *the*. We evaluated the precision for 15 statements that

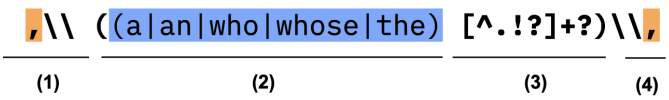

Figure 3.2: Descriptive Statement Regex - **(1)** and **(4)** capture the surrounding commas. **(2)** forces the leading token within the commas to be one of our defined relativizers. Since the chunk of text the pattern matching is carried out on, is not constrained to single sentences **(3)** ensures that any text, but no sentence separating character i.e. ".", "!", "?" occurs between the surrounding commas.

used *the* as their relativizer. The precision was only 73%. However, 19% of all the descriptive statements for "Trump" use *the* as their relativizer, so we decided not to exclude it.

Worse for **Trump**, *the CNU poll shows military voters*, a traditionally Republican constituency, back Clinton.

Figure 3.3: Example of a false positive descriptive statement captured by the regex

We also tried alternative regular expressions. Before constraining the relativizers, we tried a regex limiting the length of a clause, in the hope of detecting small pieces of information. The regex allowed for clauses of nine words or less. This regex had a 20% precision. Another assumption we tried, was that descriptive statements occur early in the sentence. We expanded our regex (**Figure 3.2**) so that it only detected descriptive statements at the start of the sentence. The precision of this regex was 93.33%. However, its recall performed poorly, as we were only able to retrieve 1129 for a head of "Trump" compared to the 2058 statements our final regex retrieved.

The regular expression only constitutes the first stage of the detection mechanism. After a match is found, named entity recognition (NER) is employed to detect the bounding box of the descriptive statement's head, if present. (Denoted as **(1)** in **Figure 3.1**). The statement's head and content then get concatenated into a descriptive statement (**Figure 3.4**). We used *SPACEY*'s *en_core_web_trf* pipe to perform the NER. The order of executing matters, since NER is an expensive operation, only performing it on possible candidates, qualified by the regular expression, increases the runtime performance of the detection significantly.

The English language capitalizes proper nouns. One might be inclined to exploit this and substitute the NER for another pattern matching solution, detecting all leading capitalized tokens. However examples like "Vincent van Gogh" show what kind of pitfalls can arise.

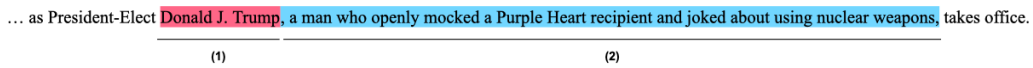

Figure 3.4: Descriptive statement component detection - **(1)** denotes the head detected using NER. **(2)** denotes the components detected using the Regular Expression.

Target Entity The descriptive statement's head specifies its target entity. However entities are not always referred to using the same exact name. "Trump" for example is mentioned as "Trump", "Donald Trump", "Donald J. Trump", "Donald J Trump" or "the Orange Man". These references need to be grouped so the descriptive statements can be associated with the same entity. Additionally the surname alone is often ambiguous and the entity in question can only be decided based on the statement's context. To address these problems we employed *GENRE* (Cao et al. [2021]), Facebook's generative entity retrieval system for entity disambiguation. *GENRE* provides us with a confidence-score, we later use. The score is negative and the closer it is to zero, the more confident the model is. In order to perform entity disambiguation, *GENRE* needs the entities bounding box within its surrounding text. For this we reused our previously separately detected statement head.

Source Article Context Since our definition of bias is dependent on the descriptive statements context, we retrieved the entire article as well. However, the original corpus only split the articles into paragraphs. Since we want to have more granular control over the context we consider, we used *SPACY*'s sentenizicer to split each article into a two-dimensional list representing paragraphs and sentences.

Bias Labels The bias labels in the original data are "left", "center-left", "least", "center-right" and "right". We assume that center-left and center-right might not be extreme enough, for us to be able to pick up on subtle cues with the amount of data we have. Therefore we merged "left-center", "least" and "right-center" into the label "center".

Extracted Raw Data Statistics Using this extraction pipeline we were able to retrieve 198,659 descriptive statements on 97,868 target entities. Considering we had 600.000 source articles, on average around every third article uses a descriptive statement on a person.

The mean number of statements per entity is 2.03, with 2645 statements being the maximum (Donald Trump) and 1 statement being the minimum.

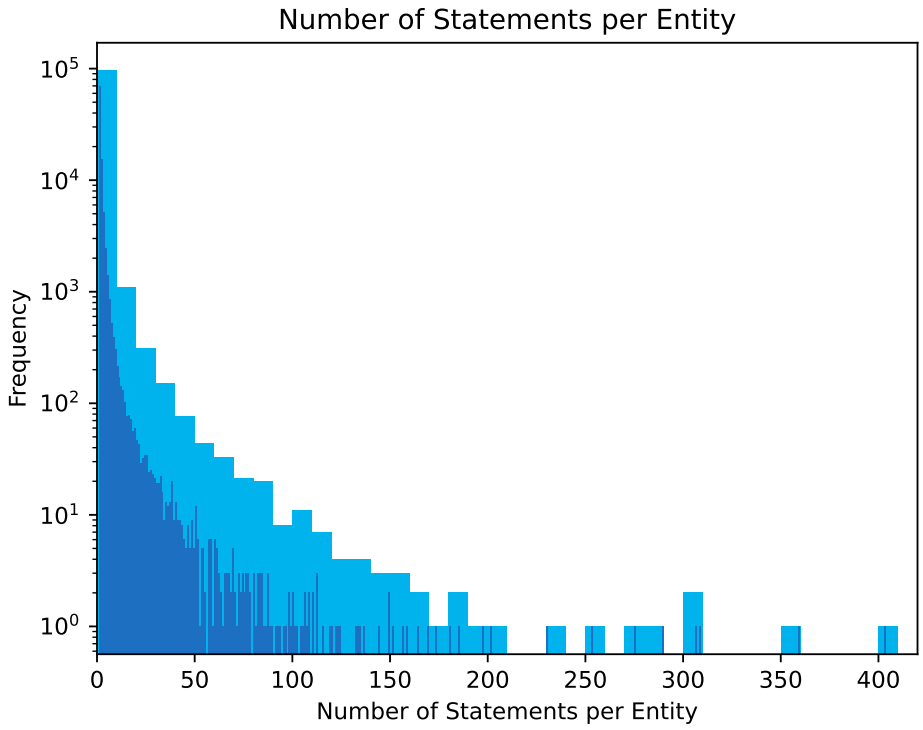

Figure 3.5: Raw Descriptive Statement Data Distribution - The x-axis represents the number of descriptive statements associated with a single entity and the y-axis represents the number of entities with such a associated statement count. The lighter bars display buckets of size 10.

The standard deviation of 10.93 and **Figure 3.5** imply that most entities only have one or very few descriptive statements, while only some exceed 50 statements. This is to be expected, considering that only some entities have a lot of news coverage while others only appear occasionally.

41.54% of the descriptive statements are "center", 37.33% are "left" and 21.13% are "right". This is curious we would expect a similar amount of labels on the left as on the right.

For multiple entities we observed recurring identical descriptive statements for the same targeting the entity. For example the statement "Trumps, who campaigned on warmer ties with Putin" occurs 96 times. We conclude that this is due to news outlets working with agencies that provide them with information. This information might not always get paraphrased.

Attribute	Description
descriptive statement	the statement itself
target entity	the disambiguated entity the descriptive statement gives information on
target entity confidence score	the confidence score of the *GENRE* model on the predicted target entity
context	the article the descriptive statement occurred in, split into paragraphs and sentences
bias	the bias of the article the descriptive statement occurred in. *(left / center / right)*

Table 3.1: Extracted Descriptive Statement Schema

3.2.3 Preprocessing

As stated before and as suggested by **Figure 3.5** many entities only have one or few associated statements. Our definition of bias is based on the affiliation of a descriptive statement with an ideology. In order to develop features representing ideology affiliation, we need a reasonably sized sample of statements, such that similar statements might be prevalent within an ideology. If however, the sample size is too small, we have no representative indication of ideology.

	Entity	statement count	center	left	right
1	Donald Trump	2645	45.29%	23.52%	31.19%
2	Barack Obama	966	22.26%	48.65%	29.09%
3	Suzanne M. Levine	884	0.0%	0.11%	99.89%
4	Hillary Clinton	506	19.57%	42.69%	37.75%
5	John McCain	403	31.27%	48.39%	20.35%
6	George W. Bush	359	13.93%	77.16%	8.91%
7	Mitt Romney	308	15.91%	64.61%	19.48%
8	Bernie Sanders	306	21.57%	46.73%	31.7%
9	Steve Bannon	289	57.09%	20.42%	22.49%
10	Bill Clinton	275	15.27%	56.36%	28.36%
11	Vladimir Putin	253	56.52%	26.88%	16.6%
12	David MacKay (VC)	230	0.0%	100.0%	0.0%
13	Mike Pence	201	46.27%	30.85%	22.89%
14	Aleksandr Kogan	197	100.0%	0.0%	0.0%
15	James Comey	185	76	47	62

Table 3.2: First 15 Entities sorted by Statement Count

Considering **Table 3.2**, two more problems become apparent. Some en-

tities' statements primarily occur in articles with the same ideology. For the same argument as before, this skewed distribution prevents us from creating representative features indicating ideology. In **Table 3.2**, this is the case for "Suzanne M. Levine" *(3)*, "David MacKay" *(12)* and "Aleksandr Kogan" *(14)*. There also seems to be a correlation between heavy repetition of descriptive statements and a skewed label distribution. For "Suzanne M. Levine" all statements are identical, 77.25% for "David MacKay" and 49.75% for "Aleksandr Kogan".

The other problem is that the entity disambiguation is not always correct. "Suzanne M. Levine", for example, is actually "Suzanne Frey", an executive at Alphabet.

We conducted further processing of the data to counteract the described problems. Note that, since the preprocessing steps are dependent on each other, they have to be carried out in the presented order.

Target Entity Preprocessing Some entities are not disambiguated correctly. We used the *GENRE*-model's confidence score as an indicator to which references might have been mapped to the wrong entity. The confidence score is negative. A value closer to zero denotes a higher confidence of the model.

Figure 3.6 suggests that a significant amount of statements has a confidence score between -0.5 and 0. To decide if -0.5 is a reasonable minimum-threshold, we compare the precision of the disambiguation between -0.5 and 0 to the precision between -1.0 and -0.5.

Whether a disambiguation is correct has to be decided manually. Since some prior knowledge about the entity is needed to make an educated judgement on the disambiguation, it is not feasible to pick random statements from the corpus. Instead we used the 10 entities with the most statements. For each score interval and for each entity we sorted the statements chronologically. If applicable, we chose 5 statements evenly distributed throughout the sorted lists of entities for each score interval. The chronological sorting is to avoid the disambiguation performing better on newer or older events. We assessed each disambiguation using the source article as context.

The precision between -0.5 and 0 was 97.78%. The precision between -1 and -0.5 was 32.14%. Since this is a significant drop in precision, we choose -0.5 to be our threshold and discard all descriptive statements with a lower score.

After setting this threshold we are left with 75,999 statements (38.26%) and 27,411 entities (28%). Although a lot of statements are lost, 77.57% of the removed statements belonged to entities with less than 5 associated statements and 94.33% of the statements removed belonged to entities with less than 30 associated statements. As stated before, we are not able to profit of entities

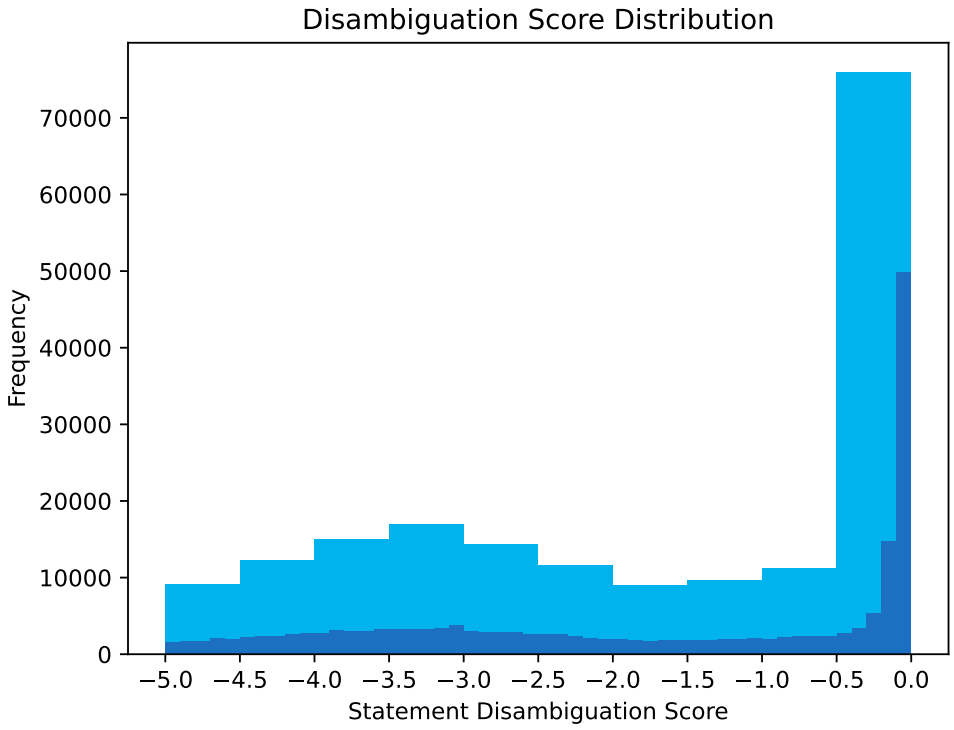

Figure 3.6: *GENRE*-Model Confidence Score Distribution - The x-axis represents the the confidence scores. The y-axis represents the frequency of scores within the interval. The darker bars represent an interval of 0.1. The lighter bars represent an interval of 0.5.

with few statements anyway.

Statement Bias Distribution To create a representative statement distribution feature, an entity's statements should be evenly distributed across the bias labels to some extent. In order to ensure at least a low degree of uniform distribution across the bias labels, we employed *Shannon Entropy*.

Given a discrete random variable X, with possible outcomes x_1,..., x_n, which occur with probability $P(x_1), ..., P(x_n)$, the entropy of X is formally defined as:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i)$$

(Wikipedia [2021])

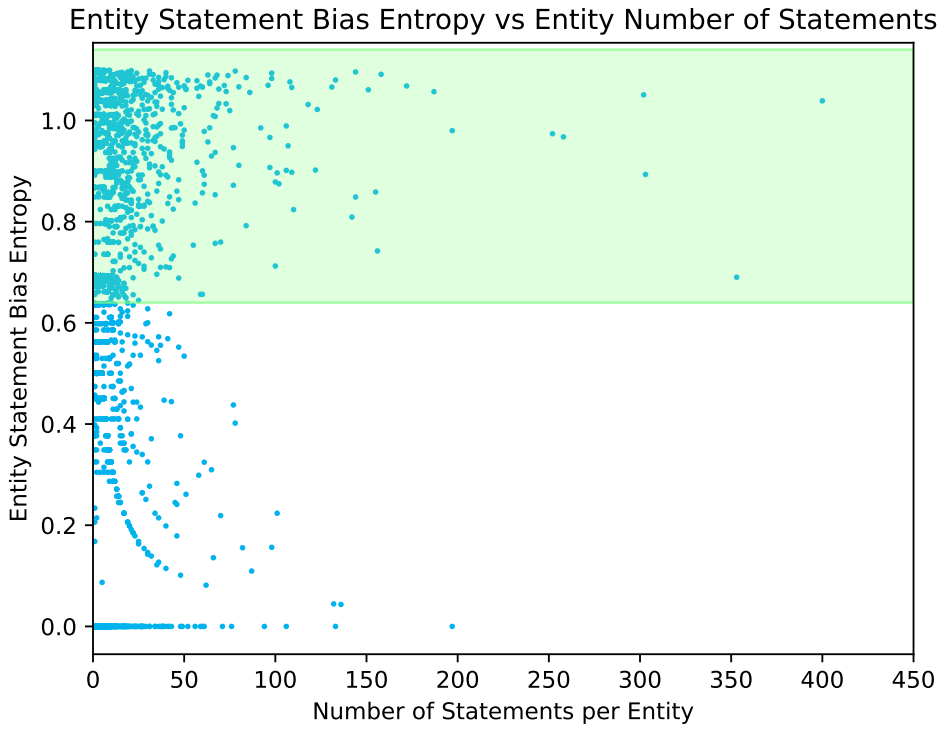

Figure 3.7: Entity Descriptive Statement Bias Entropy - Each marker represents an entity. The x-axis represents the entity's the number of statements. The y-axis represents the entity's statements' bias distribution entropy. Within the shaded area are the entities that qualify above the threshold.

We can treat the distribution of bias labels for an entity's statements as a random variable X, where the possible outcomes are x_{center}, x_{left}, x_{right}. Then, in case of x_{center}:

x_{center} = the number of statements with bias 'center'

$P(x_{center}) = \frac{x_{center}}{x_{center}+x_{left}+x_{right}}$

A uniform probability, i.e. in our case an equal amount of statements labeled center, left, and right, yields maximum entropy and can then only decrease (Wikipedia [2021]). Thus, we can introduce an entropy threshold, below which we disregard the entity. Depending on this threshold, we can control the allowed deviation from a uniform distribution.

The goal of the entropy threshold is not to force a close to uniform distribution, but rather to eliminate entities with very skewed bias label distributions.

The threshold should be an adequate tradeoff between the label distributions we use and the amount of data we can use.

Based on **Figure 3.7** we chose an entropy of 0.639 as our threshold. Applying it excludes low entropy entities while preserving entities with reasonable entropies and a large number of statements. The chosen entropy threshold ensures that no bias label is present in under 10% of the statements. After utilizing the threshold, we are left with 32764 statements (43.11%) for 3161 entities (11.53%). 70.67% of the statements came from entities with under 5 associated statements and 90.46% came from entities with under 30 associated statements.

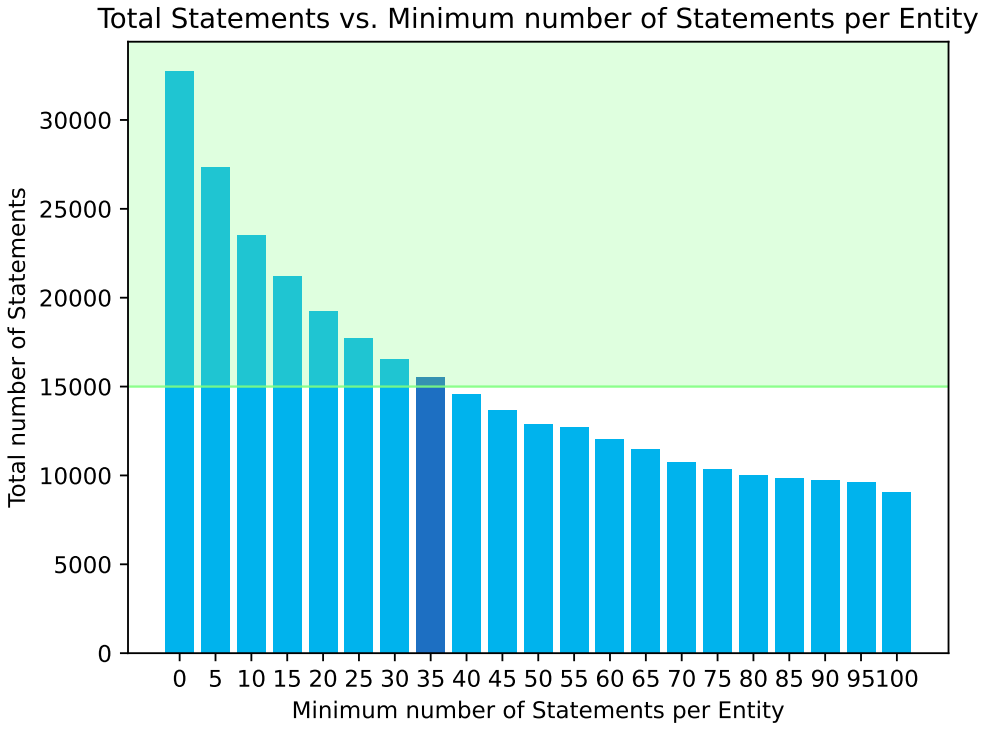

Figure 3.8: Minimum Statement Threshold - This figure shows how the minimum statements per entity threshold influences the total number of statements. The dark bar represents the chosen threshold of 35. The shaded area and the bards reaching it, represent thresholds that result in at least 15,000 descriptive statements.

Minimum Statement Count As previously stated, our definition of bias considers how prevalent a statement is within an ideology. The prevalence is not representative if the statement or similar statements only occur very

few times. Since many entities don't have a lot of associated statements, the expressiveness of their statements' bias labels might be limited. Therefore we want to exclude entities with a deficient statement count. We introduce another threshold exercising a minimum statement count. This threshold should not come at the cost of losing too many descriptive statements overall.

Figure 3.8 suggests, that if we want to work with over 15,000 descriptive statements overall, the threshold should be below 40. Therefore we choose a minimum of 35 descriptive phrases per entity. We think that within 35 statements, the most common information has a high chance of appearing multiple times.

Preprocessed Data Statistics After these preprocessing steps, we are left with 15.714 statements for 156 entities. The maximum number of statements per entity is 2543 (Trump) and the minimum is 35. The mean number of statements per entity is 100.73. 34.57% of the descriptive statements are labeled as "center". 38.04% are labeled as "left" and 27.39% are labeled as "right".

Preliminary Data Inspection To investigate the viability of our approach we conducted some early experiments. For the entity "Trump", we explored if any co-occurrences or n-grams have a prevalence in descriptive statements stemming from articles labelled with a certain bias.

co-occurrence	left occurrences	overall occurrences
climate, denier	3	3
reality, tycoon	3	3
star, television	2	3

Table 3.3: Co-Occurrences prevalent in Descriptive Statements from article labelled as *left*.

co-occurrence	right occurrences	overall occurrences
annual, growth	4	4
blue, collar	3	3
net, worth	3	3

Table 3.4: Co-Occurrences prevalent in Descriptive Statements from article labelled as *right*.

Although some observed prevalent co-occurrences seem to be caused by chance, such as "GOP, nominee" used in left articles 6 out of 6 times, our

preliminary experiments suggest that descriptive statements exhibit tendencies linked to ideology.

We continued to examine semantically similar statements, to further investigate if there were tendencies related to ideology affiliation among them. To do this, we encoded the descriptive statements for "Trump" using *SBERT*'s sentence embeddings [Nils Reimers, 2021] and grouped them by using agglomorotive clustering. An approach we will later also use to create a bias representation. (We will go into greater detail on this in **section 3.3.1**). To explore these clusters we built a web-gui.

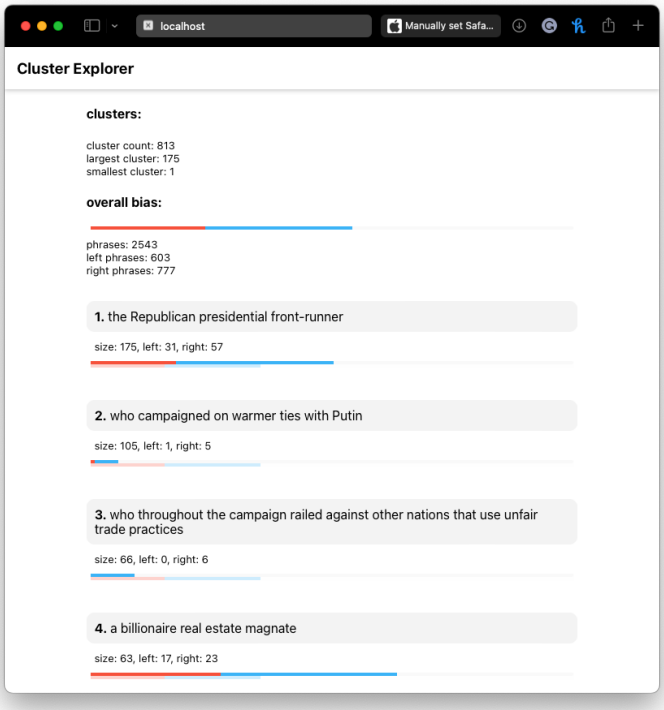

Figure 3.9: Cluster Explorer Interface - Each numbered statement contained in a gray box represents a cluster. The statement itself represents all similar statements within the cluster. The diagram with the red and blue sections is the "bias-meter". Red represents left bias, blue represents right bias. The opaque bias-meter represents the bias distribution within the cluster, the pale bias-meter presents the overall bias for comparison. Tapping on a cluster reveals each of the statements contained within the cluster.

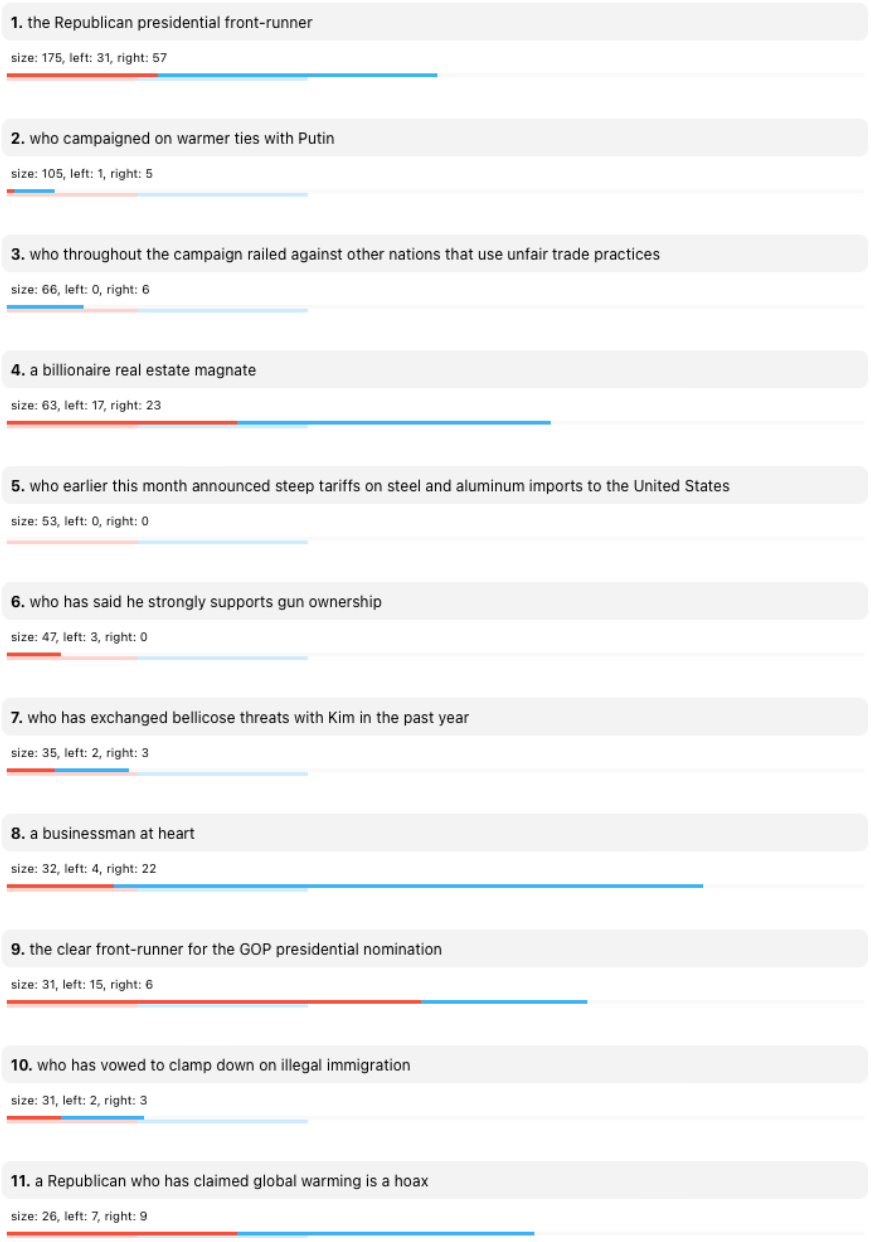

Figure 3.10: First 11 Clusters (by size) - Although the bias distribution of some clusters is not very distinctive others seem indicative of bias.

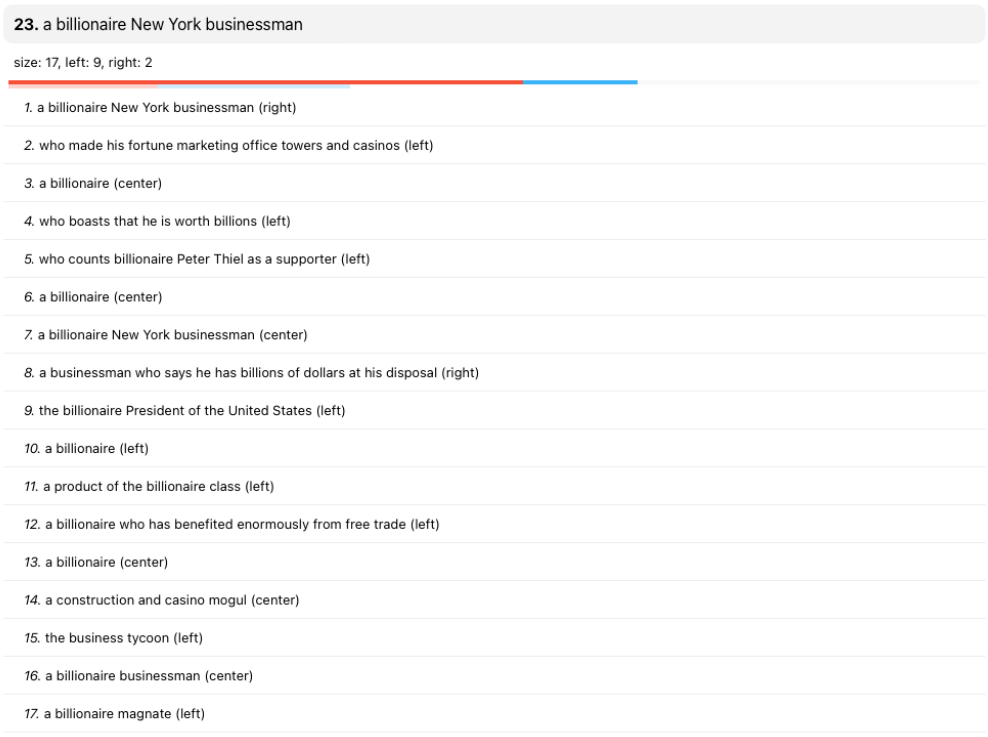

Figure 3.11: a cluster were the majority of statements come from left articles.

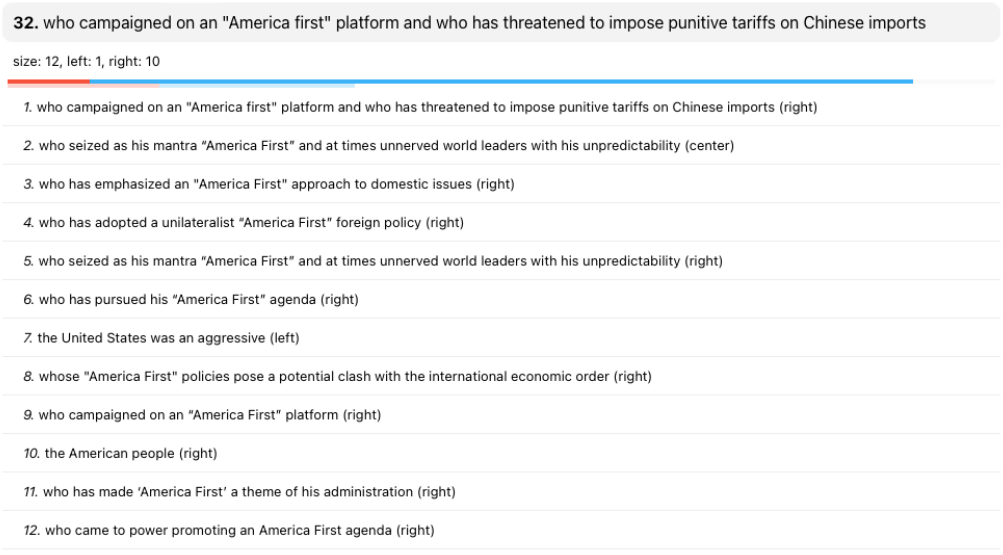

Figure 3.12: a cluster were the majority of statements come from right articles.

3.3 Model

Due to the novelty of our approach, our model should merely indicate the viability of our features. Therefore we prefer a simple model that allows for interpretability. Additionally, the limited data we have, does not allow for complex deep learning models. Thus, we opted for a basic *logistic regression* classifier.

3.3.1 Feature Engineering

In order to train a model, we must decide which information we deem relevant for the classification of an article's bias and how we can represent this information in a multi-dimensional numerical form.

Descriptive Statement The descriptive statement itself may be a feasible feature. As examined in our related work, lexical features i.e. word choice can be a signal of ideology. In order to be able to pick up on lexical characteristics, we created *bag of words* vector representation of the descriptive statements. We used the following pipeline: each statement was tokenized, stop words were removed and the remaining tokens were lemmatized. We used *SPACY* for this. From these lemmatized tokens, we created vectors were each component is associated with one of the tokens. The value of each component represents the number of occurrences of the token within the encoded statement. For vectorization, we used *SKLearn*'s *CountVectorizer*.

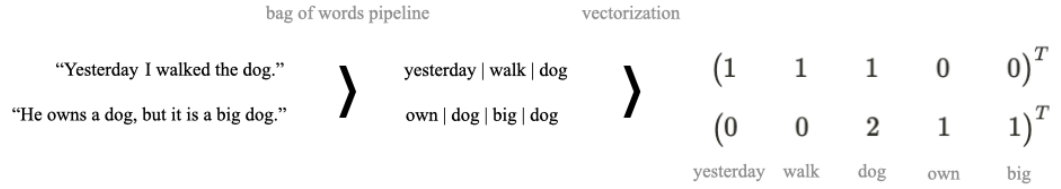

Figure 3.13: Descriptive statement lexical vector pipeline

Furthermore, our definition of bias is also based on the affiliation of a descriptive statements' information with an ideology. We believe that, if semantically similar statements are prevalent within articles of the same ideology in the training data, inputting unseen statements that also have similar semantics, is a strong indicator for the unseen source articles' biases. To represent semantic similarity, we used sentence embeddings from *SBERT* of the descriptive statements [Nils Reimers, 2021].

Target Entity The target entity of the descriptive statement may be important. How a statement frames an entity is dependent on context. This context may be beyond the articles content and concern general knowledge about the entity. Therefore statements on separate entities might be similar, but their effects might differ. This influence how authors use the statements depending on their outlet's ideology. The representation of the entity is a single numerical value from 1 to n, where n is the number of entities.

Bias Distribution The encodings of the descriptive statements themselves provide information on the distributions of bias labels for similar statements implicitly. However, we conceive this feature to be so important, that we engineer a feature representing the distributions of bias labels for similar statements explicitly. This gives us control over what to consider similar and how the distributions of bias labels should be interpreted.

The idea is to group training data examples by descriptive statement similarity. For each group, the bias distribution is determined by the biases of its members. This bias distribution is propagated down to each member as their bias distribution for training. When inferring the bias distribution for unseen examples, the group most similar to the given example is retrieved and its bias distribution will be used. **Firgure 3.14** illustrates this process.

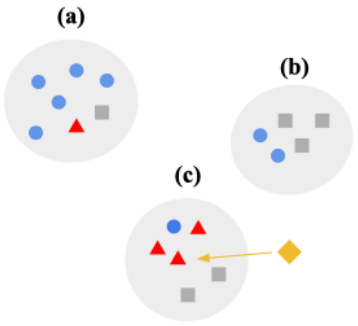

Figure 3.14: Bias Distribution Assignment - **(a)**, **(b)** and **(c)** are groups of similar descriptive statements. Their members (square, circle or triangle) are training examples. The members' shapes represent their biases. For each group, a bias distribution is calculated from its members labels and propagated back to the members. Therefore, each member in a group has the same bias distribution. The diamond represents an unseen example. Its descriptive statement is most similar to the statements of **(c)**. Therefore the diamond example is assigned the same bias distribution as the members of **(c)**.

In order to group the training examples, we employ *SKLearn*'s implemen-

tation of *agglomorotive clustering*. We chose this method since it does not need a predetermined number of clusters. Agglomerative clustering is a hierarchical bottom-up approach. Each example starts in its own single element cluster. Gradually the clusters get merged (Wikipedia [2021]). Whether clusters should be merged depends on two parameters. A *distance threshold* determines the maximum separation above which clusters are not merged. The *linkage type* controls where the distance between two clusters is measured i.e. between their centroids (average point), their closest, or their furthest points. This is illustrated in **Figure 3.15**.

These parameterizations can only be made with respect to the chosen metric; the function representing the distance between two clusters.

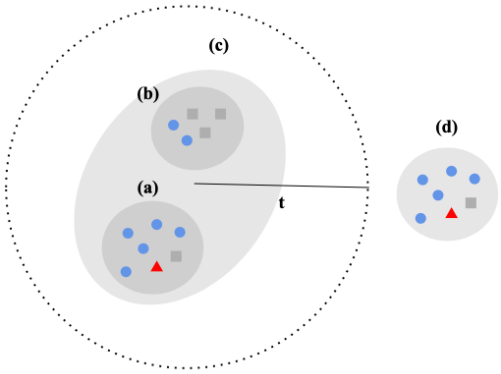

Figure 3.15: Cluster Distance Threshold - Clusters **(a)** and **(b)** get merged into cluster **(c)**, as the distance between their centroids is below threshold **t**. In this example linkage type is "average". In the next iteration, however, **(c)** and **(d)** will not be merged as their distance exceeds **t**.

We utilize two different metrics to assess descriptive statement similarity. Both metrics constitute a sub-variation of the bias distribution feature. We employed a bag of words approach in order to cluster statements, and thus to pick up on lexical cues. The descriptive statements were tokenized, stop words removed and the remaining tokens were lemmatized. The similarity between two descriptive statements' bag of words was assessed using the *Jaccard index*.

Let A and B be sets. Then the Jaccard Index is

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Wikipedia [2021]

In our case, A and B are sets of tokens, generated by the bag of words pipeline.

As argued before, we also want to group statements that encode the same information. Therefore we used semantic similarity as our metric. For this, we utilized *SBERT*s sentence embeddings and calculated their *cosine similarity.*

$$S_C(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

In our case, A and B are the embedded vector representations of the descriptive statements.

For both, the Jaccard index and cosine similarity metric, we needed to find an appropriate parameterization for distance threshold and linkage type. The parameterization should result in a clustering where elements are homogeneous within a cluster and heterogeneous between different clusters. Choosing these parameters needs to be done manually. We fixed linkage type to "average", allowing us to fine-tune the distance threshold. Due to the number of statements, judging the quality of a clustering is not a trivial task and needs to be systemically done to be adequate. Assessing whether clusters are heterogeneous, forces one to compare all clusters. This is not feasible manually. However, assessing whether a single cluster is homogeneous can be done by only assessing the cluster itself. Due to this, we start with a relaxed distance threshold and gradually tighten the threshold until it is strict enough to form a homogeneous outcome. This observation is done on the 50 largest clusters of the entity with the most statements (Trump). This should also be representative for all other entities, as this should not be entity-dependent. For the lexical metric, we chose a distance threshold of 0.8. Thus, for clusters c_1 and c_2 to get merged, a descriptive statement from c_1 must share at least 20% of the tokens for all statements from c_2 on average. For the semantic threshold, we chose 0.5. Thus, in order for clusters to get merged, the cosine similarity between their centroids must be at least 0.5.

Other sub-variations of the feature come from the calculation of the bias distribution of a cluster. One sub-variant encode the bias distribution as is. It is represented as a vector v, with $|v| = 3$, where each component v_i represents a bias label. The value v_i is represent the portion of its represented label within the cluster.

E.g. v_1 represents the label "center", with

$$v_1 = \frac{\text{number of examples with bias label "center" in the cluster}}{\text{number of examples in the cluster}}$$

The other sub-variant encode the distribution as a one-hot vector, where only the component representing the most frequent bias label has value 1 and all others are 0. The intuition is that having more extreme values might help the model to distinguish better.

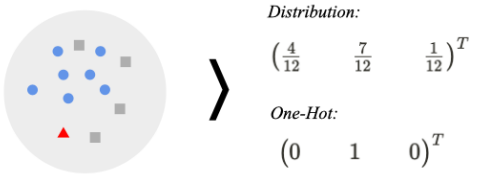

Figure 3.16: Bias Distribution Vector Sub-Variants - The *distribution bias vector* renders the portions for each bias label. 4 out of 12 examples are labeled as "square", 7 out of 12 as "circle" and 1 out of 12 as "triangle". The *one-hot bias vector* only returns the most frequent label, resulting in the component associated with "circle" to be 1 and all others 0.

The third sub-variation of the feature arises from what statements are clustered. As said before the frame established by the descriptive statement's information, might depend on the recipient's common knowledge of the entity. Thus, similar statements might have different effects depending on the entity and this, in turn, might affect the bias distribution. Therefore we have a "within entity" distribution feature, where each entities' descriptive statements are clustered separately and therefore, the calculated distributions are only be raised upon the entity itself. However, it is also possible that the influence of the target entity is minimal and that the feature profits of a larger selection of statements, which is why we also consider a "cross entity" distribution feature sub-variant, where the clustering is performed on all descriptive statements, disregarding the target entity.

The distance metric (*lexical* / *semantic*), the bias distribution calculation (*distribution* / *one-hot*), and the statement separation (*within-entity*, *across-entity*) are three sub-variations. Their combinations result in eight different bias distribution feature variants.

Context Fit Our definition of bias is not only based on the affiliation of a descriptive statement with an ideology, but also considers the relevance of a statement to its context. This is to take into account whether the information is tangential to its context and thus merely given to sway the reader's opinion on the entity.

The context fit feature should therefore represent the relevance of the descriptive statement to a context. We had to decide how this relevance can be calculated and what to consider as context. We concluded to utilize the semantic similarity between the descriptive statement and the context as our measurement of relevance. Although this might seem like a crude approach, our intuition is that broadly related information still bear a larger similarity than completely unconnected material. We use the *SBERT* embeddings and

their cosine similarity again. For the context, we agreed to create four different variations regarding different fragments of the statements source article as context.

We call the first variation *paragraph context fit*. This variation measures the similarity between the statement and the paragraph the statement occurred in. If a paragraph represents a unit of meaning, we believe the information within a paragraph should be somewhat related to the descriptive statement.

A close variation we call *window context fit*, considers the two sentences before and after the sentence containing the descriptive phrase. Similar to the intuition of the paragraph similarity, we suspect the information in the neighborhood of the source statement to bear a level of resemblance.

title context fit measures the similarity of the source article's title and the descriptive statement. The idea is that the title roughly indicates the article's topic and therefore relevant statements should be similar to an extent.

For the same argument *lead context fit* measures the similarity between to first three sentences and the descriptive statement. The beginning of articles often summarizes the content and gives background information. Therefore relevant descriptive statements should have some connection to these introductory sentences.

Feature	**Variations**
Descriptive Statement	Lexical Encoding Semantic Encoding
Target Entity	-
Bias Distribution	**Combinable sub-variations (8 Variations):** *Similarity Metric*: (lexical / semantic) *Bias vector*: (distribution / one hot) *Statement selection*: (cross-entity / within-entity)
Context Fit	paragraph context fit window context fit title context fit lead context fit

Table 3.5: Extracted Features - Considering all sub-variations combinations, we have a total of 15 features.

Chapter 4

Experiments & Results

To analyze the performances of our features, we broke our extracted descriptive statement data into training and testing splits, created multiple combinations of our engineered features, and used them to train and test logistic regression models. We evaluated the models performances, in order to assess the extent to which descriptive statements encode bias.

4.1 Experiments

4.1.1 Data Split

We decided to allocate around 90% of our data to training, leaving around 10% to testing. We have highlighted the importance of the entity to the descriptive statement multiple times throughout this thesis. Since the entity can be a feature itself, we ensured that the distribution of entities within the training and testing is the same, i.e. that the data is stratified with respect to the entities. However, we also want the bias label distribution to be similar within the training and testing splits, not only throughout the entire data but also on the entity level. This means, for each entity we want the distribution of bias labels to be the same in the training and testing data. Additionally, to not artificially enhance our results, we made sure, that no exact descriptive statement occurs in both; the training and the testing data. Also, there are no repeating statements within the testing data, where possible. This leaves us with 14,742 (93.81%) training and 972 (6.19%) testing examples. The deviation from the 90:10 distribution is due to our multiple constraints.

4.1.2 Implementation

Due to the novelty of the approach, we can not make assumptions about which features are important and how they may perform in combination. Therefore we chose to test all combinations of features, except single features. Considering we have 15 features, this results in $2^{15} - 16 = 32.768 - 16 = 32.753$ combinations. (16 combinations have only one or no features.) For the model, we chose *SKLearn*'s implementation of a logistic regression model. Since the creation of our features is runtime intensive, we derived the training and testing feature vectors from our data once and then only concatenated the vectors for each combination resulting in the final combination feature vector for each model. For each combination, we measured the performance using the model's accuracy.

$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

4.1.3 Upper & Lower Limits

To compare our results, we generated three other models. As a lower limit, we will use a model that guessing the bias label at random. For our upper limit, we wanted to see how approaches perform, that are not constrained to descriptive statements but have the same foundation otherwise. We created two logistic regression models trained on the large portions of the entire article. One model was trained on semantic characteristics. We used *SBERT*'s word embeddings on the article as feature vectors. However, *SBERT* is limited to 512 "word pieces", corresponding to the first 300-400 words of the articles (Nils Reimers [2021]). The other model was trained on lexical cues. We used the same tokenization-pipeline and token-count-vector representation as our lexical statement encoding feature.

4.2 Results

To get a better understanding of what feature combinations are influential, we grouped the feature combinations into cohorts. The cohorts are differentiated by the use of distribution and context features, but can all contain combinations using entity and descriptive statements features. The *distribution+context* cohort contains combinations using distribution and context features, the *distribution* cohort contains combinations using distribution features but no context features, *context* contains context features but not distribution features, and *plain* contains entity and descriptive statement encoding features only.

To evaluate our results we will start analyzing individual features. We will examine the *distribution* and *context* cohort individually and then move on to the overall results. We will weigh up the cohorts against each other and finally, we will compare our results to the lower and upper limits.

4.2.1 Individual Features

Table 4.1: Features ranked by Average Accuracy

	Feature	**Avg. Acc.**
1	distribution - *semantic metric, distribution bias vector, within-entity*	44.58%
2	distribution - *lexical metric, distribution bias vector, within-entity*	44.35%
3	entity	43.74%
4	distribution - *semantic metric, one-hot bias vector, within-entity*	43.65%
5	distribution - *lexical metric, one-hot bias vector, within-entity*	43.51%
6	distribution - *semantic metric, distribution bias vector, cross-entity*	43.42%
7	distribution - *semantic metric, one-hot bias vector, cross-entity*	43.33%
8	context - *paragraph*	43.25%
9	statement - *semantic*	43.24%
10	context - *window*	43.23%
11	context - *title*	43.23%
12	context - *lead*	43.22%
13	statement - *lexical*	43.02%
14	distribution - *lexical metric, one-hot bias vector, cross-entity*	42.77%
15	distribution - *lexical metric, distribution bias vector, cross-entity*	41.81%

Table 4.1 ranks all features by the mean accuracy of the combinations they occur in. The mean accuracies themselves are not deciding here, but the order of features should give a first feel for their compared viability, especially between feature variants. The distribution feature performs relatively well, as almost all of them place in the top ranks. When comparing distribution features only differing in metric, the *semantic metric* always achieves a higher mean accuracy compared to the *lexical metric*. The same can be said for the *distribution bias vector* and *one-hot bias vector*, as every combination that only differs in the bias vector representation, obtains a higher mean accuracy when using the *distribution bias vector*. All distribution variants using the *within-entity* clustering outperform the *cross-entity* setting. The *entity* features places at rank 3. This might be tied to the *within-entity* clustering setting. On average the context features perform similarly, with their scores only varying by around 0.01%. The *semantic* encoding for the statement itself outperforms the *lexical* encoding.

4.2.2 Distribution Cohort

Table 4.2: Top 5 Performing Distribution Feature Combinations

	Feature Combination	Acc.
1	entity distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	50.82%
2	entity distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity*	50.41%
3	entity distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity*	50.31%
4	distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity*	50.21%
5	distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity*	50.21%

The top five distribution feature combinations score 50.39% on average. Consistent with **Table 4.1**, all distribution variants use the *semantic metric*. In fact, the first combination to utilize a distribution feature employing the *lexical metric* only occurs at rank 33. This strongly indicates that the semantic clustering results in a more potent distribution representation. Also congruent with **Table 4.1**; the *distribution bias vector* is more prevalent compared to the *one-hot* representation. In **Table 4.2** there is no combination with a *one-hot bias vector* representation, that does not include the same feature with a *distribution bias vector* representation, thus suggesting, that the one-hot-features is indifferent and dispensable. In the listed combinations in **Table 4.2**, the *cross-* and *within-entity* variations are primarily used simultaneously. However, in contrast to **Table 4.1**, the *cross-entity* sub-variation seems to be favored. None of the first five combinations use the statement as a feature. The average accuracy in the *distribution* cohort is 43.93%, 30.76% is the minimum.

4.2.3 Context Cohort

Table 4.3: Top 5 Performing Context Feature Combinations

	Feature Combination	**Acc.**
1	entity statement - *semantic* statement - *lexical* context - *paragraph* context - *window* context - *lead*	49.49%
2	entity statement - *semantic* statement - *lexical* context - *title*	49.38%
3	entity statement - *semantic* statement - *lexical* context - *title* context - *window*	49.38%
4	entity statement - *semantic* statement - *lexical* context - *paragraph* context - *title* context - *window*	49.38%
5	entity statement - *semantic* statement - *lexical* context - *paragraph* context - *lead*	49.28%

The top five feature combinations in the *context* cohort score an accuracy of 49.38% on average. Similar to **Table 4.1**, considering **Table 4.3** no order of importance can be deduced from the context feature variants. However, all listed combinations contain the semantic and lexical encodings of the descriptive statement. When ordered by accuracy, the first 32 combinations from the *context* cohort all include the semantic representation of the descriptive statement itself and 76.67% of the combinations using the semantic representation score above the *context*-median. This illustrates how important the semantic encoded statement feature is for the *context* cohort. A possible explanation could be, that the context itself can not be used for classification across the bias labels, as it only signals the similarity between the descriptive statement and the context. The semantic encoding of the statement, however, can be used to link similar examples to an ideology. The average accuracy in the

context cohort is 45.14%, the minimum is 36.62%.

4.2.4 All Combinations

Table 4.4: Top 10 Performing Feature Combinations

	Feature Combination	**Acc.**
1	entity context - *paragraph* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	51.03%
2	entity context - *title* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	51.03%
3	entity context - *title* context - *paragraph* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	50.93%
4	entity distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	50.82%
5	entity context - *title* context - *paragraph* context - *window* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	50.82%

Continued on next page

Table 4.4 – continued from previous page

	Feature Combination	**Acc.**
6	entity context - *paragraph* context - *window* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	50.72%
7	entity context - *window* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	50.62%
8	entity context - *paragraph* context - *window* context - *lead* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity* distribution - *semantic metric, one-hot bias vector, cross-entity* distribution - *semantic metric, one-hot bias vector, within-entity*	50.62%
9	entity context - *window* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity*	50.51%
10	entity context - *paragraph* context - *window* distribution - *semantic metric, distribution bias vector, cross-entity* distribution - *semantic metric, distribution bias vector, within-entity*	50.51%

Table 4.4 states that the best performing feature combinations have around 51% accuracy. The differences between the combinations of successive ranks are minute, as they only vary by a single feature. This and the similarity of the scores hint at robust results, as it suggests that the top combinations and scores are not caused by chance, but that combinations of the kind presented in **Table 4.4** are actually more indicative of bias than others. All of the top combinations contain the entity feature. None of the combinations contain an encoding of the descriptive statement, however, they all contain distribution features. This suggests that the listed combinations perform better with the explicitly modeled distribution feature, compared to the implicit distribution

given by the statements themselves. 9 out of the 10 highest performing feature combinations are from the *distribution+context* cohorts. When combining distribution and context, *title* and *paragraph* seem to be the strongest context features, as they occur frequently in the top five combinations. The top eight combinations all feature all of the semantic distribution variations. All ten combinations contain semantic distribution features utilizing the *distribution bias vector*, as seen in the *distribution* cohort. The overall mean accuracy is 43.41%.

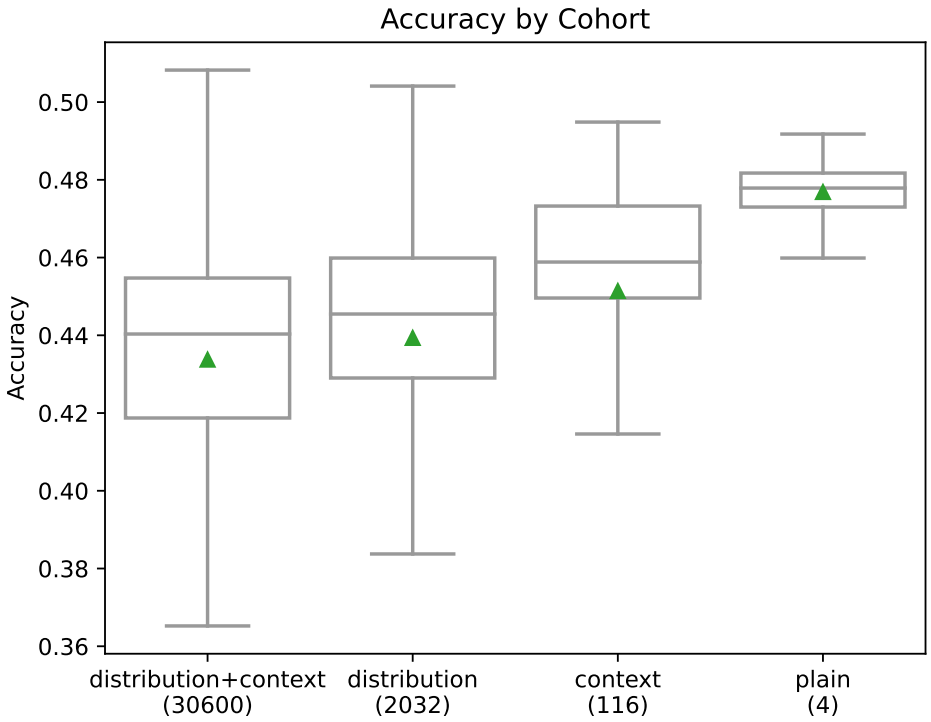

Figure 4.1: Accuracy by Cohort - the x-axis represent the different cohorts. The number below the cohort-label displays the number of combinations within the cohort. The y-axis represents the accuracy. The triangle indicates the mean accuracy of the cohort. The box represents the interquartile range. The horizontal line within the box represents the median value. The whiskers represent the maximum and minimum (excluding outliers).

Figure 4.1 compares all accuracies of the cohorts. It suggests that the range of accuracies of a cohort is correlated to the number of feature combinations evaluated within the cohort. This is plausible because more combinations increase the chances for outliers. The maximums and minimums of the interquartile ranges, the mean and median accuracy seem to be inversely

Cohort	Size	Mean Acc.	Max Acc.	Min Acc.
distribution+context	30600	43.37%	51.03%	30.04%
distribution	2032	43.93%	50.82%	30.76%
context	116	45.14%	49.49%	36.63%
plain	4	47.69%	49.18%	45.99%

Table 4.5: Accuracy Scores by Cohort

correlated to the number of combinations. This relationship, however, can not be explained, as it would suggest that more combinations perform worse on average than fewer combinations. We assume this inverse correlation to be a coincidence. Therefore, **Figure 4.1** and **Table 4.5** displays that the *distribution+context* cohort performs worst on average and that *plain* feature combinations perform best.

This suggests that many feature combinations from the *distribution+context* cohort are low-grade, except for a few exceptions. Also, the *plain* cohort performs above the overall average for all its four combinations.

4.2.5 Lower & Upper Limit

	Model	Accuracy
1	Article - *Lexical*	80.97%
2	Article - *Semantic*	73.97%
3	Our Best Performing Combination	51.03%
4	Guessing Model	33.33%

Table 4.6: Limits Accuracy Scores

Table 4.6 displays how our best model compares to the our lower and upper limits. We were able to outperform a guessing baseline by around 18%. However the semantic upper limit performed around 23% better than our model and the lexical upper limit surpassed our model by around 30%. We will the exceptional performance of our upper limits in the next chapter.

Chapter 5

Limitations, Future Work & Conclusion

5.1 Limitations

Our approach in this thesis relies heavily on the affiliation of a descriptive statement with an ideology with respect to the statement's target entity. Therefore, the availability of a suitable sized pool of statements for each considered entity can be crucial to get a representative notion of affiliation for the most common statements. We use the threshold of a minimum of 35 statements per entity. A larger threshold may lead to more meaningful representation.

The detection method for a descriptive statement achieved around 87% accuracy. However, more advanced detection mechanisms may lead to higher accuracy.

Although most of the retrieved descriptive statements were of a general information nature, some were very specific. Specific information may not be recurring and can therefore not be associated with an ideology. Statements like this might have introduced some noise into the data.

The way our source data has been labeled holds some weaknesses. The descriptive statements were labeled using their source article, which in turn, was labeled by their publisher. Hence, the resulting label for a descriptive statement may not be accurate.

As explored in the related work, the *BASIL* corpus contains an article for each ideology reporting on the same event. This ensures that the models are not trained on the discussed topics, which may have a tendency to occur less or more frequently depending on the ideology. This leakage might have positively impacted the performances of our descriptive statement models and especially our upper limits, as they were trained on the entire article.

5.2 Future Work

Due to the novelty of our approach, we constrained our scope to people and the left-right political spectrum. Future work may consider more named entities, such as organizations or locations. The bias spectrum may be expanded to a multidimensional space, such as the Political Compass, representing the economic (left-right) and social (authoritarian-libertarian) axis (Wikipedia [2021]). Also, a hard classification unsuitable for bias and altered to a continuous value on a spectrum.

Our definition of a *descriptive statement* is very constraining, thus making its occurrence sparse. Employing a different strategy to detect introductory information could result in the ability to retrieve a wider range of initial information and therefore an extended collection of bias signals.

Future work may also leverage the collection of descriptive statements and their distributions for debiasing documents by exchanging bias affiliated descriptive statements against neutral ones.

5.3 Conclusion

This thesis addresses political bias in a certain part of article discourse, namely, the text part that introduces people. In particular, we have proposed a tangible definition for introductory information: "descriptive statement", asking the question of *To what extent does a descriptive statement encode bias?*

To answer this question, we have built a pipeline to retrieve such statements and used disambiguation to group them by 'entity'. We have performed a series of preprocessing steps on our data, enabling it to be used as a corpus for training a bias classification model.

We have engineered features that reflect our definition of bias by representing ideology affiliation and the contextual relevance of a descriptive statement. We trained and tested models for predicting article-level bias for all encouraging feature combinations. To have a comparison, we included a lower baseline model that guesses the article bias label. We also trained two upper limit models; a model trained on a semantic representation of the article and a model trained on a lexical representation of the article.

The results of our experiments have demonstrated that our best-performing feature combination surpasses our baseline by around 18% with an accuracy of 51.03%. Even the average feature combination performs around 10% better than the baseline. The upper limits performed exceptionally well. The semantic model scored an accuracy of 73.97% and the lexical model scored an accuracy of 80.97%. The upper limits were expected to exceed our approach, as they can access more information than our model.

Ultimately, we observe that the classification of the article bias using descriptive statements works relatively well, especially considering the limited information a statement itself manifests. Hence, as an answer to our research question: descriptive statements encode bias to an extent that allows them to be used for article-level bias detection to a certain degree.

Bibliography

Jonathan S Blake et al. *News in a digital age: Comparing the presentation of news information over time and across media platforms*. Rand Corporation, 2019.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=5k8F6UU39V`.

Wei-Fan Chen, Khalid Al-Khatib, Benno Stein, and Henning Wachsmuth. Detecting media bias in news articles using gaussian bias distributions. *arXiv preprint arXiv:2010.10649*, 2020.

Lisa Fan, Marshall White, Eva Sharma, Ruisi Su, Prafulla Kumar Choubey, Ruihong Huang, and Lu Wang. In plain sight: Media bias through the lens of factual reporting. *arXiv preprint arXiv:1909.02670*, 2019.

Felix Hamborg, Karsten Donnay, and Bela Gipp. Automated identification of media bias in news articles: an interdisciplinary literature review. *International Journal on Digital Libraries*, 20(4):391–415, 2019.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1113–1122, 2014.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, 2019.

Nils Reimers. Input sequence length. `https://www.sbert.net/examples/applications/computing-embeddings/README.html`, 2021. Accessed: 2021-01-03.

Andrew Radford. *Relative clauses: Structure and variation in everyday English*, volume 161. Cambridge University Press, 2019.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659, 2013.

Wikipedia. bias. `https://en.wikipedia.org/wiki/Bias`, 2021. Accessed: 2021-11-27.

Tae Yano, Philip Resnik, and Noah A Smith. Shedding (a thousand points of) light on biased language. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 152–158, 2010.