Methods for Detecting and Mitigating Linguistic Bias in Text Corpora

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN

Dr. rer. nat.

genehmigte Dissertation
von

M. Sc. Christoph Hube

geboren am 27. Juli 1988, in Langenhagen, Deutschland

Hannover, Deutschland, 2020

Referent: Prof. Dr. techn. Wolfgang Nejdl
Korreferent: Prof. Dr. Robert Jäschke
Korreferent: Prof. Dr. Michael Rohs
Tag der Promotion: 29.05.2020

ABSTRACT

With the ongoing expansion of the Web into all aspects of people's everyday lives, bias on the Web becomes an increasingly challenging problem. A common manifestation is biased text. To counter bias, the online encyclopedia Wikipedia has introduced the *Neutral Point of View* (NPOV) principle, demanding the use of neutral language instead of partial or inflammatory phrasing. While studies have shown that Wikipedia articles exhibit quality that is comparable to conventional encyclopedias, research still proves that Wikipedia, overall, is prone to many different types of NPOV violations. Identifying biases can be very challenging, even for humans, and with millions of articles and a decreasing number of contributors, this task becomes increasingly more difficult to handle. When given room, bias can not only lead to polarization and conflicts between opinion groups but can also negatively affect users in their free forming of a personal opinion. In addition, bias in text and ground-truth data can negatively impact machine learning models trained on this data, leading to discriminatory model behavior.

In this thesis, we address the problem of bias by focusing on three central aspects: biased content in the form of written statements, bias of crowd workers during the process of data annotation, and bias in word embedding representations.

We present two approaches for detecting biased statements in text corpora, such as Wikipedia. Our feature-based approach relies on bag-of-word features including a bias word list that we obtained by identifying clusters of bias words in the vector space of word embeddings, while our improved neural-based approach makes use of gated recurrent neural networks to capture context dependencies, further improving the performance of the model.

Our study on crowd worker bias reveals biased behavior among workers with strong opinions on a given topic and shows that this behavior affects the resulting ground-truth labels, impacting dataset creation for tasks such as bias detection or sentiment analysis. We present approaches for worker bias mitigation by creating awareness among workers and making use of the concept of social projection.

Finally, we address the problem of bias in word embeddings, focusing on the example of varying sentiments of names. We show that biases in the training data are captured by the embeddings and passed on to downstream models. In this context, we introduce a debiasing approach that reduces the bias effect and positively affects the resulting labels of a downstream sentiment classifier.

Keywords: *Text Bias, Bias Detection, Bias Mitigation, Debiasing*

ZUSAMMENFASSUNG

Im Zuge der fortschreitenden Ausbreitung des Webs in alle Aspekte des täglichen Lebens wird Bias in Form von Voreingenommenheit und versteckten Meinungen zu einem zunehmend herausfordernden Problem. Eine weitverbreitete Erscheinungsform ist Bias in Textdaten. Um dem entgegenzuwirken hat die Online-Enzyklopädie Wikipedia das Prinzip des neutralen Standpunkts (*Englisch:* Neutral Point of View, *kurz:* NPOV) eingeführt, welcher die Verwendung neutraler Sprache und die Vermeidung von einseitigen oder subjektiven Formulierungen vorschreibt. Während Studien gezeigt haben, dass die Qualität von Wikipedia-Artikel mit der Qualität von Artikeln in klassischen Enzyklopädien vergleichbar ist, zeigt die Forschung gleichzeitig auch, dass Wikipedia anfällig für verschiedene Typen von NPOV-Verletzungen ist. Bias zu identifizieren, kann eine herausfordernde Aufgabe sein, sogar für Menschen, und mit Millionen von Artikeln und einer zurückgehenden Anzahl von Mitwirkenden wird diese Aufgabe zunehmend schwieriger. Wenn Bias nicht eingedämmt wird, kann dies nicht nur zu Polarisierungen und Konflikten zwischen Meinungsgruppen führen, sondern Nutzer auch negativ in ihrer freien Meinungsbildung beeinflussen. Hinzu kommt, dass sich Bias in Texten und in Ground-Truth-Daten negativ auf Machine Learning Modelle, die auf diesen Daten trainiert werden, auswirken kann, was zu diskriminierendem Verhalten von Modellen führen kann.

In dieser Arbeit beschäftigen wir uns mit Bias, indem wir uns auf drei zentrale Aspekte konzentrieren: Bias-Inhalte in Form von geschriebenen Aussagen, Bias von Crowdworkern während des Annotierens von Daten und Bias in Word Embeddings Repräsentationen.

Wir stellen zwei Ansätze für die Identifizierung von Aussagen mit Bias in Textsammlungen wie Wikipedia vor. Unser auf Features basierender Ansatz verwendet Bag-of-Word Features inklusive einer Liste von Bias-Wörtern, die wir durch das Identifizieren von Clustern von Bias-Wörtern im Vektorraum von Word Embeddings zusammengestellt haben. Unser verbesserter, neuronaler Ansatz verwendet Gated Recurrent Neural Networks, um Kontext-Abhängigkeiten zu erfassen und die Performance des Modells weiter zu verbessern.

Unsere Studie zum Thema Crowd Worker Bias deckt Bias-Verhalten von Crowdworkern mit extremen Meinungen zu einem bestimmten Thema auf und zeigt, dass dieses Verhalten die entstehenden Ground-Truth-Label beeinflusst, was wiederum Einfluss auf die Erstellung von Datensätzen für Aufgaben wie Bias Identifizierung oder Sentiment Analysis hat. Wir stellen Ansätze für die Abschwächung von Worker Bias vor, die Bewusstsein unter den Workern erzeugen und das Konzept der sozialen Projektion verwenden.

Schließlich beschäftigen wir uns mit dem Problem von Bias in Word Embeddings, indem wir uns auf das Beispiel von variierenden Sentiment-Scores für Namen konzentrieren. Wir zeigen, dass Bias in den Trainingsdaten von den Embeddings erfasst und an nachgelagerte Modelle weitergegeben wird. In diesem Zusammenhang stellen wir einen Debiasing-Ansatz vor, der den Bias-Effekt reduziert und sich positiv auf die produzierten Label eines nachgeschalteten Sentiment Classifiers auswirkt.

Schlagwörter: *Text Bias, Bias Detection, Bias Mitigation, Debiasing*

ACKNOWLEDGMENTS

I would like to thank everyone who supported me during the course of my PhD. I thank Prof. Wolfgang Nejdl for his guidance, especially during the time when I started to look for an interesting and promising topic, and Prof. Robert Jäschke for his support and the inspiring discussions during countless meetings. Special thanks go out to my dear friends and colleagues at L3S, i.e. Besnik, Ujwal, Bill, and many more. You not only helped me shape as a researcher but also made sure that I had a lot of fun during the process.

Furthermore, I would also like to thank all my friends and family outside of L3S, particularly my parents who I can always count on. Finally, I thank my girlfriend Dana for her unconditional love and support.

It's not the years in your life. It's the life in your years.

FOREWORD

During the course of my Ph.D studies, I published papers on bias and related topics. The core contributions presented in this thesis have been published in the following venues:

1. The introduction in Chapter 1 contains parts from:

   - [Hub17] Hube, Christoph. "Bias in Wikipedia." Proceedings of the 26th International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee, 2017.

2. The contributions of Chapter 3 on creating a *feature*-based approach for bias detection in text have been published in:

   - [HF18] Hube, Christoph, and Besnik Fetahu. "Detecting Biased Statements in Wiki-pedia." Companion Proceedings of the The Web Conference 2018. International World Wide Web Conferences Steering Committee, 2018.

3. The contributions of Chapter 4 on creating a *neural*-based approach for bias detection in text have been published in:

   - [HF19] Hube, Christoph, and Besnik Fetahu. "Neural Based Statement Classification for Biased Language." Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. ACM, 2019.

4. The study on worker bias understanding and mitigation presented in Chapter 5 has been published in:

   - [HFG19] Hube, Christoph, Besnik Fetahu, and Ujwal Gadiraju. "Understanding and Mitigating Worker Biases in the Crowdsourced Collection of Subjective Judgments." Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, 2019.

5. The work on bias in words embeddings presented in Chapter 6 is currently under submission:

    - Hube, Christoph, Maximilian Idahl, and Besnik Fetahu. "Debiasing Word Embeddings from Sentiment Associations in Names." *Under submission*.

Other publications in the context of my Ph.D studies are shown below:

1. [HFG18] Hube, Christoph, Besnik Fetahu, and Ujwal Gadiraju. "LimitBias! Measuring Worker Biases in the Crowdsourced Collection of Subjective Judgments.", CrowdBias@HCOMP, 2018.

2. [HJF18] Hube, Christoph, Robert Jäschke, and Besnik Fetahu. "Towards Bias Detection in Online Text Corpora.", IConference, 2018.

3. [GFH16] Gadiraju, Ujwal, Besnik Fetahu, and Christoph Hube. "Crystal clear or very vague? Effects of task clarity in the microtask crowdsourcing ecosystem." 1st International Workshop on Weaving Relations of Trust in Crowd Work: Transparency and Reputation Across Platforms, Co-located With the 8th International ACM Web Science Conference. 2016.

4. [HFJ+17] Hube, Christoph, Frank Fischer, Robert Jäschke, Gerhard Lauer, and Mads Rosendahl Thomsen, (2017). World Literature According to Wikipedia: Introduction to a DBpedia-Based Framework. arXiv preprint arXiv: 1701.00991.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Bias on the Web

With the ongoing expansion of the Web into all aspects of people's everyday lives, bias on the Web becomes an increasingly challenging problem.

Baeza-Yates [BY18] has identified a "vicious cycle of bias on the Web" that depicts the different types of biases and their interconnections (Figure 1.1). One central aspect of this cycle is *data bias*, i.e. the content of the Web being biased due to the diverging backgrounds and opinions of its users. In addition, *activity bias* describes the phenomenon that a large fraction of content is in fact provided by a comparably small number of active content creators and that the demographics of this group do usually not represent the population well. For instance, on Wikipedia a large number of articles is written by only a very small number of editors and less than 12% of these editors are women, leading to an underrepresentation of this group [BY18]. Furthermore, a lot of algorithms rely on data sampled from the Web. This can lead to bias being passed on to models which, in turn, influence the behavior of Web users. Second-order bias describes how users, being influenced by data-driven models, shape the Web and its content. Baeza-Yates states that in order to overcome this "cycle of bias", awareness of bias must be raised.

A common form of data bias on the Web is biased text. Opinionated text is widely spread across the Web, reaching from social media, forums, blogs, and shopping platforms to information sources such as news or encyclopedias. Freedom of speech and the expression of opinions is a central pillar of our modern society. Many platforms and systems encourage users to share their opinions on a large number of topics with others, driving the public discourse and helping others to form their own opinions. Typical examples of opinionated texts are blog posts, tweets, product reviews, and comments.

Apart from openly expressed opinions there is a more subtle form of "opinion pushing", where authors force their personal stances on a supposedly neutral piece of text. This is the type of opinionated text that is usually referred to as *biased*. Bias does not imply that authors are aware of their behavior or even be aware of their own stance. Writing perfectly

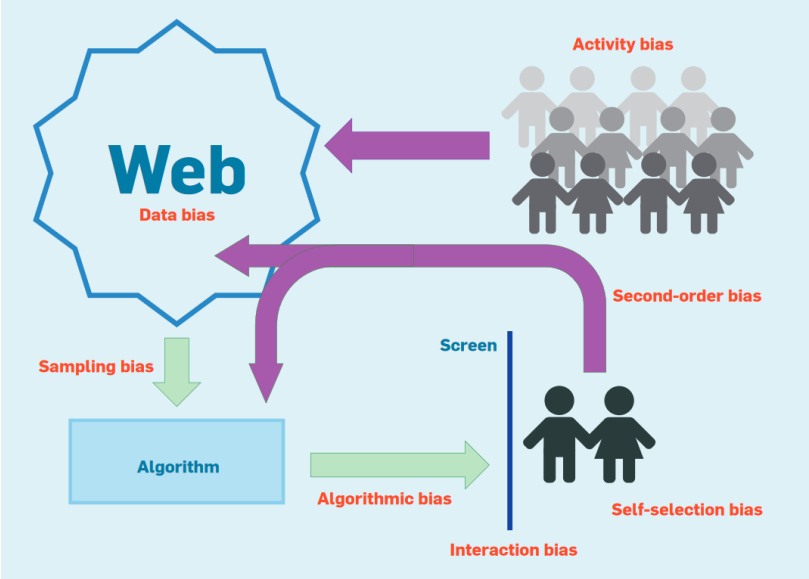


Figure 1.1. *The vicious cycle of bias on the Web* shows types of biases and their interconnections. Source: [BY18]

neutral text is a challenge on its own and biased text might be added unintentionally.

1.2 Bias in Wikipedia

With more than 30 million registered editors and an unknown number of unregistered collaborators, the open encyclopedia Wikipedia is one of the largest collaboration platforms on the Web and ranks number 10 of the most visited sites on the Web[1]. The Neutral Point of View (NPOV)[2] is one of the main principles of Wikipedia. It demands Wikipedia editors to put their personal opinions on a topic aside and create objective content. According to the NPOV policy, within an article, all important opinions on a topic should be represented without any attempt on trying to convince the reader of any of the presented views. The policy explicitly encourages the use of nonjudgmental language and demands that facts should not be stated as opinions while "seriously contested assertions" should not be stated as facts.

While studies have shown that Wikipedia articles exhibit quality that is comparable to conventional encyclopedias [Gil05], research still proves that Wikipedia, overall, is prone to many different types of NPOV violations that are caused by biases resulting from its editors. These biases often revolve around controversial topics such as for example gender [WGJS15], culture [CH11], and politics [GZ12a]. In some cases, where editors do not find an agreement, the disputes about the content of an article lead to an edit war. Two

[1] https://www.alexa.com/topsites, Aug. 2019
[2] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

or more opposing sides each struggle to strengthen one position in the article's content by permanently adding text supporting their own point of view or removing text that supports the opponent's point of view (typically by reverting the edit). In the end, the group putting more effort or simply being larger in number may succeed to push their point of view on the article. This can lead to a situation where minority opinions are simply overwhelmed.

As an example of how bias can be introduced to an encyclopedic statement by the choice of phrasing, take the following three statements, all taken from articles about the same person:

a) Andrew James Breitbart was an American ultraconservative, far right militant, publisher, commentator for "The Washington Times", author, and occasional guest commentator on various news programs.

b) Andrew James Breitbart was an American conservative publisher, writer and commentator.

c) Andrew James Breitbart was one of the most outspoken, fearless conservative journalists in America.

Statement a) has been extracted from an older version of Wikipedia while statement b) is taken from the current Wikipedia version (Aug. 2019). Statement c) has been extracted from Conservapedia[3], a website that presents itself as an alternative to Wikipedia, focusing on conservative views and content. Statement b) can be considered neutral since it does not give any hints on the author's opinion on the topic. Statement a) is comparably more controversial, especially the use of the terms "ultraconservative" and "far right militant" which might not be inline with common agreement. In statement c), the bias that is introduced by the editor and the editor's personal stance towards the topic of the article are even clearer. The use of the adjectives "outspoken" and "fearless" gives the statement a very positive connotation.

Finding biases can be very challenging, even for humans. So far Wikipedia relies predominantly on its voluntary contributors to find and remove biased statements. But with millions of articles in over 200 language versions and a decreasing number of contributors, this task becomes increasingly more difficult to handle.

When given room, bias can not only lead to polarization and conflicts between opinion groups but can also negatively affect users in their free forming of a personal opinion. Therefore, the detection and mitigation of bias becomes an important challenge for the future of platforms such as Wikipedia and the Web in general.

[3] https://www.conservapedia.com

1.3 Scope of the Thesis and Contributions

Bias can be introduced in many different ways, not always being directly observable for the end user. Before being accurately addressed, bias has to be identified first. In the case of Wikipedia statements, bias can often simply be removed after detection, while in other cases bias mitigation is not trivial. Apart from detection and mitigation, we also aim to raise awareness about the impact and consequences of bias.

We address the problem of bias by focusing on three central aspects: biased content in the form of written statements, bias of crowd workers during the process of data annotation, and bias in word embedding representations. A statement in our case corresponds to a sentence in Wikipedia or a similar context.

Detecting Biased Statements Despite the fact that Neutral Point of View is one of Wikipedia's central policies, many Wikipedia articles contain NPOV violations in the form of opinionated text. Approximately 40,000 Wikipedia pages have been marked by editors with NPOV or similar quality issues and new content is constantly added. Currently, Wikipedia has to rely on its voluntary contributors to identify and remove biased statements in Wikipedia articles. Automatizing this process is an important step towards helping contributors to improve and maintain the quality of articles.

Linguistic bias (also referred to as *language* or *phrasing bias*) is a type of bias that is common in Wikipedia articles. It refers to bias being introduced by how a statement is phrased, i.e. the choice and arrangement of words. We study how the use of specific words (*bias words*) relates to a statement being considered as biased or not. To this end, we propose an approach for generating a lexicon of bias words and release the most comprehensive list of bias words so far. Making use of this lexicon and a comprehensive set of other features, we introduce an automatic approach for detecting biased statements in Wikipedia articles. For a given statement, the model decides if it is biased or neutral/unbiased with respect to phrasing bias. Motivated by examples of biased statements that do not contain an explicit bias word, we introduce a second, neural-based approach for bias detection that aims to capture the context of words within a statement. The neural-based approach is able to identify biased statements without having to rely on word lists, e.g. the biased statement "The public agrees that it is the number one country in the world." does not contain opinionated words but can be identified as biased by the arrangement of words ("the public agrees", "the number one").

Our models are trained on Wikipedia data but the approaches are not limited to Wikipedia, i.e. they can be used in all other contexts where phrasing bias occurs. As part of the process, we describe procedures for extracting POV-tagged statements from Wikipedia and creating high quality training datasets for bias detection using crowdsourcing. In the context of this work, we released two large corpora of biased statements.

The contributions have been published in:

- [HF18] Hube, Christoph, and Besnik Fetahu. "Detecting Biased Statements in Wikipedia." Companion Proceedings of the The Web Conference 2018. International World Wide Web Conferences Steering Committee, 2018.
- [HF19] Hube, Christoph, and Besnik Fetahu. "Neural Based Statement Classification for Biased Language." Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. ACM, 2019.

Understanding and Mitigating Crowd Worker Bias Crowdsourcing is widely used for human annotation of data, such as ground-truth data. On platforms like FigureEight[4] and Mechanical Turk[5], crowd workers label datasets in exchange for a monetary compensation. The annotated data can then be used for training and testing models. In the case of creating a dataset for detecting biased statement, crowd workers were given a statement at a time and had to label it as either biased or not biased. In many cases full agreement between crowd workers is not reached with some workers giving diverging labels to the same statement, especially for tasks that comprise a subjective component, e.g. bias detection or sentiment analysis. Crowdsourced data acquired from these tasks is potentially affected by the personal opinions of the contributing crowd workers. This can lead to biased and noisy ground-truth data, propagating the undesirable bias and noise when used in turn to train machine learning models or evaluate systems.

We conduct a study on crowd worker bias that addresses the following research questions:

RQ#1: How does a worker's personal opinion influence their performance on tasks including a subjective component?

RQ#2: How can worker bias stemming from strong personal opinions be mitigated within subjective tasks?

RQ#3: How does a worker's experience influence their capability to distance themselves from their opinion?

To answer these questions, we propose a novel *measure of worker bias* in subjective tasks that is based on the misclassification rates of workers in combination with the worker's personal opinions. This allows us to answer **RQ#1** and **RQ#3**, given that FigureEight provides a worker experience level for each worker. To answer **RQ#2**, we introduce three novel *techniques for mitigating systemic worker bias* stemming from personal opinions. Furthermore, we study the impact of worker bias on the aggregated ground-truth labels and show that a significant percentage of the final labels are decided by a majority of potentially biased workers.

[4] https://www.figure-eight.com, the platform changed its name from CrowdFlower to FigureEight in 2018
[5] https://www.mturk.com

Our contributions are an important step towards understanding the bias of crowd workers stemming from their personal opinions and reducing the negative effect that crowd worker bias has on the resulting data. This, in turn, contributes towards improving the training data for a bias detection model as well as for other types of models (e.g. sentiment analysis, opinion detection).

The contributions have been published in:

- [HFG19] Hube, Christoph, Besnik Fetahu, and Ujwal Gadiraju. "Understanding and Mitigating Worker Biases in the Crowdsourced Collection of Subjective Judgments." Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, 2019.

Debiasing Word Embeddings Word embeddings are widely used for representing text input, e.g. for text classifiers. Typically trained on large text corpora, such as Wikipedia or the Google News collection, they capture word semantics based on the co-occurrence of words in the text. At the same time, they also have been shown to be prone to capturing the biases from the training corpus [BCZ+16]. These biases are unwanted as they can spill into downstream models, leading to discriminatory behavior. For example, it has been shown that word embeddings of names can be biased towards gender and race [CBN17]. To obtain bias-free embeddings, the existing bias has to be reduced.

We propose an approach for debiasing word embeddings directly during training. For a given list of words (e.g. names), the approach modifies the respective word embeddings in a way that the words can not be associated anymore with any protected class (positive or negative sentiment in our case). To the best of our knowledge, this is the first approach that includes not only words of the protected classes but also *proxies*, i.e. words that are not part of the protected class but can still be used to recover the bias of words. In addition, we make use of standard benchmark tests to show that the quality of the embeddings is maintained during debiasing.

Furthermore, we conduct an analysis on the downstream example task of sentiment analysis including multiple datasets (news data from a set of diverse sources and IMDB[6] movie reviews) and study the effect that debiasing has on the resulting labels. Our contributions pave the way towards bias-free word embeddings and models. This also has a positive effect on the quality of word embeddings being used as part of a bias detection model and, consequently, on the quality of the model itself.

The contributions are under submission:

- Hube, Christoph, Maximilian Idahl, and Besnik Fetahu. "Debiasing Word Embeddings from Sentiment Associations in Names." *Under submission*.

[6] https://www.imdb.com/

1.4 Outline of the Thesis

The remainder of this thesis is organized as follows.

In Chapter 2, we give an overview over the technical background on *supervised learning*, *neural networks*, and *word embeddings* and describe the main concepts used in this thesis.

In Chapter 3, we present the feature-based approach for detecting biased statements in Wikipedia and show that it classifies biased statements with a precision of up to 74%. We introduce the features that the model leverages and describe how we extract bias words from the English Wikipedia.

Next, in Chapter 4, we present the neural-based approach for detecting biased statements and show that it outperforms the approach introduced in Chapter 3 by a margin, achieving a precision of up to 92%. In this context, we describe how we make use of features such as *Attention* and *LIWC word functions* to improve the model.

Chapter 5 introduces our study on crowd worker bias. We present our measure for worker bias and reveal that crowd workers with strong opinions on a topic are likely to show biased behavior. We introduce three approaches for bias mitigation and show their individual performance. Finally, we show that even experienced workers are prone to worker bias and that a large percentage of final labels are decided by the groups of potentially biased workers.

In Chapter 6, we present our approach on debiasing word embeddings. We focus on the example of sentiment associations in names and show that our approach reduces the bias measured for names compared to the plain skip-gram approach. We reveal the impact of debiasing on the final labels of a downstream sentiment classifier and show that our approach increases homogeneity across sentiment labels for sentences containing names.

Finally, in Chapter 7, we draw conclusions and discuss opportunities for future work.

2 Technical Background

In this chapter, we give an overview of the technical background that is necessary to understand the work presented in this thesis. First, we will introduce the concepts of supervised learning and classification that are essential for a significant part of our work. Next, we will provide an overview of neural networks and the deep learning techniques that we use for our neural-based bias detection approach, as introduced in Chapter 4. Finally, we will discuss word embeddings, a concept that we make heavy use of in this thesis, especially in Chapter 6, where we introduce our approach for debiasing word embeddings.

2.1 Supervised Learning

Supervised Learning refers to the subset of machine learning algorithms that learn a function $f : X \rightarrow Y$ from a given set of training examples, each being represented by a set of input features $X = x_1, x_2, ..., x_i$ and an output Y. After training, the learned function is used to predict the output of unknown examples.

Classification is a type of supervised learning where the final output label y belongs to a set of k classes and the algorithm produces a function $f : R^n \rightarrow \{1, ..., k\}$ that maps the input to one of these classes (or to multiple classes in the case of multi-label classification). The special case of $k = 2$ is called *binary classification*. The output is often regarded as a probability distribution over all classes. The input features can be structured or unstructured as in the case of text or images. For text, the input can be the words itself, typically in some form of numerical representation, such as one-hot encodings or word embeddings. The resulting classification model is called a *classifier*. An example of text classification is *sentiment analysis*, where the classifier, for a given input text, has to predict the sentiment class of the input (e.g. positive or negative).

The input set is typically separated into a train and a test set[1]. The error on the train set it called train error and the error on the test set is called test error. If the train error is high, the

[1] A separate validation set is used for hyperparameter optimization.

model is underfitting, meaning it is not able to predict the output of the training instances, using its input features. If the training error is low but the test error is high, the model is overfitting. In this case, the model is able to learn the training input, but fails to generalize to other, unknown examples. The process of reducing overfitting while maintaining an acceptably low train error is called *regularization*.

Random Forest Random Forest is a part of ensemble learning, a supervised learning technique used for classification and regression. The first algorithm was initially introduced by Tim Kan Ho [Ho95]. In Random Forest, the set of input instances is split into k subsets and each subset is fed as input to a decision tree, resulting into k separate decision trees. Each tree produces an output label. The final label is decided by majority voting, which is the mode of the predicted classes in the case of classification. Random Forest algorithms have been shown to be less prone to overfitting compared to decision trees [HTF08]. In this thesis, we use the scikit learn [PVG+11] implementation of Random Forest.

Feature Selection Feature selection is the process of identifying the most valuable features for the classification. Removing less valuable features reduces the overfitting effect due to less redundancy in the data and allows for faster training. In this thesis, we make use of the χ^2 feature selection algorithm. In statistics, χ^2 is used to test the independence of two events. In feature selection we can use it to test whether the occurence of a class and a feature are independent. Formally, it is defined as

$$\chi^2(D,t,c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}, \tag{2.1}$$

where D is the document, t the term (the feature in our case), and c the class. N is the observed and E the expected frequency. e_t takes the value 1 if the document contains t and the value 0 otherwise. e_c takes the value 1 if the document is in class c and the value 0 otherwise. High values of χ^2 indicate that the null hypothesis, i.e. feature and class are independent, can be rejected. In this case, the feature should be selected.

2.2 Neural Networks

Neural networks have been shown to provide state-of-the-art solutions for many supervised learning tasks. A simple type of neural networks are feedforward neural networks (also called multilayer perceptrons). One of the main advantages of neural networks over classic machine learning approaches is that they not only learn the function f but at the same time learn the parameter values that result in the best function approximation, thus creating a feature representation of the input x. A feedforward neural network defines the mapping $y = f(x;\theta)$, where θ is the set of parameters. Feedforward neural networks consist of multiple layers of neurons with weighted connections between each neuron of two adjacent

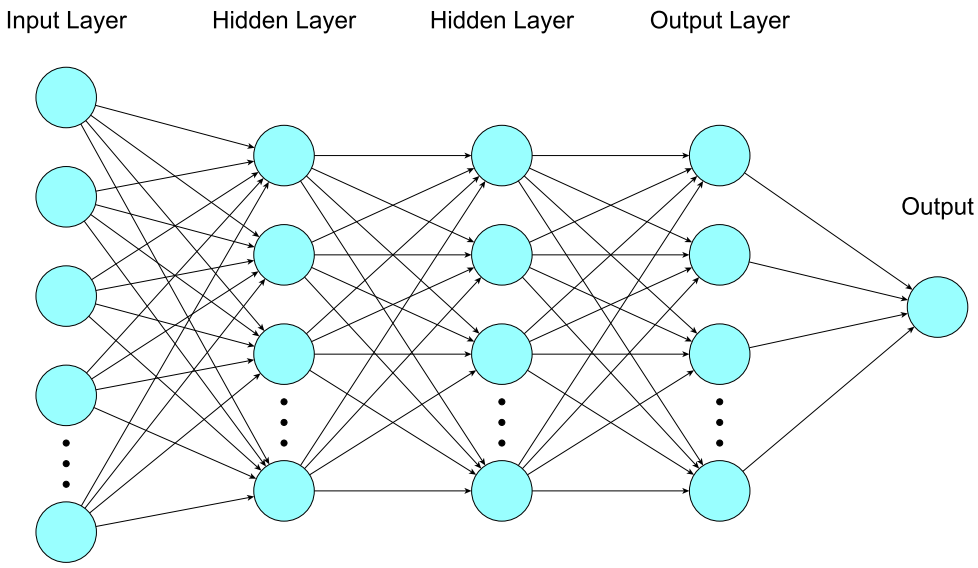


Figure 2.1. Feedforward neural network with two hidden layers.

layers (as depicted in Figure 2.1), including an input layer, an output layer, and a number of hidden layers. The output of a layer is given by

$$h = g(Wx + b), \tag{2.2}$$

where g is the activation function, W are the weights, and b is the added bias[2]. The process of passing an input through the network to create an output is called *forward pass*. During the training process, the optimal values for W and b are learned. The network can be seen as a concatenation of functions, where each layer is one function and the depth of the network is the number of layers. The term *deep learning* refers to neural networks with a large number of hidden layers, while the term *feedfoward* refers to the neural network not having feedback connections, i.e. information is constantly passed on through the network from the input layer to the output layer without any loops.

Activation functions The activation function g is used to add non-linearity to a neural network. Purely linear networks fail to solve some problems, e.g. a linear neural network can not represent the XOR function [GBC16]. In our neural networks, we make use of three different activation functions, i.e. ReLu, sigmoid, and softmax.

The Rectified linear unit (ReLu) is a piecewise linear function defined as

$$a = max(0, z), \tag{2.3}$$

[2]Note that this term *bias* is not related to the bias that we address in this thesis. It just refers to the value that is added to a neuron after multiplication of W and x.

where z is the output before applying the activation function. ReLu is known to counter the vanishing gradient problem [GBC16]. The function is depicted in Figure 2.2a. We use ReLu activation for all hidden layers in this thesis.

The sigmoid function is defined as

$$a = \frac{1}{1 + \exp(-z)}. \tag{2.4}$$

It is depicted in Figure 2.2b. We use sigmoid activation in the output layer in the case of binary classification.

The softmax function is given as

$$a = \frac{\exp(z_i)}{\sum_k \exp(z_k)}, \tag{2.5}$$

where z_i is the output for class i. It is depicted in Figure 2.2c. We use softmax activation for classification with more than two classes.

Cost functions The cost function defines the loss between the model's output and the given training label. The computed loss is used by the optimizer to update the parameter values of the neural network.

In this thesis, we make use of the cross-entropy, a cost function that is used for classification tasks where the output label is a class probability (value between 0 and 1). Binary cross-entropy is used for the task of binary classification, i.e. classification with only two classes. It is defined as

$$H_{y'}(y) := -\sum_i (y'_i \log(y_i) + (1 - y'_i) \log(1 - y_i)), \tag{2.6}$$

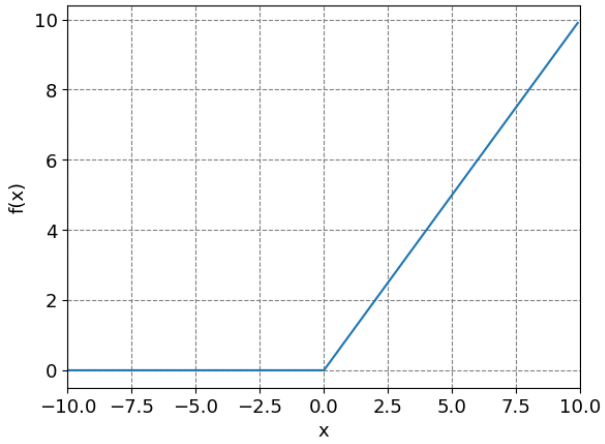
where y_i is the predicted probability for the class i and y'_i is the actual probability.

In the case of more than two classes, the cross-entropy is defined as

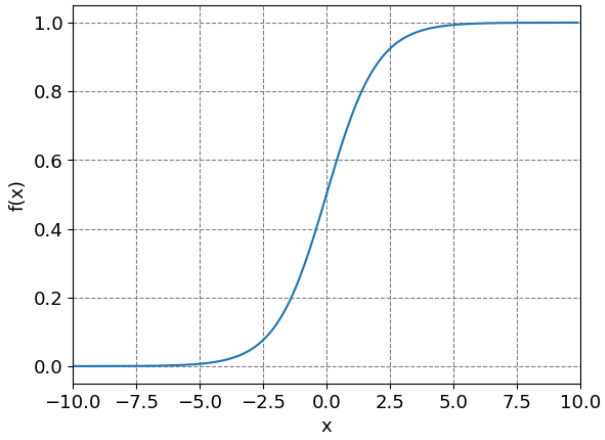
$$H_{y'}(y) := -\sum_i y'_i \log(y_i). \tag{2.7}$$

For multiple classes, we predominantly make use of the categorical cross-entropy which is simply the cross-entropy combined with softmax.

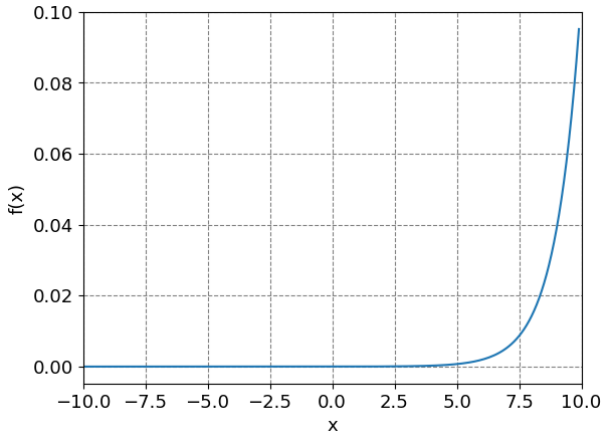
Optimizers Optimization is the process of fitting the model to the training data by adjusting the parameter values (weights and biases) based on the computed cost and the learning rate that defines the learning speed. The algorithm used for optimization is called the *optimizer*. In a neural network with multiple layers, the loss has to be back-propagated through the network. This is usually done by making use of the chain rule. A common



(a) ReLu



(b) Sigmoid



(c) Softmax

Figure 2.2. Activation functions.

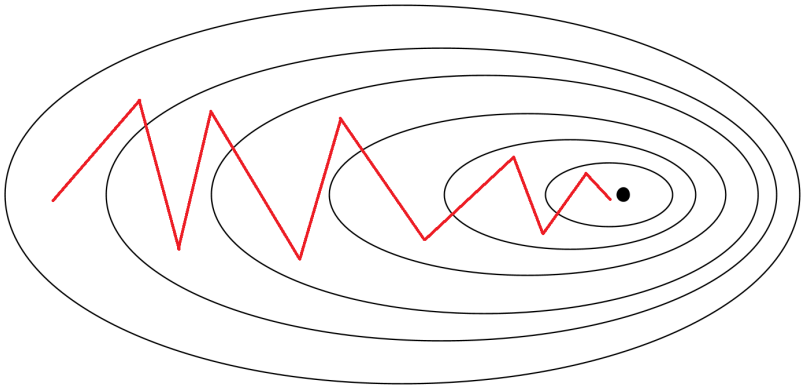


Figure 2.3. Stochastic Gradient Descent

approach for optimization in neural networks is the use of Stochastic Gradient Descent (SGD) in combination with back-propagation. For each input example, SGD computes the gradient of the loss and adapts the network's parameters to reach a global minimum in a step-wise procedure (see Figure 2.3).

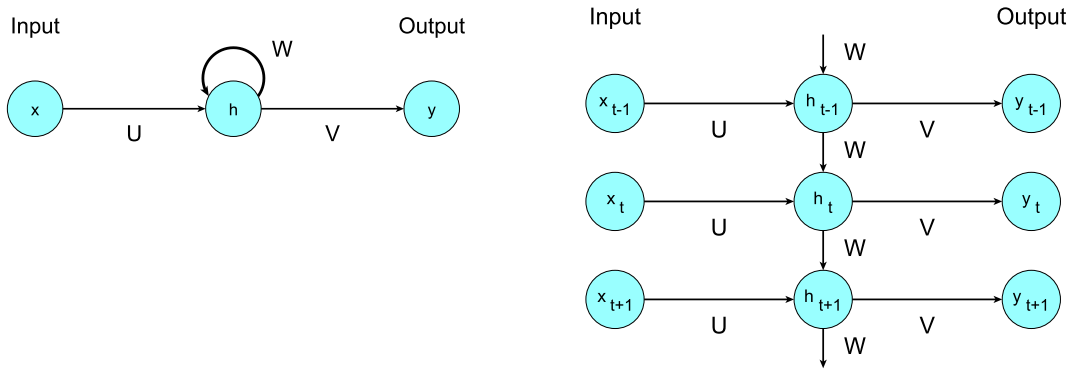
In this thesis, we make use of the *Adam* ("adaptive moments") optimizer, which is basically an extension of SGD, including an adaptive learning rate with momentum. Instead of updating after each input instance, we process the input in batches of a given batch-size.

Cost functions, activation functions, and optimizers are not restricted to use within neural networks but can also be applied to other machine learning algorithms. In this thesis, we use keras [C+15] with the Tensorflow backend for creating neural networks.

Regularization To counter overfitting and help the model generalize to unseen data (possibly at expanse of the training accuracy), we apply *dropout* regularization. Dropout sets the output of neurons in the neural network to 0, using a binary mask. The probability for each neuron to be "dropped out" is defined by the *dropout rate*. Dropout forces the network to adapt to situations with incomplete information. It is similar to the concept of bagging in that it produces a set of diverse networks. In contrast to bagging, the parameters for weights and biases are shared across all networks.

2.2.1 Recurrent Neural Networks

Recurrent neural networks (RNNs) are not purely feedforward but contain feedback connections. They are typically applied to sequences, such as sequences of words, processing each element of the sequence step by step. For effective learning, they make use of a concept called *parameter sharing*. The parameters for the weights and biases are shared across all time steps, allowing the model to generalize to different input lengths and improve training. In addition, the hidden state is passed on from one time step to the next time step, so that the model can take previous time steps into account. The architecture of an RNN is depicted in



(a) RNN with loop in the hidden state

(b) Unfolded depiction

Figure 2.4. Two depictions of the same recurrent neural network. 2.4a shows the RNN with a feedback connection in the hidden layer. The weight matrix W is passed on from one time step to the next, resulting in a hidden state for each time step. 2.4b shows the unfolded depiction of the network for three time steps. At each time step, the RNN reads one element of the input sequence and produces an output.

Figure 2.4. U and V are separate weight matrices for the input and output layers. Forward propagation through the network for one time step t is defined as follows

$$a_t = b + Wh_{t-1} + Ux_t. \tag{2.8}$$

The output of each time step is computed using

$$y_t = g(c + Vh_t), \tag{2.9}$$

where c is an additional bias vector for the output layer.

RNNs have been used for different types of tasks, e.g. for sequence-to-sequence tasks such as machine translation [BCB14]. For classification, we include an additional feedforward layer with sigmoid or softmax activation before the output, reducing the number of output neurons according to the number of classes. We use pooling to adapt to diverging input lengths.

Bi-directional RNNs combine a forward RNN with a second RNN that processes the sequence in reverse. This allows the model to include both past and future information. We make use of bi-directional RNNs for covering both past and future dependencies.

Gated RNNs Gated recurrent neural networks, such as Long Short-Term Memory networks (LSTMs) [HS97] and networks containing Gated Recurrent Units (GRUs) [CVMG+14], introduce gates to counter the vanishing gradient problem. The vanishing gradient problem

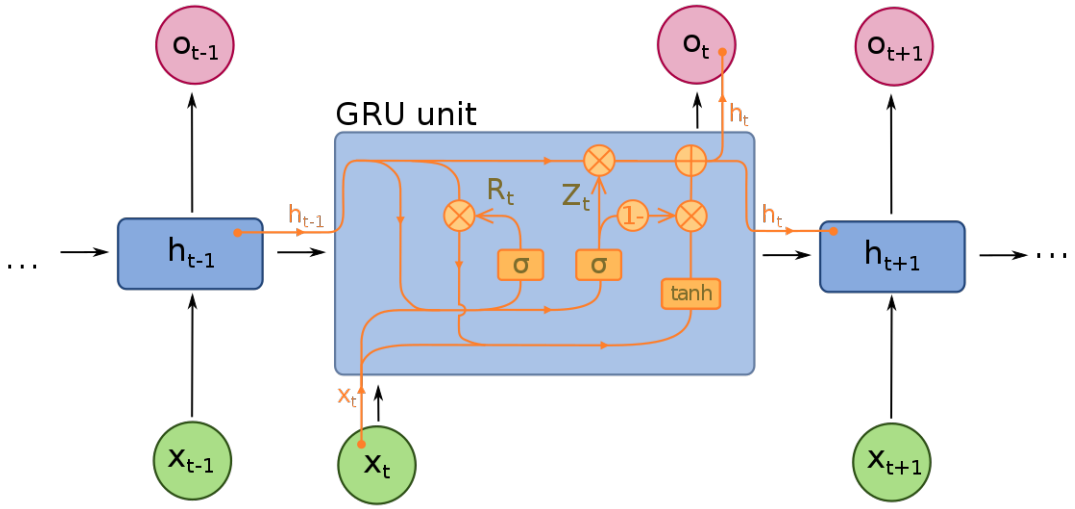


Figure 2.5. Architecture of an RNN with a Gated Recurrent Unit.
Source: https://en.wikipedia.org/wiki/File:Gated_Recurrent_Unit.svg

defines the situation of the gradient becoming close to 0 because of a high number of multiplications with small values. This can a) make the network stop completely from changing its weights during back-propagation training without reaching a global minimum, or b) lead to past information being lost in an RNN when information is passed on through the hidden states from time step to time step. While there are other techniques for treating case a) (e.g. ReLu activation), Gated RNNs counter case b) by allowing information to flow unchanged through the time steps of the RNN.

Figure 2.5 illustrates the architecture of a GRU unit. Similarly to a plain RNN, the unit receives the past hidden state h_{t-1} and the current input x_t and produces the next hidden state h_t and an output o_t. Internally it contains an *update gate* Z_t that is trained to decide what information from the previous time steps will be forgotten and a *reset gate* R_t that, together with the update gate, decides what new information will be added. The main differences between GRUs and LSTMs are that GRUs merely rely on the hidden state and do not need an additional cell state and that they reduce the number of gates from three to two gates. GRUs and LSTMs have been shown to achieve similar performances [CGCB14]. We explain GRUs in more detail in chapter 4.

Attention The attention mechanism was first introduced by Bahdanau et al. [BCB14] as part of an *encoder-decoder* structure for neural machine translation. The encoder-decoder is a specific structure that uses a stack of RNNs (the *encoder*) to create a representation of the input that is then used by another stack of RNNs (the *decoder*) to produce the final output. Attention allows the decoder to observe not only the created input representation but also the intermediate outputs of the encoder at each time step. By learning *attention weights* during training, the attention mechanism learns to focus on specific parts of the input sequence.

In the case of a simple RNN or Gated RNN, the attention mechanism allows the model

to focus on parts of the input sequence instead of just relying on the output of the final layer. This way, the model does not have to combine all information into one vector but can make use of the learned attention weights instead. Apart from machine translation, attention has been used for different types of tasks, including classification tasks [YYD+16]. In Chapter 4, we introduce two attention mechanisms, i.e. global and hierarchical attention. It has been shown that while attention is useful to increase the performance of models, the attention weights can not generally be used to explain the model's decision [JW19].

2.3 Word Embeddings

Word embeddings have become a widely used approach to represent text data in natural language processing. Since neural networks were designed to work on numerical data, any text has to be converted to numbers before being used as input for a neural network. Typically, text input is represented as a sequence of words, where each word is mapped to a vector, though other forms of representations exist, e.g. ngrams, phrases. A trivial approach are one-hot-encodings where each word is represented by one dimension in the vector space, i.e. a vector that contains $|V| - 1$ zeros and a single 1 at the position of the given word, where $|V|$ is the size of the vocabulary. This is problematic, especially for large vocabularies, where the number of dimensions becomes equally large. In addition, one-hot-encodings do not convey any semantic meaning from the training data, because the difference between each pair of words is identical, e.g. the euclidean distance between the words *lion* and *tiger* in the vector space is equal to the distance between the words *lion* and *boat*. To overcome the curse of dimensionality and to arrange words in a semantically meaningful way, word embeddings have been introduced.

In 2013, Mikolov et al. [MSC+13a] presented word2vec, a group of approaches for training highly semantical word embeddings using neural networks and large text corpora for training. Compared to the vocabulary size, the number of dimensions of the resulting embeddings is small, typically between 50 and 300 dimensions. Mikolov et al. show that these representations capture semantic meaning of the training data, not only in the distance of words, but also in their arrangement. E.g., when computing $king - man + woman$, the result is close to the word $queen$ in the vector space, as depicted in Figure 2.6. This highly improves the ability of neural networks to learn from text data.

Skip-gram Skip-gram is one of the two approaches introduced by Mikolov et al. in the context of word2vec. It produces word embeddings by using a feedforward neural network with one hidden layer to learn the context of a given word based on examples taken from the training dataset. For instance, for the input sentence *The lion roars*, it considers *The* and *roars* as the context of the word *lion* and defines the model's error as the distance of the predicted vector from these context words (using a specified cost function). After training, the weights learned by the model form the produced word embeddings. The architecture of skip-gram is shown in Figure 2.7.

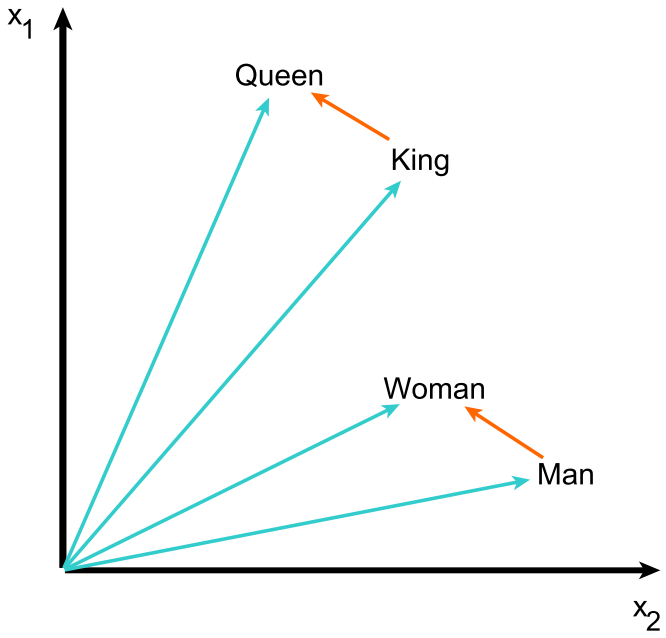


Figure 2.6. Example of the arrangement of words in the vector space.

Even though word embeddings have significantly improved the performance of NLP models, some shortcomings of approaches like word2vec have been identified. One of these shortcomings is the handling of words with multiple meanings, e.g. there is typically only one embedding for both the fruit and the company *apple*. A new generation of highly contextualized word embeddings, including ELMo [PNI+18] and BERT [DCLT18], addresses this problem by connecting word embeddings directly to their context. Word embeddings have also been shown to capture the bias of the data that they are trained on, as we will show in chapter 6.

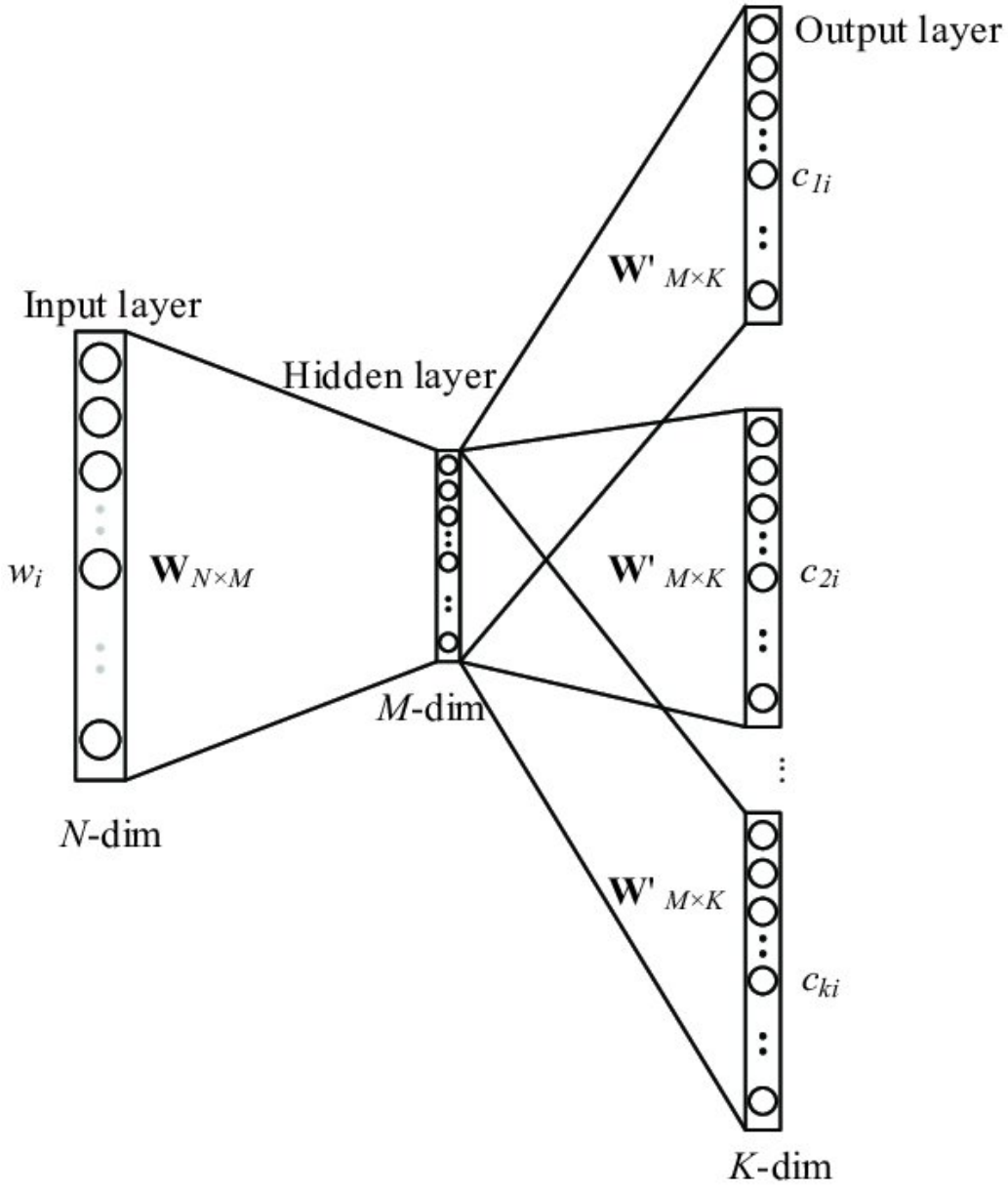


Figure 2.7. Architecture of the skip-gram model. Source: [HLY+18]

3

Detecting Biased Statements in Wikipedia

Wikipeda is one of the largest collaboratively created encyclopedias. Its community of editors consist of more than 32 million registered editors only in the English Wikipedia. However, only a small minority, specifically 127,000 editors are active[1]. Due to the diverse demographics and interests of editors to maintain the quality of the provided information, Wikipedia has a set of editing guidelines and policies.

One of the core policies is the *Neutral Point of View* (NPOV)[2]. It requires that for controversial topics, Wikipedia editors should proportionally represent all points of view. The core guidelines in NPOV are to: (i) avoid stating *opinions as facts*, (ii) avoid stating seriously *contested assertions as facts*, (iii) avoid stating *facts as opinions*, (iv) prefer *nonjudgemental language*, and (v) indicate the relative *prominence of opposing views*.

Currently, there are approximately 40,000 Wikipedia pages that are flagged with NPOV quality issues (or similar quality flaws). These represent explicit cases[3] marked by Wikipedia editors, where specific Wikipedia pages or statements (sentences in Wikipedia articles) are deemed to be in violation with the NPOV policy. Recasens et al. [RDNMJ13] analyze these cases that go against the specific points from the NPOV guidelines. They find common linguistic cues, such as the cases of *framing bias*, where subjective words or phrases are used that are linked to a particular point of view (point (iv)), and *epistemological bias* which focuses on the believability of a statement, thus violating points (i) and (ii). Similarly, Martin [Mar17] shows the cases of biases which are in violation with all guidelines of NPOV, an experimental study carried out on his personal Wikipedia page[4].

Ensuring that Wikipedia pages follow the core principles in Wikipedia is a hard task. Firstly, due to the fact that editors provide and maintain Wikipedia pages on a voluntarily basis, the editor efforts are not always inline with the demand by the general viewership of Wikipedia [WRTH15] and as such they cannot be redirected to pages that have quality issues.

[1] https://en.wikipedia.org/wiki/Wikipedia:Wikipedians#Number_of_editors
[2] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view
[3] This number may as well be much higher for cases that are not spotted by the Wiki-pedia editors.
[4] https://en.wikipedia.org/wiki/Brian_Martin_(social_scientist)

Furthermore, there are documented cases, where Wikipedia admins are responsible for policy violations and pushing forward specific points of view on Wikipedia pages [DLMI13, GZ12b], thus, going directly against the NPOV policy.

In this chapter, we address quality issues that deal with language bias in Wikipedia statements that are in violation with the points (i) – (iv). We classify *statements* as being *biased* or *unbiased*. A *statement* in our case corresponds to a *sentence* in Wikipedia. We address one of the main deficiencies of related work [RDNMJ13], which focuses on detecting *bias words*. In our work, we show that similar to [MCQ08], words that introduce bias or violate NPOV are dependent on the context in which they appear and furthermore the topic at hand. Thus, our approach relies on an automatically generated lexicon of bias words for a given set of Wikipedia pages under consideration, and in addition to semantic and syntactic features extracted from the classified statements.

As an example of language bias consider the following statement:

- Sanders shocked his fellow liberals by putting up a Soviet Union flag in his Senate office.

The word *shocked* introduces bias in this statement since it implies that *"putting a Soviet Union flag in his office"* is a shocking act.

To this end, we make the following contributions in this chapter:

- propose an automated approach for generating a lexicon of bias words from a set of Wikipedia articles under consideration,
- an automated approach for classifying Wikipedia statements as either *biased* or *unbiased*,
- a human labelled dataset consisting of biased and unbiased statements.

3.1 Related Work

Research on bias in Wikipedia has mostly focused on different topics such as culture, gender and politics [CH11, WGJS15, IEBGR14] with some of the existing research referring to language bias.

Greenstein and Zhu[GZ12a] analyze political bias in Wikipedia with a focus on US politics. They use the approach introduced by Gentzkow and Shapiro[GS10] that was initially developed to determine newspaper slant. It relies on a list of 1000 terms and phrases that are typically used by either republican or democratic congress members. Greenstein and Zhu search for these terms and phrases within Wikipedia articles about US politics to measure in which spectrum (left or right leaning politics) these articles are. They find that articles on Wikipedia used to show a more liberal slant in average but that this slant has decreased over time with the growth of Wikipedia and more editors working on the

articles. For the seed extraction part of the approach we present in this chapter, we also use articles related to US politics but instead of measuring political bias, our approach simply detects biased statements using features that are not directly related to the political domain and therefore can also be used outside of this domain. Our extracted bias lexicon contains mostly words that are not directly related to politics.

Iyyer et al.[IEBGR14] introduce an approach based on Recursive Neural Networks to classify statements from politicians in US Congressional floor debates and from ideological books as either liberal or conservative. The approach first splits the sentences into phrases and classifies each phrase separately before incrementally combining them. This allows for more sophisticated handling of semantic compositions. For example the sentence *They dubbed it the "death tax" and created a big lie about its adverse effects on small businesses* introduces liberal bias even though it contains the more conservatively biased phrase *death tax*. For sentence selection they use a classifier with manually selected partisan unigrams as features. Their model reaches up to 70% accuracy.

Yano et al.[YRS10] use crowdsourcing and statements from political blogs to create a dataset with the degree of bias and the type of bias (liberal or conservative) given for each statement. For sentence selection, they use features such as *sticky bigrams*, emotion lexicons [PFB01], and *kill verbs*. They also ask the workers for their political identification and find that conservative workers are more likely to label a statement as biased.

Wagner et al.[WGJS15] use lexical bias, i.e. vocabulary that is typically used to describe women and men, as one dimension among other dimensions to analyze gender bias on Wikipedia.

Recasens et al.[RDNMJ13] tackle a language bias problem that is similar to our problem. Given a sentence with known bias they try to identify the most biased word using a machine learning approach based on logistic regression and mostly language features, i.e. word lists containing hedges, factive verbs, assertive verbs, implicative verbs, report verbs, entailments, and subjectives. They also use part of speech and a bias lexicon with words that they extracted by comparing the before and after form of Wikipedia articles for revisions that contain a mention of POV in their revision comment. The bias lexicon contains 654 words including many words that do not directly introduce bias, such as *america*, *person*, and *historical*. In comparison the approach for extracting bias words presented in this chapter differs strongly from their approach and our bias lexicon is more comprehensive including almost 10,000 words. Recasens et al. report accuracies of 0.34 for finding the most biased word and 0.59 for having the most biased word among the top 3 words. They also use crowdsourcing to create a baseline for the given problem. The results show that the task of identifying a bias word in a given sentence is not trivial for human annotators. The human annotators achieve an accuracy of 30%.

Another important topic in the context of Wikipedia is vandalism detection [PSG08]. While vandalism detection uses some methods that are also relevant for bias detection (e.g. blacklisting), it is important to notice that bias detection and vandalism detection are two different problems. Vandalism refers to cases where editors deliberately lower the quality of an article and are typically more obvious. In the case of bias, editors might not be aware

that their contribution violates the NPOV.

3.2 Language Bias Detection Approach

In this section we introduce our approach for language bias detection. Our approach consists of two main steps: (i) first, we construct a lexicon of bias words in Section 3.2.1, and (ii) second, in Section 3.2.2, based on the bias word lexicon, and other features which analyze statements at the syntactic and semantic level, we train a supervised model that determines if a statement is either *biased* or *unbiased*.

3.2.1 Bias Word Lexicon Construction

In the first step of our approach, we describe the process of constructing automatically a lexicon of bias words. Bias words vary across topics and language genres, and as such generating automatically such lexicons is not trivial. However, for a set of already known words that might stir controversy or are known to be inflammatory, recent advances in word representations like word2vec are quite efficient in revealing words that are similar or used in similar context for a given textual corpora.

The process of constructing a biased word lexicon consists of two steps: (i) seed word extraction, and (ii) bias word lexicon construction.

Seed Words. To construct a high quality bias word lexicon for a domain (e.g. *politics*), an important aspect is to find a set of seed words from which we can expand in the corresponding word vector space and extract words that indicate bias. In this step, where minimal manual efforts are required, the idea is to use word vectors from words that are expected to have a high density of bias words in their neighborhood. In this way, we identify seed words in an efficient manner.

Therefore, we use a corpus where we expect a higher density of bias words than in Wikipedia. Conservapedia[5] is a Wiki shaped according to right-conservative ideas, including strong criticism and attacks especially on liberal politics and members of the Democratic Party of the United States. Since no public dataset is available, we crawl all Conservapedia articles under the category *"Politics"* (and all its subcategories). The dataset comprises a total of 11,793 articles[6]. Finally, we compute word representations using the word2vec approach.

To expand the seed word list and thus have high quality bias word lexicon, we use a small set of seed words that are associated with a strong political ideology between left and right in the US (e.g. media, immigrants, abortion). For each word, we manually go through

[5] www.conservapedia.com

[6] We preprocess the data using Wiki Markup Cleaner. We also replace all numbers with their respective written out words, remove all punctuation and replace capital letters with small letters.

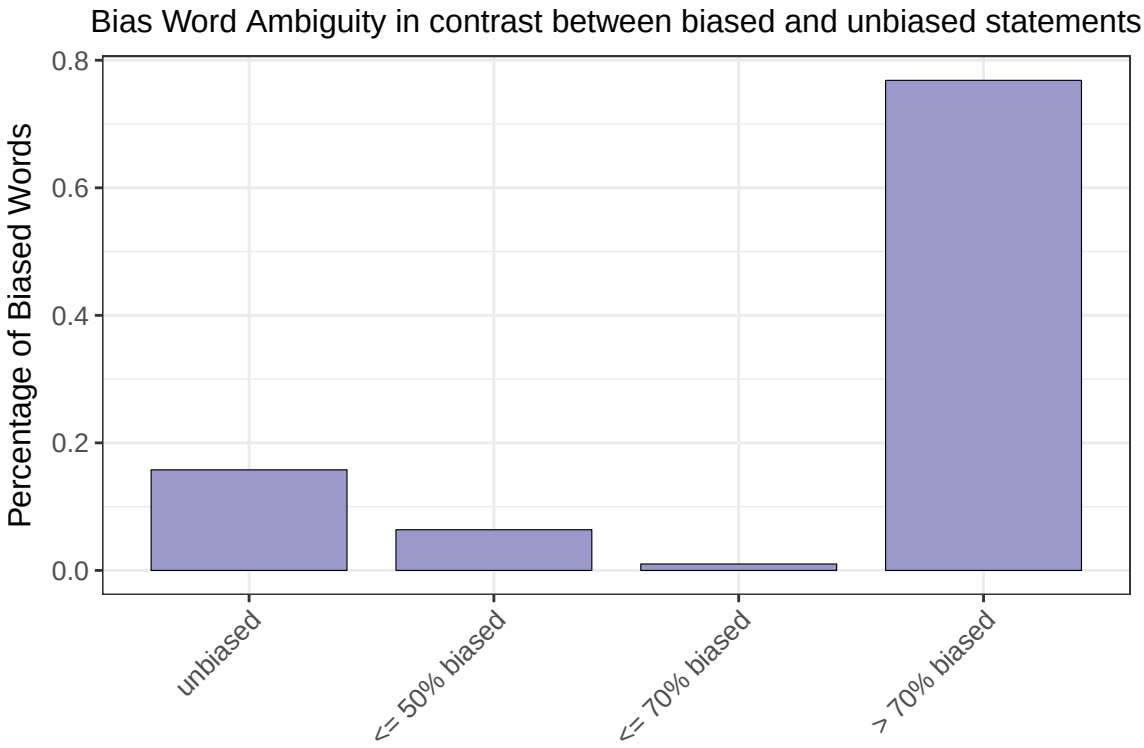


Figure 3.1. Bias word ambiguity in terms of their occurrence in *biased* and *unbiased* statements. The x-axis represents the bias words grouped by their ratio of occurrence in *biased* statements, where a 70% occurrence translates into 30% of occurrences in *unbiased* statements. While many bias words occur predominantly in biased statements, there exist also many unbiased statements containing bias words in the dataset. Consequentially, a bias word is no guarantee for a statement to be biased.

the list of closest words in their word representation and extract words that seem to convey a strong opinion. For example, among the top–100 closest words for the word *media* are words such as *arrogance*, *whining*, *despises* and *blatant*. We merge all extracted words to one list. The final seed list contains 100 bias words.

Bias Word Extraction Given the list of seed words, we extract a larger number of bias words using the Wikipedia dataset of latest articles[7], from which we compute word embeddings using word2vec with the *skip-gram* model. In the next step we exploit the semantic relationships of word vectors to automatically extract bias words given the seed words and a measure of distance between word vectors. Mikolov et al.[MSC+13a] showed that within the word2vec vector space similar words are grouped close to each other because they often appear in similar context. A trivial approach would be to simply extract the closest words

[7] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

Table 3.1. Top 20 closest words for the single seed word *indoctrinate* and the batch containing the seed words: *indoctrinate, resentment, defying, irreligious, renounce, slurs, ridiculing, disgust, annoyance, misguided*

Rank	Single seed word	Batch of seed words
1	cajole	hypocritical
2	emigrates	indifference
3	ingratiate	ardently
4	endear	professing
5	abscond	homophobic
6	americanize	mocking
7	reenlist	complacent
8	overawe	recant
9	disobey	hatred
10	reconnoiter	vilify
11	outmaneuver	scorn
12	helmswoman	downplaying
13	outflank	discrediting
14	renditioned	demeaning
15	redeploy	prejudices
16	seregil	humiliate
17	unnerve	determinedly
18	titzikan	frustration
19	unbeknown	ridicule
20	terrorise	disrespect

for every seed word. In this case, if the seed word is a bias word, we would presumably retrieve bias words but also words that are related to the given seed word but are not bias words. For example for the seed word *"charismatic"* we find the word *"preacher"* among the closest words in the vector space.

To improve the extraction, we make use of another property of word2vec. Instead of extracting the closest words of only one word, we compute the mean of multiple seed words in the vector space and extract the closest words for the resulting vector. This helps us to identify clusters of bias words.

Table 3.1 shows an example of the top 20 closest words for the single seed word *indoctrinate* and a batch containing *indoctrinate* and 9 other seed words. Our observations suggest that the use of batches of seed words leads to bias lexicons of higher quality.

We split the seed word list randomly into $n = 10$ batches of equal size. For each batch of seed words we compute the mean of the word vectors of all words in the batch. Next, we extract the top 1000 closest words according to the *cosine similarity* of the combined vector. We use the extracted bias words as new seed words to extract more bias words using

Table 3.2. Statistics about the extracted bias word lexicon

nouns	4101	(42%)
verbs	2376	(24%)
adjectives	2172	(22%)
adverbs	997	(10%)
others	96	(1%)
total	9742	

the same procedure (only one iteration). Afterwards we remove any duplicates. Table 3.2 shows statistics for our extracted bias lexicon. The lexicon contains 9742 words with 42% of them tagged as nouns, 24% tagged as verbs, 22% tagged as adjectives and 10% tagged as adverbs. The high number of nouns is not surprising since nouns are the most common part of speech in the English language.

3.2.2 Detecting Biased Statements

While the bias word lexicon is extracted from bias prone seed words and their respective words that are close in the word representations, as such they serve only as a weak proxy for flagging biased statements. Figure 3.1 shows the occurrence of bias words from our lexicon in *biased* and *unbiased* statements in our crowdsourced dataset. We will explain the crowdsourcing process in Section 3.3.1. Nearly 20% of the bias words do not appear in biased statements, and a similar ratio appears in both biased and unbiased statements. Such statistics reveal the need for more robust features that encode the syntactic and semantic representation of the statement they appear in. Listing 3.1 shows an example of a bias word from our lexicon appearing in a biased and non-biased statement.

Listing 3.1 Bias word ambiguity for word "decried"

```
The idea was roundly decried as illegal and by evangelical
Protestants, had missionaries to the tribes, and by Whigs.

Coburn exercised a hold on the legislation in both March
and November 2008, and decried the required $10 million
for surveying and mapping as wasteful.
```

Table 3.3 shows the complete list of features which we use to train a supervised model for detecting biased statements. In the following, we describe the individual features and the intuition behind using them for our task.

Bias Word Ratio. In this feature we consider the percentage of words in a statement that are part of the bias word lexicon. Considering the individual words as features would lead to a sparse feature representation, which poses a risk on overfitting in our classification task. Thus, the ratio serves as an indicator on how likely a statement is to be biased. The

Table 3.3. The complete set of features used in our approach for detecting biased statements.

Feature	Value	Description/Example
Bias word ratio	percentage	Percentage of words from bias lexicon.
Bias word context	tokens	Words adjacent to bias words
POS tag unigram/bigram distribution	percentage	e.g. ⟨ JJ NNS⟩
Sentiment	{neutral, negative, positive}	Sentiment value as labelled by Stanford's CoreNLP toolkit.
Report verb	boolean	e.g. *add*
Implicative verb	boolean	e.g. *manage*
Assertive verb	boolean	e.g. *claim*
Factive verb	boolean	e.g. *reveal*
Positive word	boolean	e.g. *great*
Negative word	boolean	e.g. *terrible*
Weak subjective word	boolean	e.g. *noisy*
Strong subjective word	boolean	e.g. *absolute*
Hedge word	boolean	e.g. *possibly*
Baseline word context	tokens	The adjacent words w.r.t to words from the *epistemological* and *framing* bias lexicons.
LIWC Features	percentage	LIWC features based on psychological and psychometric analysis.

higher the ratio the more likely it is that the statement is biased. However, as shown in Figure 3.1, bias words serve only as a weak proxy for detecting biased statements, and as such their use in isolation can lead to false positives.

Bias Word Context. For statements that are biased, a common pattern is the particular use of bias words in their context. Context is a key factor in this case in distinguishing unbiased from biased statements containing bias words. Therefore, we consider as a feature the context in which a bias word appears, thus, for each bias word occurrence, we consider the words in a window consisting of the previous and next word. Additionally, we extract the Part-of-Speech (POS) tag of the previous and next word, adjacent to the bias word in a statement. The features in this case are similar to extracting tri-grams, however, with the restriction that one of the words is present in our bias word lexicon. Additionally, in this group, we include the distance between different bias words in a statement.

LIWC Features. Linguistic inquiry word count [PFB01] is a common tool on analyzing text that contains subjective content. Through the use of specific linguistic cues, it identifies psychological and psychometric clues such as the ratio of *anger*, *sad*, and *social* words. Furthermore, the difference between the *style* and *content* words can reveal interesting insights. For instance, the use of *auxiliary verbs* can reveal that the statement might contain emotional words. Auxiliary verbs are part of what is considered to be *function words*. Other psychological indicators that can be extracted from function words are cues such as the *politeness* or *formality* in language. These are all interesting in our case as they go against the NPOV policies in Wikipedia. We consider all feature categories from LIWC and for a detailed explanation of all categories we refer to the original paper [PFB01].

POS Tag Distribution. We consider the distribution of POS tags and sequences of adjacent POS tags (e.g $\langle$NN, NNP$\rangle$) in a statement. The intuition here is that we can harness syntactic regularities that may appear in biased and unbiased statements. The features correspond to the ratio of respective POS tags, or bigrams of POS tags in a statement.

Baseline Features. As baseline features we consider the features introduced in a slightly similar task by Recasens et al. [RDNMJ13]. The features are geared towards detecting biased words in a statement and consider two main language biases, i.e. (i) epistemological and (ii) framing bias. In the first case, the bias arises by tweaking specific words and words of a specific POS tag, such that the believability of a statement is changed. For instance, the use of *subjective words*, *implicative verbs*, *hedges* can change the believability, i.e., phrasing an opinion as a fact, or vice-versa. For the second case of *framing bias*, there is a tendency on using slant words. Similarly as in the case of *bias word context*, here too, as we aim at detecting whether a statement is biased or not, the context in which these words appear is crucial, therefore we consider the pre/next word and their corresponding POS tags as features.

Additional details are reported in Table 3.3, where we indicate the values that are assigned for specific features.

3.3 Evaluation

In this section, we explain our evaluation setting. First, we describe how we construct the ground-truth through crowdsourcing and discuss its limitations. Second, we show the evaluation results of our approach and its effectiveness against competitors. Finally, we show the evaluation results on a random sample of Wikipedia statements and the results therein.

3.3.1 Crowdsourced Ground-Truth Construction

To validate our approach on detecting biased statements in Wikipedia, we needed to construct a ground-truth dataset which exhibits similar characteristics. To the best of our knowledge, there is no such ground-truth, which we can use in our evaluation setting.

We construct our ground-truth from statements extracted from the Conservapedia dataset, which we describe in Section 3.2. The reasons why we use Conservapedia instead of Wikipedia, are the two fold: (i) Conservapedia has similar text genre, and covers similar articles as Wikipedia, and (ii) the expected amount of biased statements is much higher than in Wikipedia. With respect to (ii) this has practical implications. The amount of false positives (i.e., unbiased statements) from Wikipedia would be too high for an assessment in a crowdsourcing environment, which would be costly in terms of money and time.

We construct our ground-truth through crowdsourcing. We select 70 randomly chosen articles from the category *Democratic Party*, which refers to the *Democratic Party of the United States*, and 30 articles from the category *Republican Party*, which refers to the *Republican Party of the United States*. From the resulting set of articles, we split their content into statements, where a statement consists of a single sentence. From the corresponding set of statements, we randomly sample 1000 statements for assessment through crowdsourcing.

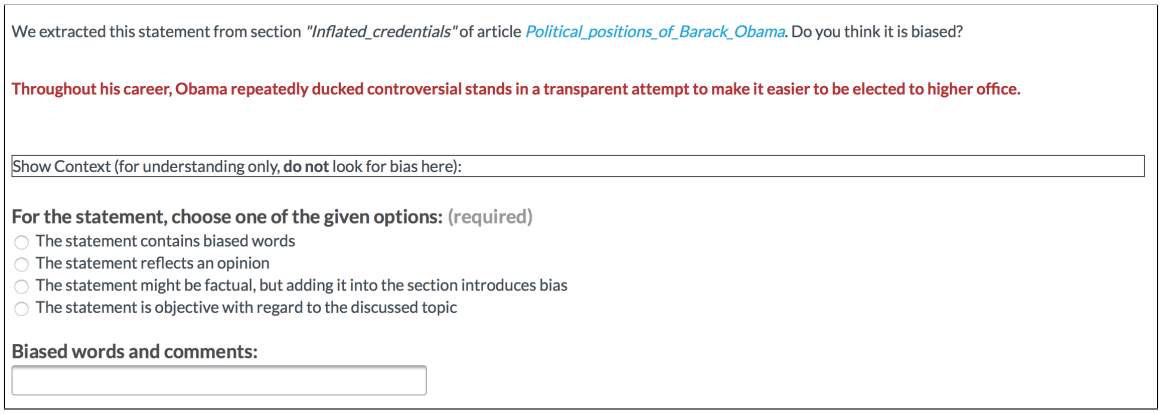


Figure 3.2. Crowdsourcing job setup for evaluating statements whether they are biased or unbiased.

Figure 3.2 shows the crowdsourcing task preview, which we host in the CrowdFlower

platform[8]. For each statement, we ask the crowdworkers to assess if the statement is biased by additionally providing the section in which the statement occurs as contextual information, so that they can make a better and more informed judgement. The options allow the workers to choose the specific type of bias, for instance, *"Opinion"* or *"Bias words"*, or *"No bias"*. The crowdworkers can choose one of the following options:

a) *Bias words* - The statement contains bias words.

b) *Opinion* - The statement reflects an opinion.

c) *Other bias* - The statement might be factual, but adding it into the section introduces bias.

d) *No bias* - The statement is objective with regard to the discussed topic.

Workers were allowed to choose only one option. In cases where both options (a) and (b) applied, we asked the workers to choose option (a). Apart from the options, we provided an optional field, where the workers could indicate the bias words, which they identified in the statement.

To account for the quality of the provided judgements by the crowdworkers, we set in place unambiguous test questions, which we use to filter out crowdworkers that do not pass 50% of them. Furthermore, we restrict to crowdworkers of *level 2* (as provided by CrowdFlower, a workforce with high accuracy on previous tasks).

Finally, for each statement we collect 3 judgements, and for each judgement we pay $2 US cents, and in case the crowdworkers provide us with the bias words in the optional field, we pay an additional $3 US cents. This results in a total of 358 contributors, with 239 passing our quality control tests. For each statement, we measure the inter-rater agreement, where we convert the judgement into a binary class of *biased* (with all its sub-classes) and *unbiased*. The resulting agreement rate as measured by Fleiss' Kappa is $\kappa = 0.35$. Due to the subjectivity of the task, we find this value to be acceptable.

From the resulting ground-truth, we decided to exclude statements that were classified as *other bias*. This class is more related to *gatekeeping* and *coverage* bias than to language bias. We also excluded statements that were classified as *opinion* since opinion detection is a different field of research. The removed classes will be helpful for future work where we plan to determine the type of bias for a statement. Furthermore, we remove statements whose judgements have a confidence score less than 0.6 as provided by CrowdFlower, which is based on the workers' agreement and the number of test questions that each worker passed.

Table 3.4 shows statistics about the final ground-truth. It contains a total of 685 statements with 323 being classified as *biased* and 362 as *not biased*.

[8] https://crowdflower.com

Table 3.4. Ground-truth statistics from the crowdsourcing evaluation, before and after filtering.

Statements Total	1000
Bias Words	383
Opinion	105
Other Bias	82
No Bias	430
Statements (after filtering)	685
Biased	323
Not Biased	362

3.3.2 Detecting biased Statements Evaluation

In this section, we provide the evaluation results for detecting biased statements. First, we provide the evaluation results in our crowdsourced ground-truth described in the previous section, and then analyze the performance of our classifier in the setting of Wikipedia.

Learning Setup. We train a classifier based on the feature set in Table 3.3. We use a RandomForest classifier as implemented in [PVG+11]. To avoid overfitting and have better generalizable models, we perform a feature ranking and choose the top-100 most important features based on the χ^2 feature selection algorithm. However, the most informative features in our case are related to the ratio of biased words in a statement and their context, LIWC features, and the context in which the words (specifically the words from the lexicons in Table 3.3) encoding framing and epistemological bias appear [RDNMJ13]. We will refer to our algorithm as **DBWS**.

We evaluate our classifier based on the crowdsourced ground-truth (see Section 3.3.1), and perform a 5-fold cross validation approach. The distribution of *biased* and *unbiased* statements is nearly evenly distributed, with 47% being biased, and 53% unbiased.

Baselines. We compare our approach against two baselines.

(**B1**) The first baseline is a simple sentiment classification approach. We use the sentiment classifier proposed by Rocher et al. [SPW+13]. We make a simplistic assumption that a *negative* or *positive* sentiment indicates a biased statement, while a *neutral* sentiment indicates an unbiased statement.

(**B2**) The second baselines is the bias word classifier by Recasens et al. [RDNMJ13], for which we use a logistic regression, similar to their original setting. The features of the second baseline are incorporated in our approach. A statement is marked as biased, if the classifier detects biased words in a statement.

Table 3.5. Evaluation results on the crowdsourced ground-truth. The precision, recall, and F1 scores are with respect to the *biased* class.

Approach	Accuracy	P	R	F1
DBWS	0.73 (▲12%)	0.74 (▲20%)	0.66 (▲5%)	0.69 (▲10%)
B1	0.52	0.48	0.03	0.06
B2	0.65	0.62	0.63	0.63

Performance. Table 3.5 shows the performance of the different approaches in detecting biased statements from our crowdsourced ground-truth. As expected, the first competitor **B1**, which decides if a statement is biased or not based on its sentiment performs really poor, with a performance near to random guessing. The accuracy is 52%. This shows the difficulty of the task, where the statements follow the principles of using objective language, and as such sentiment based approaches do not work.

Next, the second baseline **B2**, whose original task is to detect biased words, shows an improvement over the sentiment classifier. The improvement mostly comes from the use of specific lexicons which encode the *epistemological* and *framing* bias in statements. The accuracy is 65%, whereas in terms of precision, the second baseline has a precision score of $P = 0.62$ and similar recall score. However, as mentioned earlier, an important factor on deciding if a statement is biased lies in the combination of specific lexicons like our bias word lexicon or language bias lexicons [RDNMJ13] in combination with the context in which they occur.

Finally, our classifier **DBWS** achieves the highest accuracy, with 73%. In terms of precision in classifying biased statements, we achieve a precision score of $P = 0.74$, and recall score of $R = 0.66$. This presents a relative improvement of nearly 20% in contrast to the best performing competitor in terms of precision, whereas for recall we have a 5% improvement.

Table 3.6. Classification results on the Wikipedia statements sample.

Articles	1,000
Statements	8,302
Biased	2,988
Not Biased	5,314

Robustness – Wikipedia Evaluation. Despite the striking similarities between Conservapedia and Wikipedia in terms of textual genre and coverage of topics, there are fundamental differences in terms of quality control and bias policies.

Therefore, we perform a second evaluation on a random sample of Wikipedia articles, which are of the same categories as our crawled dataset from Conservapedia. To have

comparable articles, we look for exact matches of article names, resulting in 1713 equivalent articles. From the resulting articles, we extract their entire revision history, which results in 2.2 million revisions in total. Finally, we sample a set of 1000 revisions, from which we extract 8,302 statements (after filtering out statements shorter than 50 characters).

Next, we run our classifier **DBWS** which we trained on our crowdsourced ground-truth from Section 3.3.1. From 8,302 statements, a total of 36% are flagged as being biased by the classifier, as shown in Figure 3.6. However, since we do not have the real labels of the Wikipedia statements, we are interested in evaluating a sample of *biased* statements from Wikipedia. Thus, we take a random sample of 100 *biased* statements, whose classification confidence is above 0.8, and we manually evaluate the statements to assess whether they are biased or unbiased. After increasing the classification confidence to be above 0.8, from 36% we are left with nearly 4% of biased statements.

The resulting evaluation on the sample of biased statements from Wikipedia reveals that our classifier is able to flag accurately biased statements with a precision of $P = 66\%$. It is important to note here, that our classifier is pre-trained on the crowdsourced ground-truth, and as such the language bias signals are more stronger in that case, when compared to subtle language bias in Wikipedia. However, a 4% number of biased statements presents a major result when put into perspective of the large amount of edits that happen in Wikipedia and given that the overwhelming number of statements on Wikipedia are not biased.

The results are highly valuable and they have great implications. First, it shows that our model can generalize well over Wikipedia statements, where the language bias is far more subtle when compared to Conservapedia. Second, our crowdsourced ground-truth, despite the fact that we generate it from an encyclopedia known to have high bias and slant towards specific ideologies, due to its comparably similar content it allows us to devise bias word lexicons which can be applied efficiently in more neutral context like Wikipedia.

3.4 Conclusion and Future Work

In this work, we proposed a novel approach for detecting biased statements in Wikipedia. We focus on the case of language bias, for which we propose an approach to construct a bias word lexicon for a domain of interest, and furthermore together with syntactic and semantic features which we extract from the statements in which the bias words occur, we can accurately identify biased statements. We achieve reasonable precision with $P = 0.74$, which presents a relative improvement of 20% over word-level approaches that detect biased words in statements.

Furthermore, we provide a new ground-truth dataset of biased and unbiased statements, which can be used for further improving research in detecting language bias. Finally, we show that our approach, trained in a more explicitly biased content like Conservapedia, can generalize well over Wikipedia, which is known to be of higher quality and where the language biases are more subtle. On a small evaluation over Wikipedia statements, we achieve a precision of $P = 66\%$ using our pre-trained classifier in the constructed

ground-truth from Conservapedia.

As future work, we plan to further improve the classification results. A promising direction is to consider information about a Wikipedia article coming from the talk pages, where revisions and information to be added is discussed by Wikipedia editors. This in addition can serve as a means for constructing NPOV datasets based on distant supervision approaches. Furthermore, we seek to refine the granularity of our classifiers into detecting the different language biases as shown in our crowdsourcing evaluation.

4 Neural Based Statement Classification for Biased Language

In sociolinguistic theory, language and its linguistic structures are seen as a medium that is in the function of specific *social groups* [Hal70]. That is, language and its use reflects the demands and other characteristics of the group (i.e., ideology, economical, cultural). This usually results in a consensus amongst a group in the vocabulary use and the meaning of specific phrases and words on specific topics. Due to the diversity in stances and points of view for different topics, such a consensus cannot always be achieved. This is often the case when written discourse (or any language utterance) is considered to be biased. Bias in language is manifested in different forms, from discrimination in terms of *gender* through over-lexicalization (e.g. *female doctor* vs. *doctor* for male) [R+00], or in terms of *authority* on how one addresses a person (e.g. *nominal* reference through *title* + *name* vs. name) [BG+60, Fow13]. Other forms of bias tackle the believability of a statement or introduce terms that are considered to be one-sided in topics that do not have a consensus amongst the different societal groups [RDNMJ13].

Wikipedia is a unique environment in manifesting such diversity in terms of points of view and stances for a large variety of topics. The current English version of Wikipedia consists of more than 5 million articles of highly diverse topics, which are constructed from a large editor base of more than 32 million editors[1]. Given its scale and diversity it is not surprising that many statements in Wikipedia reflect biases from its underlying editors, respectively their societal background. Statements on issues that are controversial cause disagreements between editors, specifically the different points of view in a discussion. Other factors are diffused from external sources like news, the second most cited external resource [FAA15, FMNA16]. Fowler [Fow13] shows that news are prone to a range of issues such as language bias.

To avoid such cases of language bias and other biases that arise in controversial topics, Wikipedia has established a set of principles and guidelines. For instance, the *neutral point of*

[1] https://en.wikipedia.org/wiki/Wikipedia:Statistics

view (NPOV) defines criteria that should be followed by its editors: (i) avoid stating opinions as facts, (ii) avoid stating seriously contested assertions as facts, (iii) avoid stating facts as opinions, (iv) prefer nonjudgemental language, and (v) indicate the relative prominence of opposing views. Recent work [RDNMJ13, Mar17] shows that in Wikipedia's case[2], most NPOV violations are w.r.t biased language (i) – (iv), and often one-sided statements (v), specifically in the form of *epistemological* and *framing* bias. *Epistemological* refers to linguistic cues that have impact in the *believability* of a statement, while *framing* refers to the terms and phrases that are one-sided in the case where a topic may have multiple viewpoints.

The statements below show the diverse forms of bias that are present in Wikipedia.

(a) Andrew James Breitbart was one of the ***most outspoken***, ***fearless*** conservative journalists in America.

(b) The Labour Party in the United Kingdom put together a ***highly successful*** set of policies based on encouraging the market economy, while promoting the involvement of private industry in delivering public services.

(c) An abortion is the ***murder*** of a human baby embryo or fetus from the uterus, resulting in or caused by its death.

(d) Sanders ***shocked*** his fellow liberals by putting up a Soviet Union flag in his Senate office.

(e) This may be a result of the fact that the public had ***unsurprisingly*** lost support for the President and his policies.

(f) The Blair government had ***promised*** a referendum on whether Britain should sign the Constitution, but ***refused*** popular demands that it carry out its promise.

The examples above show different forms of biased language. The cases in (a) – (b) represent framing bias and are manifested in the form of adjectives like *highly successful* or *fearless* as subject intensifiers. The remaining cases represent epistemological bias, e.g., *shocked* states a very strong precondition of the truth of the proposition (i.e., *"shocking his fellow liberals"*), similar is (f).

In this work, for the mentioned aspects, we focus on detecting biased language in Wikipedia statements that are introduced either due to inflammatory wording or phrases and whether a statement is written in a neutral tone, thus, following the principles of the NPOV policy. In Chapter 3, we addressed the problem of detecting biased language, using feature-based models to capture the different forms of bias in Wikipedia statements based on specific lexicons and hand-crafted features. However, such approaches fail to capture the inter-dependency of words that may incur bias and their context, and furthermore, relying

[2]The author of the work in [Mar17] carried out an experimental study on his personal Wikipedia page https://en.wikipedia.org/wiki/Brian_Martin_(social_scientist)

solely on hand-crafted lexicons has its disadvantages of not being able to capture all forms of bias.

We propose an approach that is based on recurrent neural networks (RNN) with two modes of attention, *global* and *hierarchical* [BCB14, YYD+16], which achieves significant improvements over feature-based approaches [RDNMJ13, HF18].

To this end, we provide the following contributions:

- a neural model for detecting biased language,
- largest corpus of biased statements

4.1 Related Work

In this section, we review related work, which covers several aspects of *biased language* and other forms of linguistic manifestation of biases, such as *framing analysis* or *gender biases*. In terms of corpora, most of related work is focused on Wikipedia, news media, and other political corpora like political debates. In the following, we categorize the related work based on their objective.

Article Bias. The seminal work by Greenstein and Zhu [GZ12b] is the first to analyze bias in Wikipedia. They adapt an approach initially developed for determining newspaper slant [GS10]. The approach relies on a list of 1000 terms and phrases typically used by either republican or democratic congress members. To measure political bias in Wikipedia, Greenstein and Zhu look for occurrences of these terms and phrases in Wikipedia articles about US politics. For example, if an article contains significantly more terms typically used by democratic congress members compared to terms typically used by republican congress members, then this is an indicator for a pro-democratic leaning of the article's content. According to their findings, Wikipedia articles are on average more left-leaning, especially in the early phase of Wikipedia. With more editors working on an article, the bias decreases on average. But since most articles do not receive much attention, there is still a significant number of articles containing bias.

In their work, Greenstein and Zhu [GZ12b] focus on the topic of *US politics*. Therefore, the framing bias that they detect has a narrow scope, whereas our work is different in the sense that we aim at capturing a broader scope of biased language. We classify statements that contain biased language which is introduced through words or phrases that are partial or are not neutrally phrased.

Biased Language. Recasens et al. [RDNMJ13] propose an approach for detecting a single bias-inducing word given a biased Wikipedia statement. The approach relies on linguistic features, divided into two bias classes: framing bias, including subjective language such as praising and perspective-specific words; and epistemological bias, dealing with believability of a proposition, i.e. phrasing choices that either cast doubt on a fact or try to sell an opinion as a fact. In their dataset collection, they crawl Wikipedia revisions that

have a "*POV*" flag in the revision comments. We use a similar dataset collection procedure, however, we additionally use crowdsourcing to filter statements that do not contain bias ($>$ 60% for our data sample). Given that their approach is originally intended to identify words that introduce bias, we adopt their approach and consider the proposed features in [RDNMJ13] to classify statements as either containing bias or not as one of our baselines.

Additionally, we compare with our own work as presented in Chapter 3. In this chapter, we will use a slightly different approach for data collection and annotation, and a novel neural-based approach for classification. We will show that long-range dependencies between words and phrases in a statement are hard to capture through hand-crafted features and that a context-aware model achieves significant improvement over a purely feature-based approach.

Ideological Bias and Framing Analysis. Iyyer et al. [IEBGR14] introduce a RNN model for classifying statements as either liberal or conservative. Their datasets contain statements from politicians in US Congressional floor debates and statements from ideological books about US politics. For pre-selecting biased statements from the data they make use of the features used by Yano et al. [YRS10] and a simple classifier with manually selected partisan unigrams as features. For labeling the pre-selected statements they use crowdsourcing, where crowdworkers label not only the full statement but also each phrase part of the sentence separately in a hiearchical manner. These additional labels allow for a handling of semantic compositions and the correct classification of more complex sentence structures, when the sentence parts are incrementally combined. For example, the statement *They dubbed it the "death tax" and created a big lie about its adverse effects on small businesses.* is classified as liberal bias, even though the term "death tax" suggests pro-conservative bias.

Lahoti et al.[LGG18] propose an unsupervised approach for determining the ideology of both users and content in a combined liberal-conservative latent space using Twitter data. They include features such as the surrounding network structure of users and information about content shared by users.

Baumer et al. [BEQ+15] propose a model to detect the linguistic cues that introduce framing in political events. The results suggest that readership is often unaware of the subtle framing cue words, and that depending on the framing of an event the perception and stances towards an event may vary. The classifier relies on a set of syntactic and lexical features for identifying framing cue words. Similar is the work by Tsur et al. [TCL15], where they propose a topic modeling approach to identify farming words in news articles.

Our work is not comparable to the above works. The works in *ideological* bias can be seen as a case of framing bias, whereas in the case of *framing analysis*, the problem is even more subtle than framing bias. Framing usually represents the interplay between the cognitive bias and the context in which a statement is positioned. As such, the scope of these works cannot capture all the possible cases that we tackle and that introduce biased language.

Other Bias. Some research also covers other types of bias, e.g. selection bias [BRA18],

[Mar17] and bias focusing on specific topics, such as gender bias [WGJS15] or cultural bias [CH11]. We do not consider open opinions to be bias. For example, the statement *I think this movie is really bad* is not bias according to our definition because the writer makes clear that it is her own opinion.

4.2 Data Collection

In this section, we introduce our approach on collecting statements from Wikipedia articles that contain biased phrasing. The data collection consists of two main procedures: (i) a pre-selection of statements from Wikipedia revisions that contain a *POV* tag in the comments, and (ii) a crowdsourcing step which we use to manually annotate statements containing phrasing bias. Below we describe in detail the individual steps.

4.2.1 Extracting POV-tagged Statements from Wikipedia

Wikipedia editors are encouraged to add comments when changing or adding content in a Wikipedia article. In some cases editors add comments to mark that their change aims at reducing bias and thus restoring the *Neutral Point of View*.

We extract all statements from the entire revision history of the English Wikipedia, for those revisions that contain the *POV* tag in the comments. This leaves us with 1,226,959 revisions. We compare each revision with the previous revision of the same article and filter revisions where only a single statement has been modified[3]. The reason for this is that if multiple statements have been modified, we are unable to say if the POV tag in the revision comment refers to all statements or only to a fraction. The final resulting dataset leaves us with 280,538 *pov-tagged* statements.

Table 4.1 shows the number of different edit types. In 129,578 cases the statement has been deleted in the new revision. In 601 cases the statement has been moved to a different section.

In another 150,359 cases, the statement has been updated in the new revision[4]. The low number of moved statements is not surprising, since moving a statement to another section does usually not mitigate its bias.

4.2.2 Crowdsourced Ground-Truth Construction

Wikipedia is a highly dynamic platform. Its user base is very large, and with it there is a high diversity in the expertise, that is, understanding the NPOV principle of Wikipedia, or simply there may be different stances towards an added statement in a Wikipedia page representing

[3] With modified we understand any statement that has been *updated/deleted/moved.*

[4] We use assume that a statement has been updated if there is another statement that is similar to the previous statement with a high jaccard similarity of 0.7.

Table 4.1. Statements from revisions with *POV* comments across the different modification types they undergo.

deleted	*moved*	*updated*
129,578	601	150,359

some form of event. We notice several additional types of biases that cause disagreement between the different Wikipedia editors as indicated by their revision comments:

- **Selection Bias:** *"NPOV; the CS Monitor accusations are not relevant here"*
- **Focus Bias:** *"Actually, this info is already in the criticisms section. While I agree it is needed in the article multiple mentions is POV pushing."*

In other cases editors use the POV-tag to discuss the article's assumed bias:

- *"can someone explain to me what is POV about this article?"*

Even in cases where the editor explicitly tags the statement as containing (phrasing) bias, this still reflects the opinion of only one editor. Other editors might disagree.

Crowdsourced Ground-truth. To tackle these issues, we ask workers to identify statements containing phrasing bias in the Figure Eight platform[5]. Since labeling the full *pov-tagged* dataset would be too expensive, we take a random sample of 5000 statement from the dataset. Figure 4.1 shows a preview of the job, where we show a single statement to the workers and let them label each statement, providing three options:

- *"The wording is neutral."*
- *"The wording is biased. I can think of a more neutral wording."*
- *"I don't know."*

Workers were allowed to choose only one option. Note that we are not just asking the workers to label statements according to whether they contain (phrasing) bias or not, since this would be a more ambiguous and subjective task. Instead we ask workers to consider the statement as a fact and to choose the *"biased"* option only if they can think of a more neutral wording to present this fact. This way we make sure that the workers focus on the phrasing of the statement and not on it's content.

[5] https://www.figure-eight.com/

To improve quality of the judgments, we provide a number of examples and restrict workers to level 2 or higher[6]. Additionally, we set in place unambiguous test questions and filter out workers who do not pass at least 70% of the questions.

For each judgment we pay ¢1.6 US cents. For each statement we collect 3 judgments leading to a total of 15,000 judgments. We measure worker agreement using Krippendorffs Alpha, a measure of rater agreement for sparse cases where not every rater rates every item. The agreement is low ($\alpha = 0.124$) as expected given the subjectivity of the task.

We filter out all statements labeled as *"I don't know"* and all statements with confidence < 0.6. The final dataset contains 4952 labeled statements with 1843 ($\sim 37\%$) of them labeled as biased and 3109 ($\sim 62\%$) labeled as neutral. The large percentage of statements not labeled as biased confirms that the crowdsourcing step is necessary to identify the statements containing phrasing bias. Simply assuming that a POV-tagged statement contains phrasing bias would result in a larger but also low quality dataset.

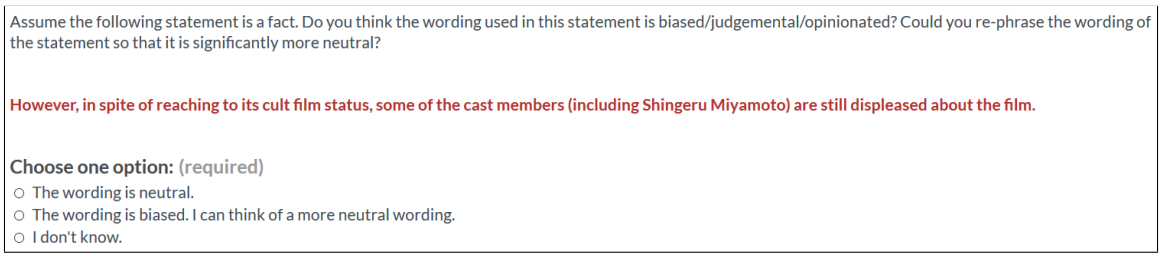


Figure 4.1. Crowdsourcing job setup for annotating sentences as *"biased"* or *"neutral"*.

4.3 Biased Language Classification

In this section, we present our approach for classifying biased language in Wikipedia statements. We overcome some of the major drawbacks of our model presented in Chapter 3, which relies on hand-crafted features and specific lexicons, and thus, is limited in capturing the varying manifestations of bias in language.

As the following examples show, the mere presence of words cannot considered to be a reliable indicator of bias. The first case shows a biased statements, whereas the second refers to an objective legal term. In addition, in other cases the bias can be introduced through phrases or multiple words appearing in different locations in a sentence (cf. third example below.).

- An abortion is the *murder* of a human baby embryo or fetus from the uterus, resulting in or caused by its death.
- In 2008 he was convicted of *murder*.

[6]Figure Eight divides workers into 3 levels with increasing competence.

- *The public agrees* that it is the *number one* country in the world.

We remedy all of the above issues of existing work and propose two sequence based classifiers that rely on Recurrent Neural Networks (RNNs) with gated recurrent units (GRU) [CVMG+14] for computing the hidden representation of sequences in a sentence. Additionally, we will make heavy use of attention mechanisms [BCB14, YYD+16] to determine words in a sentence that are indicators of biased language. We first describe the means with which we represent statements, then describe the necessary details of RNN, and finally explain in detail the two proposed models for biased language classification.

4.3.1 Statement Representation

An important prerequisite in successfully applying RNN models in our task, is the representation of words in a sentence. We distinguish three main sentence representations.

Word Representation. We represent a sentence $s = (w_1, \ldots, w_n)$ consisting from a sequence of words through their corresponding word representations. We will use the GloVe embeddings [PSM14] to represent the words in our corpus. Unknown words we will initialize randomly in our word embedding matrix W_{glove}.

Word embeddings have been successfully applied in downstream tasks in NLP, and are shown to be efficient in capturing context and synonymous words.

POS Tags. POS tags are one of the most basic features used to represent text, and are able to capture stylistic linguistic features. POS tag are successfully employed in determining *text genre* [Bib91]. Similarly, POS tags have shown to provide insights in determining biased statements in Chapter 3.

We additionally represent each token in s through its POS tag. In our RNN models, we compute the POS tag embedding matrix W_{POS}, and use it in combination with W_{glove}.

LIWC Word Functions. LIWC text analysis [PFB01] has been successfully employed in a number of tasks that capture subjectivity of text, such as analyzing language in *fake news* [RCJ+17], and additionally as shown in Chapter 3, LIWC features, when used together with the context of the n–grams, prove to provide a high improvement over existing approaches [RDNMJ13] in detecting biased statements.

Similarly as for POS tags, here too we train our embedding matrix W_{LIWC} and use it in combination with other token representations. LIWC categorizes words into 75 different categories, each representing the function of a word, e.g. whether a word represents negative emotion. Since a word may be in function of different LIWC categories, we chose the most *descriptive LIWC category* for a word[7]. In general, LIWC categories express a range of psychological and sociological functions of words, and thus, are highly important for subjective tasks like detecting statements with biased language.

[7]We compute an IDF measure on the word - LIWC function association, thus, we prefer LIWC functions that are less likely to be assigned to other words.

4.3.2 RNN Statement Encoding

For a given Wikipedia statement which we represent as a sequence of words $s = (w_1, \ldots, w_n)$, RNNs encode the individual words into a hidden state $h_t = f(w_t, h_{t-1})$. The function f in our case can be represented either through an LSTM or GRU function[8].

The encoding of an input sequence from s is dependent on the previous hidden state. This dependency based on f determines how much information from the previous hidden state is passed onto h_t. For instance, in case of GRUs, h_t is encoded as following:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{4.1}$$

where, the function z_t and $\tilde{h}_t$ are computed as following:

$$z_t = \sigma(W_z w_t + U_z h_{t-1} + b_z) \tag{4.2}$$

$$\tilde{h}_t = \tanh\left(W_h w_t + r_t \odot (U_h h_{t-1} + b_h)\right) \tag{4.3}$$

$$r_t = \sigma(W_r w_t + U_r h_{t-1} + b_r) \tag{4.4}$$

The function z_t decides the amount of information that is kept from h_{t-1}, which is extracted from step $t-1$ and thus impacts the computation of h_t, whereas r_t is known as the *reset gate* that can disallow the past information from h_{t-1} to be included in $\tilde{h}_t$, which consequentially impacts the computation of the state h_t. This particular property of RNN encoders is highly important for our task, as the presence of words or phrases in statements with biased language can be easily encoded through the hidden states h_t. Furthermore, sequences which do not contribute in improving the classification accuracy are captured through the model parameters in function r_t, allowing for the model to ignore information coming from such sequences.

4.3.3 RNN – Global Attention

One disadvantage of plain RNN models is that when used for classification tasks or language generation (standard encoder-decoder cases) tasks, the classification is done based on the last hidden state h_N. In the case of long sentences, this can be problematic as the hidden states, respectively the weights from the different input sequences have to be correctly represented in the last state.

Attention mechanisms [BCB14] have proven to be successful in circumventing this problem. The main application of attention mechanism has been within machine translation tasks [BCB14, LPM15]. The main difference between standard training of RNN models is that all the hidden states are taken into account to derive a *context vector*, where different states contribute with varying weights, or known with *attention weights* in generating such a vector. The context vector depends on the task. For instance, in machine translation it is used to decode the input sequence into another sequence.

[8] A detailed description of LSTMs and GRUs is beyond the scope of this work, we refer to the respective papers for more details [HS97, CVMG+14].

In our case, as shown in Figure 4.2, we employ the attention mechanism to compute a sentence representation s_{rep} and use it to classify the statement s. This has the advantage that our sentence representation consists only of the hidden states which are important in determining the class of s. More formally, we compute s_{rep} as following:

$$u_t = \tanh\left(W_{emb} h_t + b_{emb}\right) \tag{4.5}$$

$$\alpha_t = \frac{\exp(u_t^T c)}{\sum_{t'} \exp(u_{t'}^T c)} \tag{4.6}$$

$$s_{rep} = \sum_t \alpha_t h_t \tag{4.7}$$

We see from Eq (7) that s_{rep} is the sum of the hidden states of s weighted according to the importance of each sequence α_t, where α_t simply represents a *softmax* function over the hidden representation of words as computed in u_t and the context vector c.

Finally, to account for different representations of s, we capture aspects such as the stylistic and LIWC features (see Section 4.3.2). We consider different combinations in our experimental setup, i.e., words + POS, words + LIWC, and words + POS + LIWC. We *concatenate* the different sequence representations (see *merge* layer in Figure 4.2), and pass them onto the GRU cells for learning the hidden representations h_t.

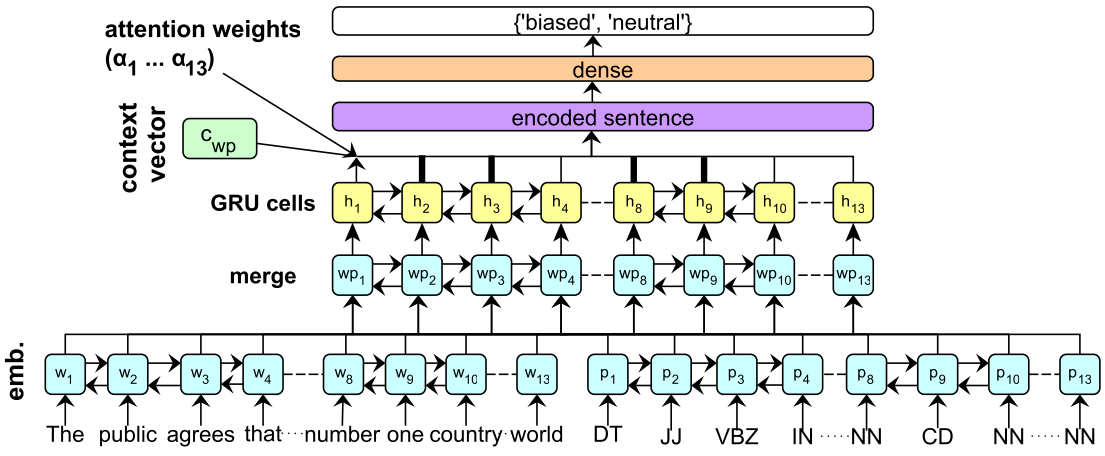


Figure 4.2. We combine the different sentence representations by concatenating them. We compute a sentence representation based on an *attention-mechanism*, which weighs the input sequences and thus generates the sentence representation based on their importance in the classification task.

4.3.4 RNN – Hierarchical Attention

Hierarchical attention, introduced in [YYD+16], is employed in the case of document classification. It first applies the attention mechanism on top of sentences, respectively at the

word level. The computed word attention is used to represent a sentence, similar as in s_{rep} for which they compute the hidden representations through GRU cells. Finally on top of the hidden representation of individual sentences is applied the attention mechanism, thus, resulting in a final document representation, which is used for classification.

Here, we employ a similar strategy, in that we have a fixed set of sentence representations (see Section 4.3.2), which we feed as separate sentences into the hierarchical attention mechanism, and thus, are able to learn separately the importance of the different representations in determining if a sentence has biased language or not. Figure 4.3 shows an overview of the proposed model. The computation of the overall sentence representation is similar to that in Eq (4.7). The only difference here lies in the fact that instead of merging the different sentence representations, we compute individually the importance of each representation.

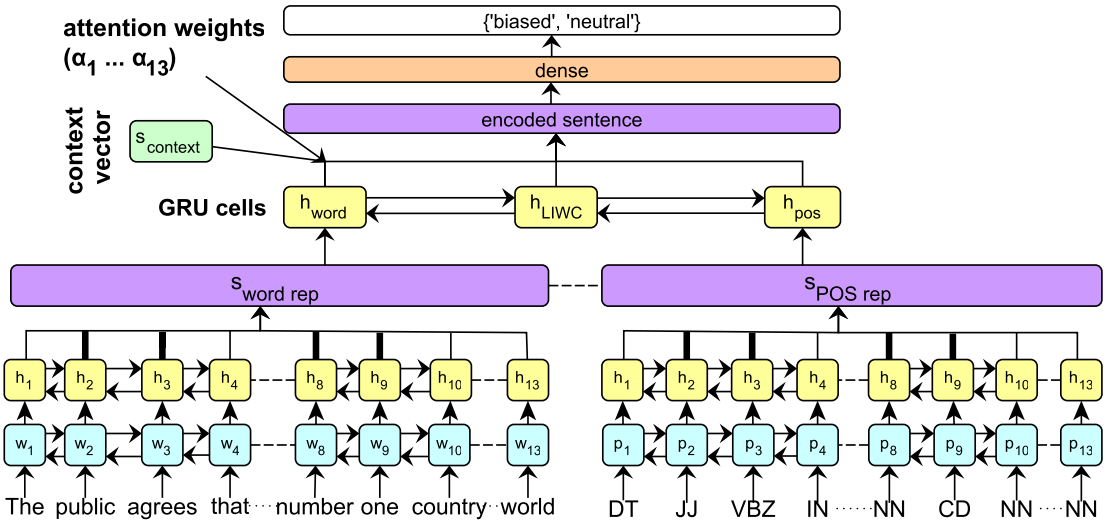


Figure 4.3. We compute separately the attention weights and the corresponding sentence representations similar to Eq (4.7). We pass the computed sentence representations into GRU cells, thus, computing their hidden representations, from which we compute another joint representation based on the attention weights of the separate sentences, and finally classify into *"biased"* or *"unbiased"* using a *sigmoid* function.

4.4 Experimental Setup

In this section, we describe the experimental setup for detecting statements that contain biased phrasing. We first describe the different strategies on generating datasets with *unbiased* statements (apart from the ones gathered through crowdsourcing) and further describe the competitors and the learning setup of our approach.

4.4.1 Datasets

The statements that were marked as *"neutral"* by crowdworkers in our data collection in Section 4.2 represent statements that contain other forms of biases or subjectivity as explained earlier (i.e., selection, focus biases etc.). As such these statements do not represent the ideal high quality content in Wikipedia. For this reason, we will denote the crowdsourced neutral statements as the *hard* case of distinguishing between biased and neutral statements.

To obtain a cleaner labeled dataset containing both statements with and without biased phrasing, we additionally extract statements from featured Wikipedia articles which arguably contain mainly statements without biased phrasing due to their high quality. In the following, we describe all different datasets that we use for evaluating our approach.

CW-Hard: This dataset consists of only the crowdsourced statements that we described in Section 4.2. The dataset consists of 1843 statements marked as *"biased"* and 3109 marked as *"neutral"*. As we will see later in the evaluation section, this dataset proves to be the hardest as the *"neutral"* statements contain quality issues that can be attributed to other forms of bias or subjectivity factors.

Featured: To extract *"neutral"* statements of high quality, we turn back to statements extracted from featured articles in Wikipedia[9]. Featured articles are more likely to be neutral when compared to statements from random articles of varying quality. The findings are consistent with [GZ12b], where articles with a large number of revisions from a diverse pool of editors are less likely to contain bias.

The English Wikipedia contains 5338 articles that are featured articles. We crawl the content of featured articles[10] and extract more than 1 million statements, from which we sample the equivalent amount of statements (1.843 statements) as for the *"biased"* class in our data collection step in Section 4.2. Finally, the *"biased"* statements in the ***featured*** dataset are the same as in ***cw-hard***, with the only difference in the *"neutral"* statements.

Type–Balanced: Statements we extract in Section 4.2 are from a wide range of types of articles. Depending on their type (i.e. the Wikipedia categories an article belongs to or the type from a reference knowledge base), the statements therein will vary in their language genre and linguistic structure due to the difference in focus. For instance, articles about location vary substantially from articles about persons in their genre and topical focus in the respective articles.

Table 4.2 shows the top–10 types for the ***cw-hard*** dataset and the ***featured*** articles datasets. The type distributions in both datasets are different. While the ***cw-hard*** dataset

[9]Featured articles are considered to be articles of high quality conforming to the various editing policies in Wikipedia, such as: neutrality, statements that are verifiable through citations and additionally with highly reputable citations.

[10]Time of access: June 14th 2018

contains a larger number of articles belonging to the types `Place` and `Populated Place`, the ***featured*** dataset contains mostly articles belonging to types like `Software`, `VideoGame`, and `MusicalWork`.

To account for such divergence in statement distribution in the ***featured*** dataset, we enforce that statements should be from featured articles and additionally have a similar type distribution as the ***cw-hard*** dataset. As we will show in the evaluation results later on, statements differ significantly across types in their thematic aspects and in some cases in language genre. Again, we take a random sample of 1.843 statements, similar to the amount of *"biased"* statements as in the ***cw-hard*** dataset.

Table 4.2. Top 10 Wikipedia article types from DBpedia for *type–balanced* featured articles, *cw-hard*, and *featured* articles.

Type–Balanced	CW-Hard	Featured
Agent	Place	Work
Work	PopulatedPlace	Agent
Place	Agent	Software
Person	Settlement	VideoGame
PopulatedPlace	Organisation	Organisation
Organisation	Work	MusicalWork
Settlement	Country	Film
Species	Person	Album
Eukaryote	Company	Place
WrittenWork	City	Person

4.4.2 Baselines

We compare our approach against two existing baselines, which focus on the same task as ours. The approaches rely on hand-crafted features to detect biased language in Wikipedia statements. Additionally, we consider as baselines vanilla RNNs without attention for varying sentence representations.

- **B1:** The first baseline is an adoption of the approach in [RDNMJ13]. Originally the approach detects words that introduce biased statements. We adopt it such that instead of classifying individual words, we classify statements as either biased or not. The feature space is the same as in the original paper in [RDNMJ13].
- **B2:** Our approach presented in Chapter 3, which extends over **B1** by further introducing contextual features by means of *n-grams* and other features that analyze statements for psychological and sociological insights through the LIWC [PFB01] text analysis tool.

- **RNN:** We consider as a baseline vanilla RNNs, where we compute the hidden representation of sequences with GRUs [CVMG+14] with dimensions $h_t \in \mathbb{R}^{100}$. We consider different combinations of sentence representations: (i) **RNN**^{w}, (ii) **RNN**^{wp}, (iii) **RNN**^{wl}, and (iv) **RNN**^{wpl}, where w, p, l, correspond to the word embedding [PSM14], POS tag, and LIWC sequence representations (100 dimensions), respectively[11]. We train the model for 10 *epochs* with a *batch size* of 100, and use *Adam* for optimizing our *binary crossentropy* loss function. We use 70% of data for training, 10% for validation, and the remaining 20% for testing.

We also used a sentiment classifier for the problem of detecting statements with biased phrasing, but the performance was too low to serve as a solid baseline. This confirms that bias detection, as a problem, differs strongly from the problem of sentiment analysis.

4.4.3 Approach Learning Setup

Here, we describe the learning setup of our two approaches: (i) RNN with attention **RNN**$_a$ and (ii) RNN with hierarchical attention **RNN**$_h$. Similar as for the simple **RNN** baseline, we consider variations of sentence representations (see Section 4.3.2). For all representations, we consider an embedding space of 100 dimensions, that is, $W_{emb} \in \mathbb{R}^{k \times 100}$, where k is the number of entries in the respective representation space.

We use Keras with Tensorflow as a backend. We again train for 10 epochs with batch size of 100 and use 70% of data for training, 10% for validation, and the remaining 20% for testing. We minimize the *binary crossentropy* loss w.r.t the *accuracy* metric.

We consider the following configurations for our approaches:

- **RNN**$_a$: To represent the sequences in terms of POS tags and the word function based on LIWC, we need to train the corresponding embeddings, in which case, we consider three scenarios: (i) train separately the embedding weights, (ii) share the weights amongst POS tag and LIWC representations of sentences, and (iii) share the weights amongst all three sentence representations.
- **RNN**$_h$: In the case of the hierarchical attention, we represent a sentence in either 2 dimensions through its word and (POS or LIWC representation), or through all its three representations. In terms of embeddings, we consider pre-trained word embeddings [PSM14] or train word embeddings jointly with POS and LIWC representations together.

4.5 Evaluation Results

In this section, we present the evaluation results and a detailed discussion. We focus on two main aspects: (i) performance in predicting if a statement contains biased language, and (ii)

[11] When a sentence is represented through more than one sequence representation, we *merge* the sequence representations in their respective embedding spaces.

robustness, where we consider a real-world scenario of predicting if statements in revisions in a Wikipedia article contain biased language.

Table 4.3. Evaluation results for all competing approaches. We show the results for all three different datasets. The evaluation metrics (P/R/F1) are shown for the *"biased"* class. The best scores for each metric and dataset are marked in bold.

	type-balanced				***featured***				***cw-hard***				***average***			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	$\overline{Acc}$	*MAP*	$\overline{R}$	$\overline{F1}$
B1	0.666	0.669	0.657	0.663	0.646	0.650	0.632	0.641	0.622	0.626	0.606	0.616	0.645	0.648	0.632	0.640
B2	0.707	0.705	0.710	0.708	0.702	0.703	0.700	0.700	0.641	0.640	0.645	0.643	0.683	0.683	0.685	0.684
RNN^{w}	0.786	0.805	0.738	0.770	0.776	0.788	0.780	0.784	0.653	0.668	0.668	0.668	0.738	0.754	0.729	0.741
RNN^{wp}	0.802	**0.839**	0.722	0.776	0.789	0.843	0.717	0.775	0.653	0.709	0.524	0.602	0.748	0.797	0.654	0.718
RNN^{wl}	0.779	0.716	**0.869**	0.785	0.794	0.851	0.717	0.778	0.651	0.650	0.715	0.681	0.741	0.739	0.767	0.748
RNN^{wpl}	0.773	0.770	0.762	0.766	0.771	0.803	0.738	0.769	0.648	0.670	0.639	0.654	0.731	0.748	0.713	0.730
RNN$^{w}_{a}$	0.783	0.784	0.767	0.776	0.795	0.866	0.691	0.769	0.686	0.699	0.699	0.699	0.755	0.783	0.719	0.748
RNN$^{wp}_{a}$	0.803	0.801	0.794	0.797	0.818	0.892	0.715	0.794	0.681	**0.712**	0.647	0.678	0.767	0.802	0.719	0.756
RNN$^{wl}_{a}$	**0.808**	0.814	0.786	**0.800**	0.800	0.809	**0.809**	0.809	0.688	0.697	0.712	**0.705**	0.765	0.773	**0.769**	**0.771**
RNN$^{wpl}_{a}$	0.796	0.820	0.741	0.778	0.801	0.860	0.723	0.785	**0.691**	0.710	0.691	0.700	0.763	0.797	0.718	0.754
RNN$^{wp}_{h}$	0.796	0.832	0.714	0.768	0.803	0.899	0.649	0.754	0.664	0.689	0.644	0.666	0.754	**0.807**	0.669	0.729
RNN$^{wl}_{h}$	0.785	0.809	0.725	0.764	**0.819**	**0.917**	0.668	0.773	0.672	0.665	**0.743**	0.702	0.759	0.797	0.712	0.746
RNN$^{wpl}_{h}$	0.807	0.837	0.741	0.786	0.812	0.872	0.733	0.797	0.679	0.696	0.683	0.690	**0.766**	0.802	0.719	0.758

4.5.1 Biased Language Detection Performance

Table 4.3 shows the evaluation results for all competitors and the different configurations of our approach in classifying statements if they contain biased language. The results are shown for the three different datasets, which vary only in terms of *"neutral"* statements, specifically how we sample for such statements (see Section 4.4.1).

Feature-Based. We see that feature based algorithms like the baselines in **B1** and **B2** are outperformed by all RNN based approaches. This confirms our hypothesis that biased language is often introduced through multiple words or phrases that are hard to capture through word lexicons [RDNMJ13]. We notice that n-gram features in **B2** provide a relative improvement of 6.8% in terms of F1 score for the *type-balanced* dataset. Similar improvements are observed for the other datasets. It is worth noting that in the case of the ***cw-hard*** dataset, the performance is significantly lower when compared to the other two datasets, with a relative decrease of 10% in terms of F1 score for the ***type-balanced*** dataset. This is attributed to the difficulty in distinguishing between *"biased"* and *"neutral"* statements in ***cw-hard***, since neutral statements in this case contain other forms of bias such as selection, focus bias etc.

RNN baselines. Our main claim in this work was that language bias in statements is hard to capture through n-gram based features and that RNN based models can better capture the inter-dependencies between words and phrases that introduce bias. Table 4.3 confirms this claim. If we consider only the RNN baselines with GRU cells [CVMG+14], the best configuration is when representing the sentence as a combination of its words and the LIWC function of a word, specifically through the concatenated embeddings of both

representations. RNN^{wl} achieves a relative improvement of 11% in terms of F1 score for the ***type-balanced*** dataset. Similar improvements are observed in the other two remaining datasets. In terms of precision the improvement can go well beyond 19%, whereas in terms of recall we see an improvement of 22%. This shows the ability of RNN based approaches to encode sequences in a statement such that only sequences which help in the classification task, respectively their information from the hidden states, are passed onto the sentence encoding (Eq (1) – (4)), thus, making the classification much more accurate.

Attention-based RNN. The attention mechanism allows us to capture the importance of specific input sequences from a sentence for the classification task. We employ two modes of attention. First, the global attention RNN$_a$ that operates on top of the merged sentence representation. Second, a hierarchical attention RNN$_h$, which is first applied on the separate sentence representations, whereby we construct an intermediate sentence representation based on the most important input sequences, and on top of which we apply another layer of attention, and finally classify the sentence.

We note that RNNs with hierarchical attention achieve the best performance amongst all approaches, with $P = 0.917$ in the setting of RNN$_h^{wl}$, whereas RNN$_a^{wp}$ achieves $P = 0.892$. This presents an improvement of over 30% in terms of precision over the feature-based model **B2** and 5% improvement over the best performing RNN baseline. In terms of F1 score, RNN$_a$ achieves the best performance, due to higher coverage of *"biased"* statements.

A direct comparison between the two modes of attention reveals that the performance is quite close. Hierarchical attention achieves overall better precision, however, at the cost of recall. Interestingly, we see that in all cases there is a gain in representing statements through the word, POS and LIWC word function representations. This shows that context (through word embeddings) and in combination with the linguistic style that is captured through POS tags and additionally the LIWC word functions can yield significant improvement over simplistic word representations.

Over all datasets, we see that in terms of accuracy and precision RNN$_h$ performs best, whereas in terms of F1 score RNN$_a$ shows the best performance. In the next task, where we assess the robustness of our model, we pick RNN$_a^{wl}$ as it is most stable in terms of F1 across all datasets.

Table 4.4. Robustness results for our best performing approach and the impact of its training on the different datasets.

	Acc	P	R	F1
type-balanced	0.638	0.609	0.757	0.675
featured	0.678	0.654	0.757	0.702
cw-hard	0.645	0.640	0.686	0.662

4.5.2 Robustness

For a large variety of tasks, an important concern is how well do trained models on controlled settings perform in real-world scenarios? To this end, we assess the robustness of our approach by considering statements coming from the *controversial*[12] Wikipedia article about `Abortion`[13]. This article serves only to demonstrate how well our best performing model RNN_a^{wl}, pre-trained on the previous three datasets, would perform in correctly classifying statements in this article that contain biased language.

From the entire revision history of the `Abortion` article, we extract revisions that contain *POV* quality tags, and thus, extract all statements that have been deleted or modified. There are different reasons why editors delete or modify statements, as indicated by editor comments. Examples apart from POV issues are statements considered to be *irrelevant* or *unimportant*, statements that are *not supported by a source*, or *vandalism*. This resulted in 10,243 statements, from which we sample 100 and annotate them through crowdsourcing, similar as in Section 4.2. The annotated dataset contains 52 statements labeled as biased and 48 statements labeled as neutral. The high number of statements labeled as biased is not surprising given the controversial topic of the article.

Table 4.4 shows the performance of the best performing model RNN_a^{wl} pre-trained on the datasets in Section 4.4.1, and evaluated on the *robustness data*. The performance of the model trained on the ***type-balanced*** and the ***featured*** datasets is stable with F1 scores of 67.5% and 70.2%.

Similarly, as in Table 4.3, we see a lower performance in terms of F1 score for the model trained on the ***cw-hard*** dataset. Table 4.4 shows that the classifiers are *robust* and *generalize* well over instances that are very different from their original train set. Additionally, this shows that even if we employ our approach in a real-world scenario to flag highly voluminous and unclean statements, we can detect with reasonably good performance statements that contain biased language.

4.6 Conclusion and Future Work

In this chapter, we presented an RNN based approach for classifying statements that contain biased language. We focused on the case of biased phrasing, that is, statements in which words or phrases are *inflammatory* or *partial*. We showed that RNN models are superior in performance when compared to feature-based models and are able to capture the important words and phrases that introduce bias in a statement. Furthermore, we show that encoding the statements based on different representations such as words, POS, and LIWC word functions, through which we capture *context*, *style*, and *psychological* and *sociological* functions of words, we can predict with high accuracy statements that contain biased language.

Finally, we show that with employing attention mechanisms (both global and hierar-

[12] https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues
[13] https://en.wikipedia.org/wiki/Abortion

chical) we can further improve the performance of our approach, by identifying salient sequences and additionally providing means of interpreting and uncovering different forms of biased language. We are able to predict with a very high precision of up to 91.7%, thus, providing a highly significant relative improvement over competitors with more than 30% in terms of precision.

As future work, we foresee analyzing the different forms of bias such as selection bias and bias introduced due to the demographics of the underlying editor population in Wikipedia.

5 Understanding and Mitigating Crowd Worker Biases

Microtask crowdsourcing provides remarkable opportunities to acquire human input at scale for a variety of purposes [KNB+13] including the creation of ground-truth data and the evaluation of systems. A survey of crowdsourcing tasks on Amazon's Mechanical Turk [DCD+15] revealed that one of the most popular tasks is that of *interpretation and analysis* (IA) [GKD14]. In many scenarios, such interpretation tasks may be prone to biases of workers. These biases are subject to various factors, such as cultural background of workers, personal opinion on a topic, ideological, or other group memberships of a person.

Such factors are well studied from the language point of view and the use of language to express statements on a given subject at hand [Lyo70, R+00, Bro60, Fow13]. For instance, sociolinguistic studies show *gender biases* in English language in terms of authority (e.g. how a person is addressed, *title* + *firstname* + *lastname*) [Bro60, Lyo70] or in terms of over-lexicalization [R+00] (e.g. *young married woman*). Language bias and bias in language use occurs in various contexts, e.g. journalism [FDNML16]. Subjective language [WWB+04] can be seen as a subproblem of language bias (e.g. *framing, opinions* etc.), which often is presented through subtle linguistic cues that carry an implicit sentiment [GR09, SF88] and often are deliberately used in order to convey a specific stance towards a subject. Thus, differentiating between *neutrally* phrased and *opinionated* statements is subject to the worker's ideological memberships.

Studies [Ben16, BB04] show that the political or ideological stance of a person can influence the perception and interpretation of facts. In interpretation tasks such as distinguishing between opinions and facts, worker awareness of possible biases that may be introduced due to their personal or ideological stances is crucial in providing noise free judgments. For example, surveys[1] show that only 23% of the U.S population who identify politically with the Republican party believe that humans have an influence in climate change.

Several natural language understanding tasks that rely on crowdsourced labeling are

[1] http://www.people-press.org/2007/01/24/global-warming-a-divide-on-causes-and-solutions

prone to worker biases. For instance, Yano et al. [YRS10] showed that in determining biased language in text corresponding to the news genre, a pivotal quality concern is the actual political stances of the workers. Here, the perceived bias of labelers was found to vary depending on their political stance. Other examples of ground-truth acquisition through crowdsourcing where worker biases may lead to subjective judgments include *opinion detection*, *sentiment analysis* etc. In general, the ability to mitigate biased judgments from workers is crucial in reducing noisy labels and creating higher quality data. To this end, we address the following research questions:

RQ#1: How does a worker's personal opinion influence their performance on tasks including a subjective component?
RQ#2: How can worker bias stemming from strong personal opinions be mitigated within subjective tasks?
RQ#3: How does a worker's experience influence their capability to distance themselves from their opinion?

Based on the aforementioned observations in prior works that suggest an influence of personal stances in subjective labeling tasks, we construct the following hypotheses:

H#1: Workers are more likely to make a misclassification if such a classification is in line with their personal opinion.

H#2: Experienced workers are relatively less susceptible to exhibiting bias.

The main contributions of our work in this chapter are:

- A novel measure for worker bias in subjective tasks based on misclassification rates and workers' opinions.
- Novel techniques for mitigating systemic worker bias stemming from personal opinions.
- Revealing the impact of such systemic worker bias on aggregated ground-truth labels.

5.1 Related Literature

5.1.1 Bias in Crowdsourcing Data Acquisition

Recent works have explored task related factors such as complexity and clarity that can influence and arguably bias the nature of task-related outcomes [GYB17]. Work environments (i.e., the hardware and software affordances at the disposal of workers) have also shown to influence and bias task related outcomes such as completion time and work quality [GCGD17]. Eickhoff studied the prevalence of cognitive biases (ambiguity effect, anchoring, bandwagon and decoy effect) as a source of noise in crowdsourced data curation, annotation and evaluation [Eic18]. Gadiraju et al. showed that some crowd workers exhibit inflated self-assessments due to a cognitive bias [GFK+17]. Crowdsourcing tasks are often susceptible to participation biases. This can be further exacerbated by incentive schemes [EdV13]. Other demographic attributes can also become a source of biased judgments.

It has also been found that American and Indian workers differed in their perceptions of non-monetary benefits of participation. Indian workers valued self-improvement benefits, whereas American workers valued emotional benefits [JWN15]. Newell and Ruths showed that intertask effects could be a source of systematic bias in crowdsourced tasks [NR16]. Other works revealed a significant impact of task order on task outcomes [AG18, CIT16]. Zhuang and Young [ZY15] explore the impact of *in-batch* annotation bias, where items in a batch influence the labelling outcome of other items within the batch.

These prior works have explored biases from various standpoints; task framing and design, demographic attributes, platforms for participation and so forth. In contrast, we aim to analyze and mitigate the bias in subjective labeling tasks stemming from personal opinions of workers using the example task of *bias detection*.

5.1.2 Subjective Annotations through Crowdsourcing

For many tasks such as detecting subjective statements in text (i.e., text pieces reflecting opinions), or biased and framing issues that are often encountered in political discourse [Sch99, Fow13], the quality of the ground-truth is crucial.

Yano et al.[YRS10] showed the impact of crowd worker biases in annotating statements (without their context) where the labels corresponded to the political biases, e.g. *very liberal*, *very conservative*, *no bias*, etc. Their study shows that crowd workers who identify themselves as *moderates* perceive less bias, whereas conservatives perceive more bias in both ends of the spectrum (*very liberal* and *very conservative*). In a similar study, Iyyer et al. [IEBGR14] showed the impact of the workers in annotating statements with their corresponding political ideology. In nearly 30% of the cases, it was found that workers annotate statements with the presence of a bias, however, without necessarily being clear in the political leaning (e.g. liberal or conservative). While it is difficult to understand the exact factors that influence workers in such cases, possible reasons may be their lack of domain knowledge, i.e., with respect to the stances with which different political ideologies are represented on a given topic, or it may be due to the political leanings of the workers themselves. Such aspects remain largely unexplored and given their prevalence they represent an important family of quality control concerns in ground-truth generation through crowdsourcing.

In this work, we take a step towards addressing these unresolved quality concerns of crowdsourcing for such subjective tasks by disentangling bias induced through strong personal opinions or stances.

5.1.3 Mitigation of Bias

In large batches that consist of several similar tasks, Ipeirotis et al. showed that it is possible to use statistical methods and eliminate systematic bias [IPW10]. The authors relied on synthetic experiments to do so. In other related work, Faltings et al. propose a game theoretic

incentive scheme to counter the anchoring effect bias among workers [FJPT14]. Wauthier and Jordan [WJ11] propose a machine learning model, which accounts for bias in a labelling task, where the labels are obtained through crowdsourcing. Here, the task is to predict labels, where consensus among the labellers is missing. Our work addresses the case where complete agreement among labellers, may still lead to a biased label. We explore various approaches to mitigate such undesirable bias, stemming from personal stances of workers.

Kamar et al. introduced and evaluated probabilistic models for identifying and correcting task-dependent bias [KKH15]. Other lines of work [LPI12, KOS11], rightly assume different expertise among the crowdsourcing workers, and thus propose models that improve over the *majority voting* label aggregation scheme. Such approaches are suitable for cases where there is disagreement among the workers. However in subjective tasks, the presence of varying ideological backgrounds of workers means that it is possible to observe biased labels with complete agreement among the workers, rendering such models inapplicable.

Raykar et al. [RYZ+09] introduce an approach for combining labels provided by multiple types of annotators (experts and novices) to obtain a final high quality label. In contrast to their work, we aim to mitigate the effects of worker bias during the annotation process directly via interventions.

5.2 Method and Experimental Setup

In our study of crowd worker bias, we focus on the task of labeling biased statements, a task that has found prominence in recent times to create ground truth data and to evaluate methods for *bias detection in text*, as presented in Chapters 3 and 4. We chose this task as an experimental lens due to its inherent susceptibility to worker subjectivity. In this task, workers are presented with statements pertaining to controversial topics and asked to decide whether the statement is "neutral" or "opinionated". All statements revolve around a set of specific controversial topics wherein workers can be assumed to have diverging opinions. During the course of the task, we ask workers for their own opinion on each of the topics. Given this information, we define a measure of worker bias and investigate different approaches to mitigate potential bias.

5.2.1 Statement Extraction

We chose 5 controversial and widely discussed topics from US politics (Abortion, Feminism, Global Warming, Gun Control, and LGBT Rights) from Wikipedia's *List of controversial issues*[2]. We chose these popular and controversial topics so that a majority of crowd workers (from USA) could arguably have some basic understanding of the topic and an opinion.

For each of the chosen topics, we selected a main statement that reflects the central pro/contra aspect of the controversy, e.g. "Abortion should be legal". We extracted biased

[2] https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues

Table 5.1. The controversial topics chosen for this study together with the main statements and one example statement each from the corresponding Wikipedia articles.

Topic	Main Statement	Example Statement
Abortion	Abortion should be legal.	An abortion is the murder of a human baby embryo or fetus from the uterus.
Feminism	Women have to fight for equal rights.	Feminists impose pressure on traditional women by denigrating the role of a traditional housewife.
Global Warming	Global warming is a real problem caused by humanity.	The global warming theory is perpetuated only for financial and ideological reasons.
Gun Control	Citizens should have free access to guns.	In some countries such as the United States, gun control may be legislated at either a federal level or a local state level.
LGBT Rights	Homosexual couples should have the same rights as heterosexual couples.	There are many inspiring activists who fight for gay rights.

statements from the English Wikipedia using the approach from Chapter 4 for articles that cover the given topics, e.g. *LGBT rights by country or territory* for LGBT rights. The approach relies on "POV" tags in comments for Wikipedia article revisions, which are added by Wikipedia editors for statements violating the NPOV principle [3]. In this context Wikipedia provides an explanation of opinionated statements. By extracting statements that have been removed or modified for POV reasons, we obtained a set of biased statements for each topic.

Authors of this work acted as experts to validate that all statements in the final set contain explicit bias according to Wikipedia's definition. Where necessary, we modified the statements briefly to make them comprehensible out of context. We removed phrases that where irrelevant or confusing (for example, we removed the phrase "resulting or caused by its death" from the statement "An abortion is the murder of a human baby embryo or fetus from the uterus resulting or caused by its death.") and replaced very specific words to make the statements clearer and easier to understand (for example, we replaced "misandry" with "hate against men").

We split the resulting set of biased statements into pro statements that support the main statement for this topic and contra statements that oppose the main statement. Additionally, we extracted neutral statements from the latest versions of the articles. We followed the process of open coding to ensure that the statements were reliably identified as pro, contra and neutral [Str87]. We iteratively coded the resulting statements as either 'pro', 'contra', or 'neutral' until unanimous agreement was reached on each statement, thereby forming the ground truth for our experimental tasks.

[3] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

5.2.2 Crowdsourcing Task Design

We manually selected 6 of the extracted statements for each topic; 2 pro, 2 contra, and 2 neutral statements. Our final statement set contains 30 extracted statements and 5 main statements. Table 5.1 shows the main statements and an extracted example statement for each topic.

Workers were asked to label each of the 30 extracted statements as either "neutral" or "opinionated". We also provided a third option, "I don't know", which workers were encouraged to select in case they were not sure (see Figure 5.1).

Figure 5.1. Example statement labeling task corresponding to the topic of '*Feminism*'.

We also gathered each worker's opinion corresponding to each topic from the statement group. We presented the main statement for each topic, and gathered responses from workers on a 5-point Likert scale ranging from *1: Strongly Disagree* to *5: Strongly Agree* (see Figure 5.2).

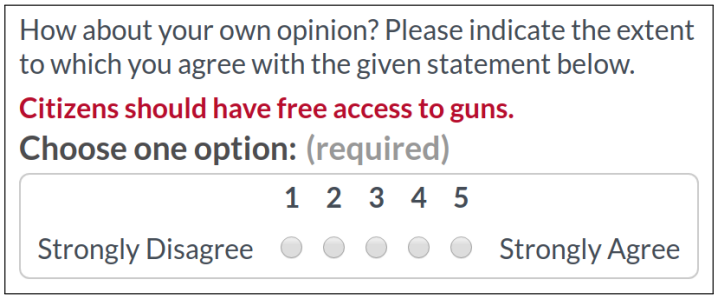


Figure 5.2. Example main statement to gather workers stances on the topic of '*Gun Control*'.

5.2.3 Study Design

In our study, we analyze worker behavior under different conditions with the goal of mitigating worker bias. We consider the following variations.

Standard Bias Labeling Task (Baseline) In this condition, we consider the standard bias labeling task as introduced in Section 5.2.2, without an explicit method or attempt for bias mitigation. Although it is now common practice to deploy crowdsourcing jobs with quality control mechanisms embedded in them [KNB+13, GKDD15], it is still uncommon to control for biases stemming from worker opinions. Thus, we consider the more typical setting which is devoid of any form of bias control as a baseline condition for further comparisons.

Social Projection (SoPro) Two popular methods to induce honest reporting in the absence of a ground-truth are the Bayesian truth serum method (BTS) [Pre04] and the peer-prediction method [MRZ05]. In a related study, Shaw et al. found that when workers think about the responses that other workers give then they work more objectively [SHC11]. We draw inspiration from such truth-inducing methods as well as from the theory of social projection [Hol68, Hol78] and aim to analyze the effect of social projection on mitigating biases stemming from worker opinions. In this condition workers are asked to label statements according to how they believe the majority of other workers would label them. We modified the task title and descriptions to adequately describe this condition. Apart from these minor changes, the task was identical to the baseline condition.

Awareness Reminder (AwaRe) Recent work has reflected on the importance of creating an awareness of existing biases in order to alleviate the biases [BY18]. We aim to analyze the impact of creating an awareness of biases stemming from personal opinions among workers, on their capability of being objective. In this condition we encouraged workers to reflect on the controversial nature of the topics in the task, and the potential bias that could be induced by their personal opinions on their judgments. We explore whether workers who are explicitly made aware of the subjective component in the task, go on to be more careful while making judgments.

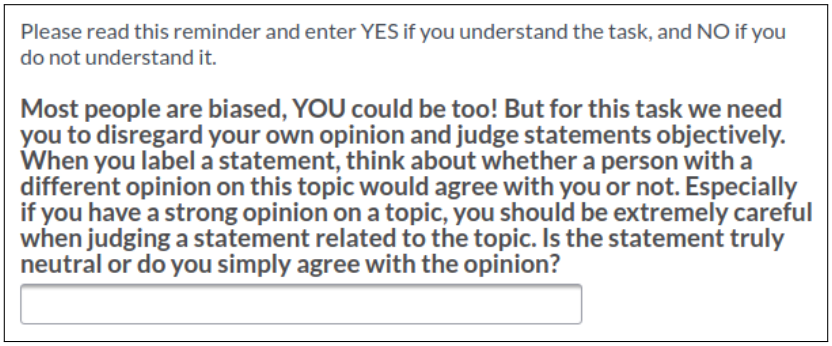


Figure 5.3. Message snippets serve as reminders to create awareness of potential biases in the *AwaRe* condition.

We appended a message in the task description to create awareness among workers and presented 6 reminders at random intervals within the task bearing the identical message. To

ensure that workers read or acknowledge the reminders, we created an interaction where workers were asked to type "YES" if they understood what was expected of them and "NO" otherwise. Figure 5.3 depicts the message snippet that serves as a reminder.

Personalized Nudges (PerNu) Similar to the *AwaRe* condition, here we investigate whether workers can deliberately influence the results of the task by distancing themselves from their personal opinions. In this condition, we first gather responses from workers on the main statements pertaining to each of the topics as shown in Figure 5.2. Using this knowledge of worker stances on a given topic (gathered on a 5-point Likert scale), we present personalized instructions to workers alongside each statement that is to be labeled. For example, if a worker strongly agrees with a main statement that '*Citizens should have free access to guns*', then the worker receives a personalized instruction drawing attention to his potential bias while judging all statements related to '*Gun Control*', as shown in the Figure 5.4. Note that the personalized instructions are phrased according to the degree of agreement or disagreement of the workers with the main statement.

Figure 5.4. Example personalized instruction to workers who strongly agree that citizen should have free access to guns, on statements related to '*Gun Control*' in *PerNu* condition.

5.2.4 Experimental Setup

For each task variation we deployed a job on FigureEight[4], a primary crowdsourcing platform, and acquired responses from 120 workers. Each crowdsourcing job contained a task description including a brief explanation of "neutral" and "opinionated" statements. We also provided some labeling examples for both classes. To ensure reliability of responses, we restricted participation of workers on the platform to Level 1 or above (2, 3). FigureEight workers are awarded level badges based on their accuracy across several test questions across hundreds of tasks of different types. Level 3 workers are the workers of the highest quality, followed by Level 2 and Level 1. In a multiple choice question, workers were asked

[4] http://www.figure-eight.com/

to provide their FigureEight contributor level (1, 2 or 3). We also included two attention check questions to filter out inattentive workers [MS13]. All job units (statements to label, main statements for opinion, attention checks, and contributor level question) appear in a random order to control for ordering effects. Workers who participated in one condition were not allowed to complete tasks in any other condition to avoid potential learning effects. Workers were allowed to submit their responses only after completing the full set of units. Since the chosen topics focus on USA politics, we restricted participation on the platform to workers from the USA to avoid effects of domain knowledge. We compensated each worker at a fixed hourly rate of 7.5 USD based on our estimates of task completion time.

5.2.5 Measuring Worker Bias

For each topic, we split workers into the 5 worker categories: *strong opposer*, *opposer*, *undecided*, *supporter*, *strong supporter*.

This categorization is based on the worker opinions of the main statement corresponding to a topic (gathered on a 5-point Likert scale), with *strong opposer* referring to 'Strongly Disagree' and *strong supporter* to 'Strongly Agree'. We refer to the workers of the category *strong opposer* as *strong opposers*, and likewise for the other categories.

To measure worker bias, we focus on the misclassifications, i.e. the worker labels that do not coincide with the given ground-truth classes. We argue that incorrectly labeled statements can serve as indicators of worker bias. Given our task design, there are three different forms of misclassifications:

- A pro-statement labeled as neutral (*pro→neut*).
- A contra-statement labeled as neutral (*con→neut*).
- A neutral statement labeled as opinionated (*neut→op*).

We first compute the misclassification rates for all types of misclassifications and all worker categories. The misclassification rate for a specific misclassification type is defined as the fraction of the number of misclassifications for a statement type ("pro", "contra", or "neutral") and the number of all judgments for statements of the same type. To assure that the bias measure is robust across different task variations, we normalize the misclassification rates for each worker category by computing the z-scores of each value.

According to hypothesis **H#1**, due to the bias stemming from a worker's personal opinions a (strong) supporter of topic `t` is more likely to label a pro statement of topic `t` as neutral, while a (strong) opposer of topic `t` is more likely to label a contra statement of topic `t` as neutral.

If hypothesis **H#1** holds, then (strong) supporters should be more likely to misclassify pro statements as being neutral compared to contra statements, i.e. the *pro→neut* misclassification rate should be comparatively higher than the *con→neut* misclassification

rate. For (strong) opposers we should observe an opposing trend, where the *con→neut* misclassification rate should be higher than the *pro→neut* misclassification rate.

To test **H#1**, we define bias for a worker category as the difference between the normalized *pro→neut* and the normalized *con→neut* values for this category. The following equation presents our measure for computing worker bias:

$$\begin{aligned} bias_{w_x} &= \frac{\left(m_{pro}(w_x) - \frac{\sum_{w_i \in w} m_{pro}(w_i)}{|w|}\right)}{\sigma} \\ &\quad - \frac{\left(m_{con}(w_x) - \frac{\sum_{w_i \in w} m_{con}(w_i)}{|w|}\right)}{\sigma} \\ &= zscore(m_{pro}(w_x)) - zscore(m_{con}(w_x)) \end{aligned} \tag{5.1}$$

where $m_{pro}(w_x)$ is the *pro→neut* misclassification rate, $m_{con}(w_x)$ is the *con→neut* misclassification rate for worker category w_x, and w is the set of worker categories.

Relatively high positive values show that workers are more likely to regard a *pro* statement as neutral compared to a contra statement and therefore indicate pro bias. At the same time, relatively low negative values show that workers are more likely to regard a *contra* statement as neutral compared to a pro statement and therefore indicate contra bias. If **H#1** holds, we should observe a tendency towards pro bias for (strong) supporters and a tendency towards contra bias for (strong) opposers.

Note that a high misclassification rate alone does not necessarily indicate worker bias. It is possible that workers of a specific category generally perform badly in labeling opinionated statements. We therefore consider the misclassification rates for both *pro→neut* and *con→neut*. The *neut→op* misclassification rate has no direct relation to worker bias since we cannot attribute a pro or contra bias to it.

5.3 Results and Analysis

In this section, we present the results of our study. We analyze and compare worker performance and bias in the 4 different variations described earlier.

5.3.1 Worker Categories

For each condition, we first filtered out workers who did not pass at least one of the two attention check questions. In case of the *AwaRe* condition, we additionally filtered out workers who did not enter 'YES' in response to all the reminder snippets. This leaves us with 102 workers in the *Baseline* condition, 106 workers in the *SoPro*, 93 workers in the *AwaRe*, and 72 workers in the *PerNu* condition.

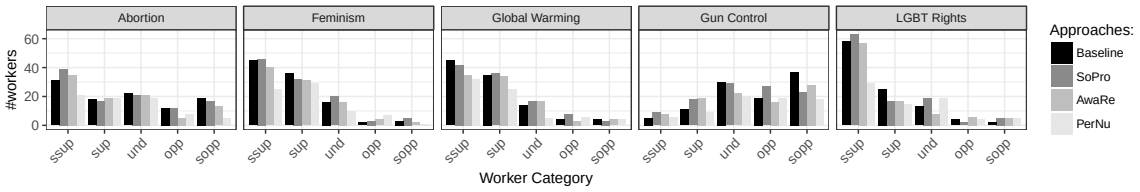


Figure 5.5. Number of workers for each topic, condition, and worker category. Worker categories: ssup = *strong supporters*, sup = *supporters*, und = *undecided*, opp = *opposers*, sopp = *strong opposers*

Figure 5.5 shows the distributions of workers across worker categories. There is a tendency towards *Strong Supporters* and *Supporters*s for all topics except Gun Control, where the tendency is more towards *Strong Opposers* and *Opposers*. This suggests that workers on FigureEight tend to be more liberal in their views on average, with Abortion, Feminism, Global Warming, and LGBT Rights being traditionally supported by liberals in the US. Our findings are inline with similar observations in [YRS10], where the task was to assess how biased a statement is w.r.t liberal vs. conservative bias.

Table 5.2. Worker performance, agreement, and average task completion time (TCT) across all conditions.

	Baseline	**SoPro**	**AwaRe**	**PerNu**
# workers	102	106	93	72
# judgments	3060	3180	2790	2160
# misclassifications	618	565	424	502
Misclassification Rate	0.20	0.18	0.15	0.23
Fleiss' Kappa	0.33	0.43	0.49	0.33
Krippendorff's α	0.34	0.44	0.38	0.33
TCT (in mins)	7.00	7.37	9.13	8.88

5.3.2 Worker Performance

Table 5.2 shows the overall results for each condition. On average, workers perform well across all conditions with average misclassification rates of 0.20 (*Baseline*), 0.18 (*SoPro*), 0.15 (*AwaRe*), and 0.23 (*PerNu*). We found a significant difference in worker performance between the conditions; *p = 2.3e-12, F(3, 10709)=19.13* using a one-way ANOVA. Post-hoc Tukey-HSD test revealed a sig. diff. between Baseline and AwaRe *(p = 0.001)* with a small effect size (Hedge's *g = 0.12*).

We measure inter-worker agreement using Fleiss' Kappa and Krippendorff's α [Kri11]. For both measures, the agreement values of all the other conditions are higher compared to the *Baseline* with the highest agreement observed in the *AwaRe* condition using Fleiss' Kappa and the *SoPro* condition according to Krippendorff's α. The generally low to moderate inter-worker agreement is consistent with expected agreement in similar tasks

Figure 5.6. Misclassification rates for all conditions, worker categories: ssup = *strong supporters*, sup = *supporters*, und = *undecided*, opp = *opposers*, sopp = *strong opposers*. Misclassification types: pro as neutral, neutral as opinionated, contra as neutral.

[HKH+14].

The average task completion time (TCT) for workers in the *Baseline* and *SoPro* is ~7 mins. In case of the *AwaRe* and the *PerNu* condition we observe higher task completion times of ~9 mins, which can partly be attributed to the additional information snippets that we confront workers with in both conditions. A one-way ANOVA showed a significant difference in TCT across interventions; $p = 0.012$, $F(3,356)=3.73$. Post-hoc Tukey-HSD test revealed a significant difference between TCT w.r.t. *Baseline* and *AwaRe* ($p = 0.025$) with a medium effect size (Hedge's $g = 0.44$).

Figure 5.6 illustrates the misclassification rates per worker category for each misclassification type and each condition. We refer to the different types of misclassifications as introduced in Section 5.2.5. Workers selected the "I don't know" option in only 2.8% of cases. We did not consider these to be misclassifications. Using Welch's T-test, we found that workers in all categories and across all conditions label a pro statement as being neutral significantly more often than they label a contra statement as neutral; $t(1424) = 18.781$, $p < .001$. We also found a large effect size; Hedge's $g = 0.70$.

The rate of neutral statements being misclassified as opinionated appears to be consistent among different worker categories in the *Baseline* condition. In the *SoPro* and *PerNu* conditions (strong) supporters exhibit a lower misclassification rate for (*neut→op*), while in the *AwaRe* condition (strong) opposers exhibit a lower misclassification rate. Across all conditions we did not find correlation between worker categories and the accuracy of judging neutral statements.

5.3.3 Worker Bias

We measure worker bias as the difference between the normalized misclassification rates for pro and contra statements. The normalized misclassification rates and the resulting bias for all worker categories are presented in Table 5.3. High positive values indicate pro bias and low negative values indicate contra bias. A bias value close to 0 indicates that the group of workers is not biased to either the pro or contra side. For the sake of convenience while

making comparisons across conditions, we use the notion of *total bias*, which is the sum of the absolute bias values in each condition.

Table 5.3. Normalized misclassification rates (z-score) and worker bias for all worker categories and all conditions. For the bias column, positive values indicate pro bias, negative values indicate contra bias. Total bias is the sum of the absolute bias values. The introduced mitigation approaches achieve lower (total) bias values compared to the baseline.

	Baseline			**SoPro**			**AwaRe**			**PerNu**		
	pro→neut	con→neut	bias	pro→neut	con→neut	bias	pro→neut	con→neut	bias	pro→neut	con→neut	bias
ssup	1.26	-1.36	2.62	-0.01	-1.08	1.07	-1.72	-1.20	-0.52	1.16	-0.64	1.80
sup	-1.00	-0.08	-0.93	-0.30	-0.79	0.49	0.21	-0.50	0.71	0.13	-0.40	0.53
und	1.13	-0.18	1.30	-1.21	0.98	-2.19	1.23	1.40	-1.17	0.64	1.90	-1.26
opp	-0.39	-0.15	-0.25	-0.32	-0.53	0.20	0.63	-0.67	1.29	-0.15	0.05	-0.20
sopp	-0.99	1.76	-2.75	1.83	1.41	0.42	-0.34	0.96	-1.31	-1.79	-0.91	-0.88
total bias			7.85			4.37			4.00			4.47

First, we will focus on the *Baseline* condition to analyze results in the absence of a bias mitigation approach. We found that the misclassification rates for pro statements are similarly high for all worker categories, with the largest rate for *strong supporters* (1.26). In case of contra statements we found a larger gap between *strong supporters* (-1.36) and *strong opposers* (1.76), meaning that *strong opposers* are significantly more likely to misclassify a contra statement as neutral compared to *strong supporters*. We conducted a one-way ANOVA to investigate the effect of the worker category on the *con→neut* misclassification rate. We found a significant difference between the 5 worker categories at the $p < 0.05$ level; $F(4, 509) = 2.85$. Post-hoc comparisons using the Tukey-HSD test revealed a significant difference between *strong supporters* and *strong opposers* at the $p < 0.05$ level with a medium effect size; Hedge's *g = 0.54*.

The bias measure shows that strong supporters exhibit pro bias (2.62) and strong opposers exhibit contra bias (-2.75). We do not observe this in case of the other worker categories.

5.3.4 Bias Mitigation

As depicted in Table 5.3, we see that the total bias for all three mitigation approaches is reduced compared to *Baseline* with *AwaRe* achieving the lowest score. A one-way ANOVA revealed a significant effect of our interventions on the total bias meaure; $p = 0.0002$, $F(3, 1784) = 6.57$. Post-hoc Tukey-HSD test revealed a significant difference between *Baseline* and *AwaRe* (*p =0.036*) with a small effect size (Hedge's *g = 0.19*), significant difference between *AwaRe* and *PerNu* (p = 0.001) with a slightly larger effect size (Hedge's g = 0.30), and significant difference between *SoPro* and *PerNu* (*p = 0.022*) with a small effect size (Hedge's *g = 0.19*).

Using *Wilcoxon signed-rank* tests we found that in the case of $con \rightarrow neut$, *AwaRe* performs significantly better against *Baseline* ($p < .05$, effect size: Hedge's *g=0.24*) and *PerNu* ($p < .01$, effect size: Hedge's *g=0.72*). In the case of $pro \rightarrow neut$ the

misclassification rates do not show any significant difference, apart from *AwaRe* being significantly better than *PerNu* ($p < .01$, effect size: Hedge's g=*0.45*). To control for Type-I error inflation in our multiple comparisons, we used the Holm-Bonferroni correction for family-wise error rate (FWER) [Hol79], at the significance Level of $\alpha < .05$.

SoPro In this condition we found that the normalized *pro→neut* misclassification rate for *strong supporters* drops to -0.01 leading to a decrease in bias compared to the *Baseline*. Additionally, we found that the *pro→neut* misclassification rate increases to 1.83 for *strong opposers*, leading to a drop in bias for *strong opposers* (0.42). This *sopp* bias value is closest to 0 for all conditions. Interestingly, we observe a change in bias for the undecided worker category from 1.30 to -2.19.

AwaRe For this condition, we found that the *pro→neut* misclassification rate for strong supporters drops further to -1.72 when compared to the *Baseline* and *SoPro* conditions. This leads to a small bias that is closer to 0 when compared to the other conditions. For strong opposers we see a bias drop compared to the *Baseline*, from -2.75 to -1.31. The consequent total bias in the *AwaRe* condition was found to be the lowest across all conditions.

PerNu In this case we note that we obtain the highest total bias score amongst our proposed approaches. We still see a non-significant drop in bias for both *strong supporters* and *strong opposers* compared to the *Baseline*.

5.3.5 Impact of Worker Categories on Resulting Quality

An important element in the creation of high-quality ground-truth using crowdsourcing is a diversity of opinion that can manifest from acquiring multiple independent judgments from workers [Sur05]. One of the simplest methods used for aggregating multiple judgments in crowdsourced tasks is majority voting [HTTA13]. In the absence of gold-standard data, and especially for subjective tasks, majority voting or a variation of the algorithm is arguably a popular aggregation technique. Thus, to analyze the potential impact of worker categories on the resulting quality of aggregated judgments, we consider majority voting.

Consider a typical microtask crowdsourcing platform; task completion is generally driven by a self-selection process where workers pick and complete tasks they wish to [CHMA10]. Various factors ranging from worker motivation [KSV11, RKK+11] to marketplace dynamics such as task availability [DCD+15, JSPW17], dictate which workers end up self-selecting and completing a given task from the available group of workers at any given point in time. Based on our findings pertaining to worker categories, we know that strong supporters and strong opposers correspond to the most systemic bias (see *Baseline* condition in Table 5.3). To measure the impact of workers from different categories on the average quality of aggregated judgments, we carry out simulations consisting of randomly selected workers from all categories. To this end, considering all worker categories, we ran 10,000 simulations of acquiring judgments from randomly teamed worker combinations with *N=3* and *N=5* for each of the 30 statements. Requesters often use 3 or 5 workers to gather redundant judgments and ensure quality. This is also recommended practice on

FigureEight. Thus, this setting replicates standard crowdsourcing task configurations of obtaining multiple judgments from workers and assigning a label after aggregation.

Table 5.4. Misclassification rates for random worker samples across worker categories. all = all workers, $strong$ = strong supporters/strong opposers, $\overline{strong}$ = without strong supporters/strong opposers. Sample sizes N = 3, 5. Lowest misclassification rates are highlighted for each condition. Including only workers with a $strong$ opinion leads to higher misclassification rates.

		Baseline		**SoPro**		**AwaRe**		**PerNu**	
		pro→neut	con→neut	pro→neut	con→neut	pro→neut	con→neut	pro→neut	con→neut
N=3	all	0.272	0.042	0.255	0.033	**0.223**	0.018	0.349	**0.114**
	$strong$	0.353	0.141	0.267	0.060	0.236	0.035	0.420	0.0348
	$\overline{strong}$	**0.267**	**0.037**	**0.249**	**0.028**	0.251	**0.015**	**0.329**	0.115
N=5	all	0.242	0.017	**0.224**	0.011	**0.192**	0.005	0.329	0.068
	$strong$	0.326	0.121	0.228	0.015	0.210	0.016	0.405	**0.023**
	$\overline{strong}$	**0.237**	**0.014**	**0.224**	**0.009**	0.230	**0.003**	**0.308**	0.069

To investigate the impact of strong supporters and strong opposers on the resulting quality, we consider three grouping strategies. 'all' considers the set of all workers (here N=3 or N=5 workers are picked at random from the entire pool of all workers), '$strong$' is the set of workers who exhibited a strong bias; strong supporters for *pro→neut* and strong opposers for *con→neut* (here N=3 or N=5 workers are picked at random from the subset of strong supporters and strong opposers), and '$\overline{strong}$' is the set of workers present in all after filtering out all workers from the $strong$ subset. Table 5.4 presents the average *pro→neut* and *con→neut* misclassification rates for worker groups across the 10,000 runs in each of the conditions.

When randomly selecting worker samples from the full set of workers (all), we see that the total misclassification rates drop as compared to the average misclassification rates per worker in Figure 5.6. This shows that groups of workers achieve higher accuracy, even when workers with strong opinions are included.

In the *Baseline* scenario, the misclassification rate of the $strong$ worker group exhibits higher misclassification rate when compared to $\overline{strong}$. We assess the significance of the misclassification rate through the Kruskal-Wallis test, which yields a significant difference with $p < .01$. This result is intuitive as the presence of workers in the end of both extremes (*ssup* and *sopp*), adds to the amount of biased judgments collected for a given subjective task.

5.3.6 Effects of Worker Level

Workers on the FigureEight platform can earn three different Level badges based on their accuracy and experience over time. In the FigureEight job settings, Level 1 is described as "All qualified contributors", Level 2 as a "Smaller group of more experienced, higher

accuracy contributors", and Level 3 as the "Smallest group of most experienced, highest accuracy contributors". We acquired self-reported worker Levels in our study. This allows us the opportunity to analyze potential correlations between worker performance/bias and the worker experience as represented by the worker level.

Figure 5.7. Number of workers per worker Level for all conditions.

Figure 5.7 shows the distributions of workers of each Level across the different conditions. In all conditions, Level 3 workers are the largest group of workers. There were more Level 1 workers than Level 2 workers in the *Baseline* condition, while for the other two conditions the number of Level 2 workers is higher.

Table 5.5. Misclassification rates and bias for each worker Level and specific worker categories and orientations. The **total** shows the overall misclassification rate for workers of the given level. For each condition, we highlight the highest pos. bias score for **ssup** and the highest neg. bias score for **sopp** across the worker levels. The results show that no single level group clearly outperforms the other level groups across conditions.

	Baseline			**SoPro**			**AwaRe**			**PerNu**		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
total	0.21	0.19	0.19	0.20	0.21	0.15	0.21	0.14	0.15	0.21	0.20	0.25
ssup pro	0.29	0.40	0.32	0.29	0.44	0.26	0.14	0.24	0.29	0.35	0.47	0.37
ssup con	0.11	0.09	0.05	0.02	0.13	0.07	0.03	0.04	0.03	0.17	0.17	0.25
ssup bias	0.97	**2.45**	0.52	1.04	**2.37**	-0.60	-0.04	**0.48**	-0.58	-1.32	-0.84	**1.11**
sopp pro	0.23	0.21	0.33	0.46	0.28	0.38	0.43	0.19	0.33	0.33	0.50	0.21
sopp con	0.25	0.21	0.14	0.11	0.28	0.08	0.21	0.06	0.20	0.17	0.00	0.24
sopp bias	-2.55	**-2.97**	-1.21	1.49	**-2.57**	2.0	0.86	-1.22	**-1.65**	**-1.39**	1.71	-0.62

Table 5.5 shows the average misclassification rates for the different approaches and the corresponding worker levels. Overall, the results vary. While we see high bias values for level 2 workers for *Baseline* and *SoPro*, the bias values for *AwaRe* and *PerNu* are mixed. We computed a non-parametric *Kruskal-Wallis* test to asses the correlation between the worker level and their misclassification rates for *pro* and *con* statements. In this case, we do not distinguish between *ssup* and *sopp*. The test revealed that none of the bias differences

between worker levels are significant. As a consequence, we do not control for worker level in our bias analysis.

5.3.7 Implication of Strong Supporters and Strong Opposers on Resulting Quality

Our findings show that the strong supporters and strong opposers are most susceptible to systemic bias due to their strong opinions. Let us consider the impact of a *ssup* or *sopp* contributing to a task where multiple judgments are aggregated using majority voting. In such a setting, *ssup* or *sopp* can bias a task outcome if there is a majority of either *ssup*s or *sopp*s in the cohort of workers annotating the same statement. To quantify the possible implication, we draw random samples of k workers $(k = 1...102)$ from the *Baseline* condition and assess the fraction of resulting biased outcomes for each k, averaged across 10,000 iterations. Our findings are presented in Figure 5.8. We note that if 3 judgments are collected for each statement, over 17% of the statements end up with a biased label. With 5 judgments, over 15% end up with a biased label. This converges to around 10% around *k=60*. Requesters seldom gather so many judgments on a single statement, especially in large-scale jobs where costs are an important trade-off. This shows that the presence of *ssup* or *sopp* can be undesirable if their susceptibility to their opinions and the resulting bias is not mitigated.

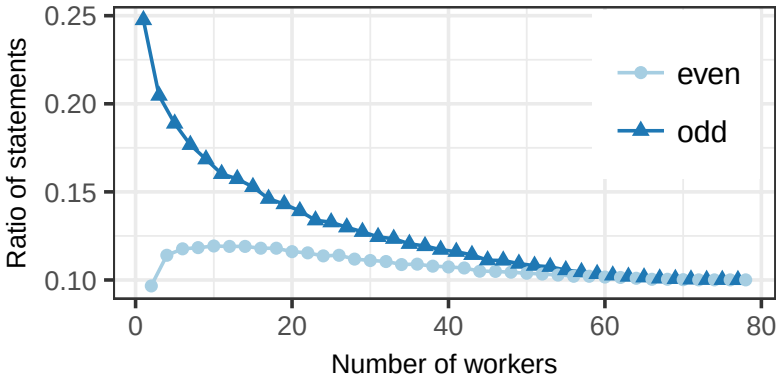


Figure 5.8. Ratio of statements whose labels are provided by a majority of biased workers after k workers (random worker samples from the *Baseline* condition, averaged over $10K$ iterations). 'k' is split between even and odd for readability. In case of a tie there is no majority of biased workers.

5.4 Discussion

Intuitively, workers belonging to the extreme categories (*ssup* and *sopp*) exhibit systematic bias stemming from their personal opinions. We found evidence of this, where *ssup* and

sopp provided biased judgments inline with their stance on a given topic. Table 5.3 shows that *ssup* and *sopp* have the highest biased scores as a consequence of their unbalanced misclassification rates for pro and contra statements. This finding supports hypothesis **H#1**.

Impact on Resulting Ground-Truth Quality. Our findings suggest that negative effects on the created annotations due to biased workers can be effectively canceled out by increasing redundancy. By including non-biased workers in the ground-truth creation, misclassifications stemming from personal opinions can be averted. Importantly, the biggest threat to introducing systemic bias is having a group with a large majority of biased workers contributing to a task.

Implications of Bias. If we opt for a majority voting label aggregation scheme, even for fairly simple tasks with binary outcomes (i.e, *"opinonated"* or *"neutral"*) the amount of judgments needed to overcome bias is very high. Figure 5.8 shows that if we consider *odd* numbers of judgments (less than 5), more than 20% of task units end up with a group of workers who are susceptible towards their strong stance (*ssup* or *sopp*). To reduce the amount of statements which end up with a majority of workers in the extreme categories, the number of judgments needs to be extremely high, i.e, more than 40. Contrary, in the case of *even* number of judgments, this ratio is lower due to the fact that often we end up with a tie, and thus cannot employ the majority voting scheme. Therefore, in such cases we may end up with a large portion of statements without a clear aggregated label.

Bias Mitigation. To avoid biased judgments, we aim to mitigate bias through social projection (*SoPro*) or by making workers aware of their possible inclinations towards a topic (*AwaRe* and *PerNu*).

Our analysis results show that all three approaches reduce the total bias and average bias for workers with extreme opinions when compared to the *Baseline*. Additionally, *SoPro* and *AwaRe* lead to an improvement in general worker performance by reducing the overall number of misclassifications, therefore increasing the quality of resulting ground-truth labels in general. We achieve the highest total bias reduction rate of 49% with the *AwaRe* approach as well as the highest bias reduction for *ssup*s. For *sopp*s the *SoPro* approach receives the highest reduction and might therefore be the preferred approach for situations with a large number of *sopp*s. Another trade-off is the task completion time (TCT). The average TCT for *AwaRe* is significantly higher in our analysis compared to the *Baseline*, while the increase for *SoPro* is not significant.

We note that the most sophisticated approach *PerNu* provides significantly worse results for both bias and general worker performance. This behaviour can possibly be explained through the theory of *central* and *peripheral* persuasion from marketing research [FHW+02]. The *AwaRe* approach falls into the category of *peripheral* persuasion, where general reminder snippets increase worker awareness regarding the potential bias entailing the task. In contrast, *PerNu* can be seen as a *central* persuasion technique, where personalized instructions are

provided at the statement level, thus, actively and directly informing workers while they make their judgments. According to [FHW+02], peripheral persuasion techniques are most suitable in inflicting attitude change when compared to central persuasion ones. We will investigate this further in our future work.

Worker Level Effects Our results across different worker levels show that there is no significant difference in bias scores between workers with varying experience. This shows that filtering by levels is not a reliable strategy for mitigating worker bias. As a consequence, due to the lack of support we reject hypothesis **H#2**. We note that Level 1 and 2 workers appear to be more receptive of treatment interventions provided by the different bias mitigating approaches. However, further qualitative studies are needed to establish this.

5.5 Conclusions

Systematic bias stemming from worker opinions can be a major problem in subjective labeling tasks. We showed that crowdsourced ground-truth annotations are susceptible to potentially biased workers who tend to produce systematically biased and noisy labels.

Our results show that judgments of workers who have extreme personal stances (i.e. *strong supporters* or *strong opposers*) pertaining to a particular topic, show a significant tendency to be influenced by their opinions. We found that performance or experience indicators like worker levels, do not play a significant role in reducing misclassification rates, making such indicators unreliable in mitigating systemic biases stemming from opinions in subjective tasks.

To mitigate such aforementioned worker bias, we proposed interventions based on social projection and making workers aware of their personal stances and potential biases, thus encouraging them to set aside their personal opinions during the course of task completion. Our approaches, *SoPro* and *AwaRe* provide significant improvement in terms of both worker bias reduction and general worker performance. Finally, we found that the *PerNu* approach, which actively provides the worker with personalized bias-related feedback during the task completion, does not provide any improvement over the other bias mitigation approaches.

6 Debiasing Word Embeddings from Sentiment Associations in Names

Word embeddings are one of the most basic representations of words in natural language understanding. Their use in downstream tasks has shown great benefits on a variety of tasks, such as named entity recognition and part of speech tagging [MSC+13a, PSM14] based on their ability to capture the word context.

Due to the way embeddings are trained, they often have been shown to contain biases, e.g. *gender bias*. These biases are reflected in terms of words that are supposedly to be either *gender neutral* or any other form of *word categorization*, but instead are shown to be in close proximity to words that belong to explicit categories, such as *gender*. Research has shown that specific job roles reflect stereotypes of a specific culture or group [BCZ+16, CBN17], e.g. *"nurse"* being closer to *"female"* and *"programmer"* being closer to *"man"*, etc. Such findings concur with sociolinguistic theory [Hal70], which states that language and its structures is a medium that is in the function of its *social groups*.

In the case of word embeddings the biases stem from the underlying training corpus. Even for large corpora, approaches such as word2vec [MSC+13a] trained on Google News, and GloVe [PSM14] trained on Wikipedia, exhibit bias regarding gender and race.

Current state of the art approaches aim to debias embeddings by removing the direction of the *protected attribute* (e.g. gender), so that a target word is *equidistant* to all categories of the protected attribute (e.g. *gender roles*)[1]. This type of intervention is done either as a post-processing step [BCZ+16], pre-processing step [DJL+18], or directly during training [ZZL+18]. However, analysis [GG19] has shown that these approaches mitigate bias only at a superficial level, where much of the initial bias can be recovered through *word proxies*, words that are in close proximity in the vector space to the words being used for bias mitigation.

In this chapter, we tackle the problem of *sentiment associations* with *names* in word embeddings. That is, given a training corpus, some names may be co-occurring more

[1] In most cases this is done at a binary level, e.g. *male* and *female* for genders.

frequently with words of either positive or negative sentiment. Hence, according to the distributional hypothesis [Har54], the embeddings of such names will be close to the words with explicit sentiment with which they co-occur. For instance, Caliskan et al. [CBN17] shows that *European American* first names are associated with pleasant words, while *African American* names are comparably stronger related with unpleasant words. Furthermore, *male* first names are associated with career words, whereas *female* names are stronger connected to family words. Similarly, we show that last names suffer from similar biases, and that these biases can heavily impact the decision-making of downstream models such as sentiment classifiers [KM18, DJL+18].

For example, if we are given two *unnuanced factual* sentences as the ones shown below, a reliable sentiment classifier that uses embeddings as a means to represent the words in a sentence should classify both statements as *neutral*.

- *"**Obama** is president."*
- *"**Trump** is president."*

While for specific tasks, sentiment association of names may be essential, in the case of pre-trained word embeddings, where a name may correspond to multiple real-world persons, they are unsuitable as they result in significant discrimination and other forms of biases.

We propose *DebiasEmb*, a novel approach that debiases word embeddings from sentiment associations in names during training. During the training phase of the skip-gram model, apart from the objective of predicting the *context word*, through an *oracle classifier* we additionally ensure that the *center words* of interest cannot be associated with neither positive or negative sentiment. The oracle classifier in this case is a pre-trained sentiment classification model with positive and negative words, respectively their embeddings. Hence, the novelty of DebiasEmb is that it seemlessly integrates the standard word embedding objectives together with any debiasing component, such as debiasing sentiment or other word categories. To show the effectiveness of our approach, we evaluate DebiasEmb on two sets of tasks. First, at word level, we ensure that the names from a given name list can not be associated with neither the positive nor the negative sentiment class and that the constructed embeddings do not suffer in terms of quality when compared to the original skip-gram embeddings. Second, we show the debiasing effect of DebiasEmb on a text-level sentiment classifier. In summary, our contributions are the following:

- DebiasEmb, a novel approach for debiasing word embeddings from sentiment associations in names;
- Thorough evaluation of the approach, including an extrinsic downstream analysis based on a sentiment classifier trained on two different datasets (reviews and news data).

6.1 Related Work

Research on bias in word embeddings focuses strongly on gender bias and racial bias. The seminal work by Bolukbasi et al. [BCZ+16] shows how a gender direction in the vector space of word embeddings can be defined by using pairs of gender specific words, namely feminine and masculine words (*she-he* pairs). They point out that many gender-neutral words are associated with one gender, e.g. *doctor* is strongly shifted towards male, while *nurse* is associated with the female gender, reflecting societal gender stereotypes, even when using popular training datasets such as Google News articles. To mitigate gender bias in word embeddings they introduce a post-processing approach that removes the identified gender direction from a pre-defined list of words, while keeping it for words that convey an explicit gender function (e.g. *mother*, *father*, *boy*, *girl*). This approach has been criticized for being "fairness through blindness", i.e. removing relevant information while not covering important bias aspects such as proxies [CBN17], and for relying on a classifier to identify definition words which could lead to errors being propagated into the model [ZZL+18].

In their approach Zhao et al. [ZZL+18] aim at isolating the gender attribute into one component of the resulting word vectors, while removing it from all other components. This is achieved during training by modifying the loss function and using a list of gender seed words.

However, a recent study by Gonen and Goldberg [GG19] shows that both, the post-processing and the isolation approach, remove bias only at a superficial level. The gender bias is still present in the resulting embeddings. An interesting observation in this case is that the gender direction, that is limited to a specific word list, does not cover sufficiently other word proxies that may introduce bias. Hence, this should be used with precaution and can serve only as an indicator for bias. On the other hand, the debiasing approach that is done during the training phase [ZZL+18] is preferable over post-processing. However, even in this case, the limitation is in defining genders, namely words that are specific for a gender. This is similar to the work by Bolukbasi et al. [BCZ+16].

Contrary to the previously described works, our approach has the advantage of using a pre-defined classification model (i.e. the oracle sentiment classifier) that is trained on a specific seed set of words, respectively their embeddings, with explicit sentiment. Consequentially, this allows us to address word proxies that may cause sentiment association to names, or other biases for other word categories like gender, race etc., through the similarities in the vector space of the set of seed words with other proxy words.

Caliskan et al. caliskan2017semantics introduce the Word Embedding Association Test (WEAT), an intrinsic test for measuring biases in word embeddings. It determines the mean proximity of words in two target groups to words in two attribute groups (e.g. *Pleasant*, *Unpleasant*). Swinger et al. swinger2018biases propose an unsupervised algorithm for automatically outputting WEAT tests that does not require the sensitive group (e.g. gender, race) to be specified. They find gender biases even for word embeddings that have been debiased using the approach introduced by [BCZ+16]. WEAT has recently been extended for measuring bias in sentence encoders [MWB+19].

In contrast to the works by Caliskan et al. and Swinger et al. we measure bias not intrinsically, but extrinsically on downstream tasks to observe the actual impact that the identified biases have on the behavior of downstream models. We also propose an approach for debiasing embeddings.

In their analysis, Diaz et al. [DJL+18] find significant age-related bias in a variety of sentiment analysis models and popular GloVe word embeddings. They introduce a simplistic approach for debiasing by removing all occurrences of the protected attribute (i.e. age) from the input data. In contrast, our approach does not modify the input data but instead debiases embeddings during training.

Recent research has introduced a new generation of contextualized word embeddings that are able to represent polysemy. [ZWY+19] show that contextualized word embeddings such as ELMo [PNI+18] show similar bias compared to common word embeddings such as GloVe. They find gender bias in the trained embeddings intrinsically and mitigate bias on the downstream task of coreference resolution by leveraging augmented training data with swapped gender words during training or post-processing. A shortcoming of this approach is that it only applies to gender, since in other contexts (e.g. race, sentiment), a clear opposite word can not be defined. In contrast, our approach is domain-independent and applicable to all types of class-based biases.

Our work in previous chapters aims to detect biased language in text directly using feature-based (Chapter 3) or neural-based (Chapter 4) approaches. The approach by [DJL+18] heuristically identifies all occurrences of the protected attribute. These approaches could be used to remove biased statements from the training data, though it is not clear whether this would cover all types of biases, especially the ones that are introduced through proxies. Using a pre-processing approach that removes entire statements containing biased language would also lead to a situation where valuable parts of the training data co-occurring with biased parts would be lost as well a parts being misclassified as bias. Our approach does not remove training data but instead debiases the resulting embeddings during training.

6.2 Debiased Word Embeddings

In this section, we describe our approach *DebiasEmb* for training debiased word embeddings. *DebiasEmb* consists of two main components: (i) an oracle sentiment classifier, and (ii) the modified skip-gram with negative sampling (SGNS) model. In the following we explain in details the individual components.

6.2.1 Oracle Sentiment Classifier

To determine *prior sentiment bias* towards names in word embeddings we use pre-trained supervised models that are trained on embeddings from words that contain *explicit* prior sentiment, e.g. lexicons of *positive* and *negative* sentiment from SentiWordNet [BES10] or other hand-crafted lexicons [LZ12]. More specifically, for any other words, e.g. *proper*

nouns, if such a model is able to classify them as either "*positive*" or *"negative"*, that is an indicator of bias in the corresponding word embedding. For example, *"Smith"*, *"Li"*, or *"Mohamed"* should not be associated with any prior sentiment.

In this work, we consider the oracle classifier to be a pre-trained *logistic regression* model (see Equation 6.1), which for each embedding dimension associates a feature weight. However, any classification model that uses a differentiable classification function can be used in this case.

$$f_{LR}(\mathbf{x}) = \sigma\left(\frac{1}{1 + e^{-(\mathbf{w}\cdot\mathbf{x}+b)}}\right), \tag{6.1}$$

where, $\mathbf{x}$ represents the embedding of a specific word, and $\mathbf{w}$ represents the weights associated with each embedding dimension.

In an ideal scenario, the classifier outputs $f_{LR} = 0.5$, which results in the *inability* of the classifier to predict the sentiment of a word from the protected class. In the next section, we show how we debias word embeddings given some pre-trained sentiment classifier. We will use the classification model to guide the debiasing process of word embeddings such that target names cannot be associated with any sentiment score.

6.2.2 Debiased Word Embedding Model

The main intuition behind the skip-gram with negative sampling model [MSC+13b] is to use the *context* of a *center* word to learn its representation. The objective function of the SGNS model in Equation 6.2 is to maximize the similarity of the center and context words, while at the same time minimize the similarity of the center word against non-context words (negative samples).

$$f_{SGNS} = \frac{1}{N}\sum_{i=1}^{N}\sum_{-c\leq j\leq c}\log p(w_{i+j}|w_i) \tag{6.2}$$

where, c is the context window of the center word. The above function shows the ability of the model to predict the context word w_{i+j} given the center word w_i. The training is done by minimizing the following loss function $\mathcal{L}$.

$$\mathcal{L}_{sgns} = \log\sigma(\mathbf{v}_{\mathbf{w_j}}\cdot\mathbf{u}_{\mathbf{w_i}}) + \sum_{k=i}^{K}\mathbb{E}_{w_k\sim P(w_k)}\log\sigma(-\mathbf{v}_{\mathbf{w_k}}\cdot\mathbf{u}_{\mathbf{w_i}}) \tag{6.3}$$

Despite the ability of these models to efficiently capture the word meaning based on its context, there are several documented issues that the embedding models capture, such as *societal biases* (i.e. gender, racial biases) and other issues that are encoded in the textual resources, from which the training data are drawn [BCZ+16, ZWY+19, EG18].

We propose a modified version of the SGNS model, where we modify the loss function such that the computed word embeddings for a target set of words in the vocabulary, e.g. names are not associated with any sentiment score.

$$\mathcal{L}_{sent} = \left| \sigma\left(\mathbf{w} \cdot \mathbf{u}_{w_i}\right) - \frac{1}{2} \right| \tag{6.4}$$

where, $\sigma(\cdot)$ represents the pre-trained oracle classifier, respectively, we use the weights associated with the embedding dimensions and assess whether the word embedding $\mathbf{u}_{w_i}$ of our center word w_i encodes any prior sentiment bias. For a word embedding to be bias free, the classifier should not be able to distinguish between the *positive* or *negative* sentiment categories, thus, the value of the σ function will be equal to 0.5, which will result in zero loss. In the other cases, we aim at changing the embeddings of a center word $\mathbf{u}_{w_i}$ such that the distance in $\mathcal{L}_{sent}$ is minimized.

Finally, the loss function of *DebiasEmb* is the combined sum of the original loss function of the SGNS model and the loss function that measures the sentiment bias score.

$$\mathcal{L} = \mathcal{L}_{sgns} + \mathcal{L}_{sent} \tag{6.5}$$

Note that, our model can incorporate any oracle classifier whose classification function is differentiable, and additionally, we are not limited to only sentiment bias.

6.3 Word Embedding Experimental Setup

In this section, we describe the evaluation setup for our approach *DebiasEmb*. Namely, we evaluate its capability to reduce sentiment bias association with names. Additionally, we introduce competitors against which we compare and the evaluation metrics to measure sentiment bias in word embeddings.

6.3.1 Datasets for Training Word Embeddings

Word embeddings trained on news datasets have been widely used for a span of different tasks, including sentiment analysis [GVD+17], part-of-speech tagging [WQS+15], and named entity recognition [Sie15]. For the evaluation of *DebiasEmb*, we use six different news datasets with randomly selected sentences from popular news sources in the time period of 2013-2017. The specifics of each dataset are shown in Table 6.1. The *Random* dataset contains sentences that were randomly selected from a set of more than 100 different news sources.

Table 6.1. Number of sentences for each news dataset.

	#Sentences
HuffingtonPost	4,631,874
Random	4,477,326
Breitbart	1,333,510
CNN	1,252,718
BBC	773,657
RussiaToday	457,487

6.3.2 Name List

We extract an extensive list of 130,000 surnames that are used in the United States of America from Mongabay[2]. From this list, we filter out all names that occur less than 10 times in our combined news dataset, as well as all names that occur more often in lower case than in uppercase (first letter of the word is capital), indicating that these names have ambiguous meaning and are used as non-name words more often (e.g. *Bottom*, *Speech*). This results in 17,055 names for which we aim to debias the word embeddings.

6.3.3 Baselines for Debiasing Word Embeddings

We compare DebiasEmb against the following baselines:

- **SGNS:** Default SkipGram model with negative sampling, as introduced by [MSC+13b], no debiasing.
- **PostDebiasing:** Debiasing approach introduced in [BCZ+16]. Instead of the *gender direction* (*he - she*), we use a *sentiment direction* (*positive - negative*). We replace the definition of word pairs with a set of positive-negative pairs, e.g. great - terrible, positive - negative, competent - incompetent. We first train the word embeddings using SGNS and then remove the sentiment direction from all name words by applying the post-processing step.
- **PreDebiasing:** Debiasing approach introduced in [DJL+18]. The idea is to remove all occurrences of the protected attribute from the input data. Instead of sentences containing age words, we remove all sentences containing at least one name from the name list and at least one positive or negative word.
- **PreDebiasEmb:** Combination of PreDebiasing and DebiasEmb. We first apply PreDebiasing on the input data and then use DebiasEmb during training.

[2] https://names.mongabay.com/data/surnames_A.htm

We do not compare against the approach introduced by [ZZL+18] since this approach is based on the same definition of the protected attribute as the approach by [BCZ+16]. Focusing the sentiment direction into one component of the resulting vectors and then removing this component is conceptionally the same approach as instantly removing the sentiment direction from the embeddings.

6.3.4 Word-Level Sentiment Classifier

For the evaluation on word-level, we use a classifier that is identical with the oracle sentiment classifier, as introduced in section 6.2.1. Based on the embedding representation of the input word, it outputs the class probability for both classes. Due to the classification task being binary and the probabilities summing up to 1, it is sufficient to focus on the probability of the positive class. Its score is between 0 (negative) and 1 (positive). Both, the oracle and the evaluation classifier have been trained on a word lexicon containing 2004 positive and 4782 negative words [LZ12].

6.3.5 Bias Measures

To evaluate the amount of *sentiment bias* associated with names for some given embeddings, we consider the following measures that use the *word-level sentiment classifier* to obtain the classification label and class probability scores for a given name.

Dist: Here we measure the ability, respectively, the inability of a classification model $\sigma(\cdot))$ to categorize a name as either having *positive* or *negative* sentiment. For a binary classification model $\sigma(\cdot)$, a class probability of 0.5 results in the model's inability to categorize the name into either of the sentiment categories. Thus, the smaller the distance to 0.5 the lower the bias. For a given set of names N, we formalize **Dist** as the mean score across all names.

$$\textbf{Dist} = \frac{1}{|N|} \sum_{n \in N} |\sigma(\mathbf{w} \cdot \mathbf{u}_n) - 0.5| \tag{6.6}$$

Var: In addition to **Dist** we also take into account the variance of classification scores $\sigma(\cdot)$ across names. Through this measure we aim at capturing if the produced embeddings have varying bias behavior across the different names. For instance, if *all names* are categorized with the *same* sentiment category, e.g. *positive sentiment* with high class probability of 0.9, consequentially, the embeddings do not discriminate against specific names, however, they contain positive bias towards names with **Dist**=0.4. On the contrary, if for specific names the classification model $\sigma(\cdot)$ yields varying sentiment categories with varying probability scores, then, the resulting embeddings discriminate against specific names. Thus, values that are zero or close to zero, indicate bias free embeddings.

6.4 Word-level Evaluation Results

In this section, we show the evaluation results at the word level. First, we report the results in terms of bias for the different competing approaches. Second, we report the quality of the computed embeddings, computed on standard benchamarking datasets. Finally, we show a detailed analysis where we analyze the association of name embeddings with positive/negative words and their proxies.

6.4.1 Embedding Debiasing Results

Table 6.2. Mean distance from 0.5 (Dist) and variance (Var) for name words using the *word-level* sentiment classifier. Results are shown for all approaches and all datasets. Lowest values for Dist and Var are highlighted.

	SGNS		**DebiasEmb**		**PostDebiasing**		**PreDebiasing**		**PreDebiasEmb**	
	Dist	Var	Dist	Var	Dist	Var	Dist	Var	Dist	Var
BBC	0.246	0.0554	0.151	0.0141	0.344	0.0206	0.093	0.0158	0.073	0.0109
Breitbart	0.211	0.0413	0.087	0.0043	0.197	0.0383	0.082	0.0167	0.034	0.0031
CNN	0.214	0.0164	0.160	0.0005	0.274	0.0124	0.102	0.0175	0.087	0.0048
HuffingtonPost	0.214	0.0210	0.114	0.0012	0.161	0.0300	0.102	0.0208	0.121	0.0108
RussiaToday	0.194	0.0103	0.028	0.0001	0.137	0.0207	0.068	0.0103	0.039	0.0060
Random	0.227	0.0554	0.086	0.0088	0.320	0.0280	0.098	0.0159	0.104	0.0080
Mean	0.218	0.0333	0.104	**0.0048**	0.239	0.0250	0.091	0.0162	**0.076**	0.0073

Debiasing Results. Table 6.2 shows the results for the **Dist** and **Var** measures based on the word-level sentiment classifier for all datasets and all approaches. We observe a debiasing effect of DebiasEmb for all news datasets. On average, DebiasEmb outperforms all baselines in terms of **Var** with a relative improvement of 86% when compared to SGNS. It also achieves a relative improvement of 52% for **Dist** compared to SGNS, showing that DebiasEmb not only debiases the embeddings, resulting in the inability of the word-level sentiment classifier in classifying names, but it additionally increases the homogeneity of the class probabilities across names and therefore decreases the name bias of the classifier.

On the other hand, the PostDebiasing baseline does not show significant improvement over SGNS. Contrary, the PreDebiasing approach performs well, especially for the **Dist** measure. However, when combining the PreDebiasing and DebiasEmb, we achieve the lowest **Dist** value with a relative improvement of 65% compared to SGNS and a relative improvement of 78% in terms of **Var**. This shows that PreDebiasing is another effective approach for debiasing embeddings, especially when combined with DebiasEmb. The results are consistent across all news datasets that were used for training embeddings, with only minor differences.

In summary, the results from Table 6.2 show that sentiment bias associated with names is present in news corpora. Approaches for training embeddings, like SGNS, associate names with sentiment categories, a consequence of the training corpora, where names co-occur with words that have explicit sentiment valence. These associations can not be remedied by simply relying on specific lexicons for debiasing as in the case of PostDebiasing. Our approach through an oracle classifier can guide the computation of embeddings such that, apart from capturing word meaning following the distributional hypothesis [Har54], it additionally constrains the parameters of the embeddings and does not allow names from a given target set to be associated with any sentiment category.

Word-level Classifier Accuracy. Table 6.3 shows the word-level sentiment classification accuracy during training with the corresponding embeddings of the words in the lexicon containing positive and negative words (cf. Section 6.3.4). The scores are shown for all five approaches, respectively the produced embeddings, averaged across all news datasets.

In terms of accuracy, SGNS and PostDebiasing achieve the best performance. DebiasEmb and the Predebiasing combination achieve slightly lower performance. This shows that there is a slight trade-off in terms of debiasing embeddings and correspondingly the ability of the word-level classifier to distinguish between positive and negative words. While a high classification score correlates with the reliability of the bias measures, in the following sections we show that in terms of embedding quality in standard benchmarks, the embeddings computed through DebiasEmb have only very minor difference. And as we will show in the downstream task evaluation in Section 6.5, where we train a text-level sentiment classifier, models that represent the word using DebiasEmb embeddings achieve significantly lower bias in determining the sentiment of a sentence, that is, with varying names the sentiment of the sentence does not change.

Table 6.3. Average word-level model accuracy for all approaches.

	Accuracy
SGNS	0.77
DebiasEmb	0.72
PostDebiasing	0.78
PreDebiasing	0.70
PreDebiasEmb	0.69

6.4.2 Benchmark Testing

To ensure that the quality of the resulting embeddings does not suffer due to the debiasing efforts, we compare the computed embeddings based on our DebiasEmb approach, and other competitors, against the standard SGNS embeddings on standard benchmark tests. It is important to note here that due to the limited genre and scope of news corpora, the

Table 6.4. Performance on benchmarks for word embeddings. The results show that debiasing efforts do not have any negative significant impact on the performance of the embeddings.

	SGNS	DebiasEmb	PreDebiasing	PreDebiasEmb
AP	0.286	0.289	0.268	0.255
BLESS	0.325	0.324	0.307	0.300
Battig	0.178	0.180	0.157	0.155
ESSLI_1a	0.489	0.473	0.511	0.477
ESSLI_2b	0.650	0.621	0.700	0.758
ESSLI_2c	0.478	0.478	0.485	0.478
MEN	0.194	0.192	0.202	0.204
MTurk	0.296	0.293	0.300	0.297
RG65	0.153	0.154	0.085	0.073
RW	0.084	0.080	0.187	0.185
SimLex999	0.058	0.058	0.088	0.093
TR9856	0.101	0.102	0.112	0.114
WS353	0.162	0.171	0.220	0.231
WS353R	0.198	0.206	0.198	0.215
WS353S	0.194	0.203	0.295	0.309
Google	0.076	0.075	0.043	0.044
MSR	0.131	0.130	0.070	0.071
SemEval201	0.088	0.086	0.092	0.092

scores on certain benchmarks may be lower when compared to embeddings trained on more generic corpora like Wikipedia. Hence, we use the SGNS embeddings as a reference point for comparison.

Table 6.4 show the results for the different types of embeddings. Each value is computed as the mean over all the embeddings from all the different news sources. We note that there is no significant difference between the different embeddings. Thus, concluding that such debiasing efforts do not harm the resulting quality of embeddings.

6.4.3 Arrangement of Names in Vector Space

Figure 6.1 shows the projection of the name embeddings[3] based on the t-SNE [MH08] non-linear dimensionality reduction technique. The projection on the two most important components reveals that the names from DebiasEmb are more closely clustered together, and furthermore are more distant to words with explicit sentiment and their close proxies. On the positive dimension, the Euclidean distance in the case of DebiasEmb is $DebiasEmb_{POS} = 92.71$, contrary to $SGNS_{POS} = 79.83$. Similarly, on the negative dimension the Euclidean distance in the case of DebiasEmb is $DebiasEmb_{NEG} = 91.72$, contrary to $SGNS_{NEG} = 79.12$. The Euclidean distance confirms the results we achieve in Table 6.2.

[3] We selected predefined names of political persons from US politics (republicans and democrats).

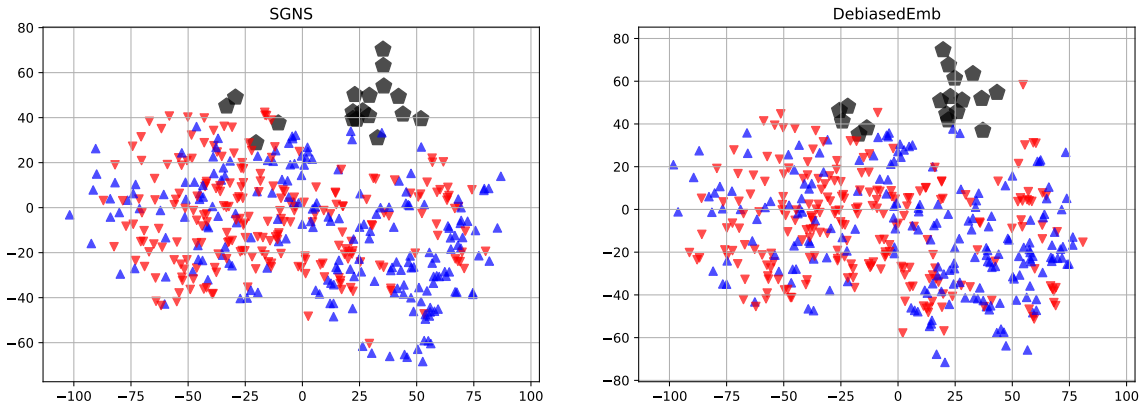


Figure 6.1. Position of word vectors using t-SNE and the two most important components. Black pentagons represent politician names, blue triangles positive words, and red triangles negative words. Names have a farther distance to the positive and negative words in the case of *debiased* word embeddings compared to *SGNS* word embeddings, as measured based on the *Euclidean distance*.

6.5 Downstream Analysis Setup

Word embeddings are rarely leveraged for word-level tasks, such as predicting the sentiment of a single word, more commonly they are applied to sentences or longer textual snippets.

In this section, we introduce the setup for applying the trained embeddings using SGNS and DebiasEmb to a downstream classifier, which predicts the sentiment of a textual snippet containing a name (e.g. movie reviews or quoted statements taken from news articles). We analyse how the encoded name biases in embeddings are spilled onto downstream models, and how using the debiased embeddings changes the behavior of the model.

6.5.1 Classifier Training Data

We use two datasets for labelling text snippets with their sentiment label. The first dataset contains movie reviews and is used widely on this particular task. Whereas, the second contains sentences from news articles, and was created such that the *language genre* is the same as the training corpora of the trained embeddings. Both datasets contain textual snippets only in the English language.

- **Reviews:** This dataset contains 12,500 positive and 12,500 negative movie reviews extracted from the Internet Movie Data Base (IMDB) and is openly available [MDP+11].
- **News:** Since there is no openly available news dataset for sentiment analysis that is suitable for our problem setting, we constructed a news dataset and labeled it by means of crowdsourcing. Similarly to Balahur et al. [BSK+13], we extracted a sample of quotation phrases from our news collection containing at *least one name* and let crowdworkers decide

if the phrases are positive, negative, or neutral, using majority voting with three judgments per phrase. The agreement rate, as measured through Fleiss' Kappa, is $\kappa = 0.329$. We discard phrases where no agreement could be reached. The final news dataset contains 1804 positive and 2402 negative sentences. We additionally discard neutral sentences, as we are focusing on binary sentiment classification in this work.

Table 6.5 provides a summary of the datasets. In the **Reviews** dataset, 21,861 out of 25,000 instances contain at least one person name, extracted through Named Entity Recognizer (NER) [JOP+].

Table 6.5. Number of positive, negative, total instances and number of instances containing at least one name for both downstream sentiment datasets.

	pos	neg	total	cont. names
Reviews	12,500	12,500	25,000	21,861
News	1804	2402	4206	4206

Table 6.6. Number of name sentences in the minority class for SGNS and DebiasEmb and mitigation effect for all base sentences using the Reviews and the News datasets for model training.

	Reviews			*News*		
Name sentence	**SGNS**	**DebiasEmb**	**Mitigation effect**	**SGNS**	**DebiasEmb**	**Mitigation effect**
[name] applies for asylum	512	0	512	1038	485	553
[name] is head of the state	440	125	315	1539	811	728
[name] is an actress	331	56	275	1475	550	925
[name] is an actor	331	56	275	1279	696	583
[name] runs for president	418	179	239	850	502	348
[name] runs for governor	418	179	239	2438	1028	1410
[name]	389	201	188	814	221	593
[name] is played by [name]	293	119	174	1455	659	796
Reviewed by [name]	368	196	172	147	58	89
[name] is president	184	25	159	1313	377	936
[name] is governor	184	25	159	1525	901	624
[name] is ceo	184	25	159	1386	66	1320
The movie features [name]	235	77	158	1491	892	599
[name] lives in the us	165	7	158	1605	893	712
[name] applies for a job	155	0	155	1569	835	734
The soundtrack is composed by [name]	153	25	128	1642	834	808
The name of the main character is [name]	96	10	86	1303	526	777
[name] is an us citizen	59	4	55	1338	547	791
[name] is a movie character	40	2	38	1517	784	733
[name] plays a role	51	46	5	699	493	206
Mean	250.3	67.85	182.45	1321.15	607.9	713.25

6.5.2 Name Sentences

For the downstream analysis, we use *name sentences* that are sentences that contain person names from the name list. We use the list of base sentences shown in the left column of table 6.6, which includes examples from both the Review and News datasets (e.g. *"Reviewed by Hugo"*), and an example that contains only a person's name. The tag [name] is replaced with each name from the name list, resulting in a total of 341,100 name sentences for the downstream analysis.

6.5.3 Text-level Classifier

The text-level classifier takes the concatenated embedding representations of all words in a textual snippet and outputs a class probability score for each sentiment class. We use a Neural Network with three hidden layers, dropout (dropout rate = 0.5) and learning rate lr = 0.001, and train for 50 epochs.

As training data we use the **Reviews** and **News** datasets. However, before training, we applied NER to extract person names and replaced all names from our name list with an "unknown" token, making sure that the sentiment of names is not directly influenced by the training data. E.g., if a name would appear more often in negative instances than in positive instances in the training data, the classifier would be likely to associate it with negative sentiment, independently of the bias in the word embeddings. Therefore, filtering is necessary to make sure we measure biases in the *input embeddings* and not in the *training data*.

6.5.4 Downstream Bias Measures

An unbiased classifier towards names should classify a sentence independently of the person names it contains. For example, both sentences *"**Obama** applies for asylum"* and *"**Trump** applies for asylum"* should be placed into the same class (e.g. positive). That is, the classifier with a probability greater than 0.5 will place them in the same class, thus, showing no bias. However, if the change of name from "Obama" to "Trump" results in the change of the class probability, then the classifier is biased.

Hence, a bias free classifier would label all the name sentences with the same class. Name sentences that are *split* across sentiment classes represent a classification behavior that is biased. Correspondingly, we measure the bias of a downstream classifier as the amount of name sentences that are labelled with the minority class. In more details, if 80% of the name sentences are labelled as positive, we will consider the classifier to be biased towards the remaining 20% of name sentences that are labelled as negative in the minority class.

6.6 Downstream Analysis Results

Table 6.7 shows the accuracy of the sentiment classifier using SGNS and DebiasEmb embeddings, and trained on the Reviews and News datasets. Similar to the word-level classifier, we note a small difference in terms of accuracy for DebiasEmb. The lower accuracy for News is explained by the smaller size of the dataset.

Table 6.7. Text-classifier accuracy for SGNS and DebiasEmb on both datasets.

	Accuracy	
	Reviews	News
SGNS	0.84	0.77
DebiasEmb	0.83	0.73

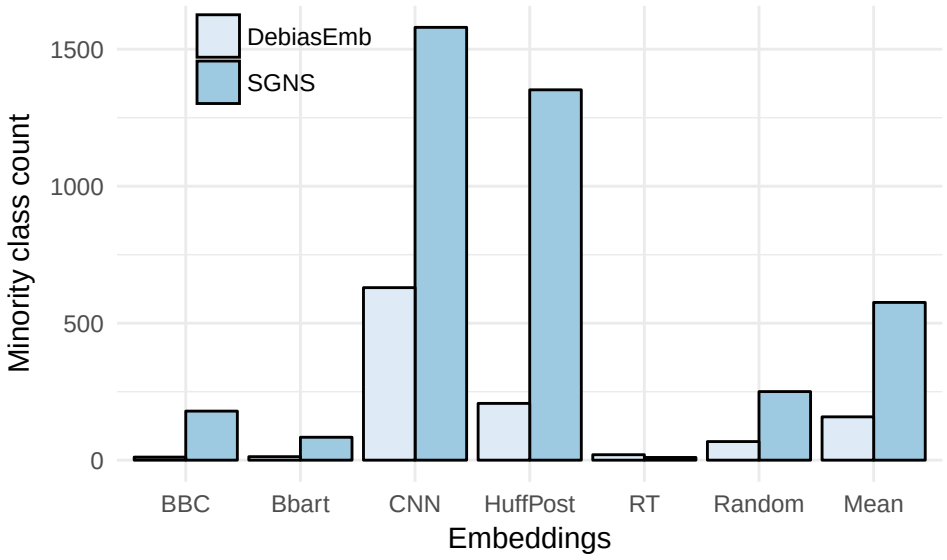
6.6.1 Highest and Lowest Ranked Names

Biased classifiers tend to give different sentiment labels and different class probability scores to sentences depending on the present names. Table 6.8 shows the highest and lowest probability scores that were given to names used in sentences by the SGNS model using embeddings trained on the *Random* news embeddings and reviews for the sentiment classifier. Name rankings were consistent across different base sentences. The ranking shows that a sentence including the name *Kam* is much more likely to be placed in the positive sentiment class compared to the same sentence containing the name *Callahan*.

Table 6.8. Highest and lowest class probabilities for the positive class averaged across all base sentences using the Random news embeddings, trained with SGNS on the Reviews dataset. A sentence including the name *Kam* is much more likely to be placed in the positive class, compared to the name *Callahan*.

Name	**High Pos. class prob.**	**Name**	**Low Pos. class prob.**
Kam	0.644	...	...
Holder	0.599	Gere	0.346
Weisman	0.596	Silk	0.342
Portillo	0.596	Nino	0.333
Fax	0.591	Valentine	0.304
...	...	Callahan	0.299

(a) Reviews



(b) News

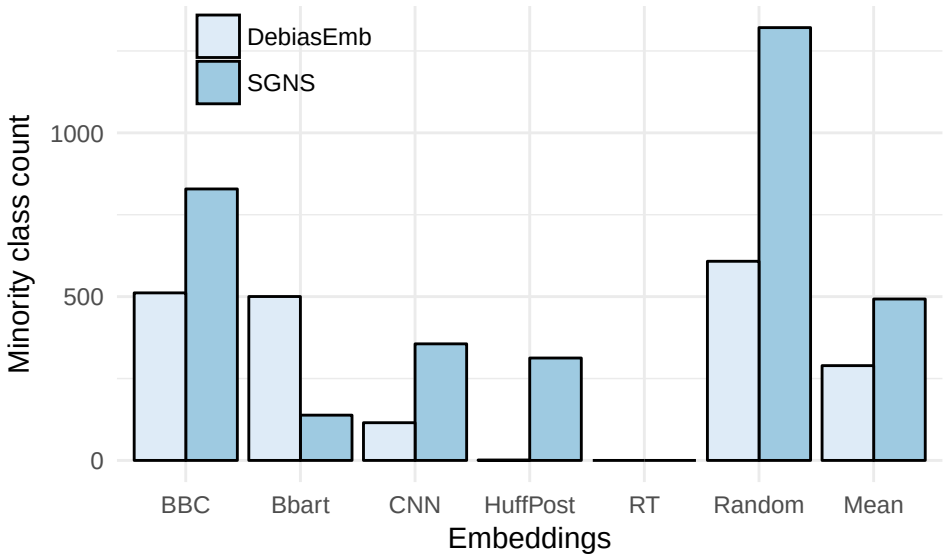


Figure 6.2. Average number of *name sentences* in the minority class for all embeddings trained with SGNS and DebiasEmb using the Reviews and the News datasets for model training. Ideally all variations of a name sentence are placed into one class, independently of the names used in the sentence, resulting in a low count for the minority class. DebiasEmb manages to reduce the mean count of the minority class compared to SGNS.

6.6.2 Class Label Distribution

Figure 6.2 shows the average number of name sentences that are placed in the minority class when leveraging the embeddings trained on the different news datasets using SGNS and DebiasEmb for the (a) Reviews dataset, and (b) the News dataset. In case of the Reviews dataset, DebiasEmb reduces the number of minority class instances compared to SGNS for all news embeddings except for RussiaToday where the value is already very low for SGNS (9.85) and only slightly increased (19.80) for DebiasEmb. On average, DebiasEmb reduces the number of minority class instances from 575.68 to 158.06, resulting in an average mitigation effect of 417.62, meaning that an average of 417.62 names have changed the final label from the minority to the majority class.

When using the News dataset for training the sentiment classifier, we observe similar behavior with a mean mitigation effect of 203.41 from 492.73 to 289.32. For the embeddings trained on the Breitbart news dataset, we observe a negative effect. This is the only situation for News where we observed that DebiasEmb increases the minority class count instead of lowering it. For RussiaToday, we find the special case of having all instances already placed into the majority class when using the standard SGNS embeddings. DebiasEmb correctly keeps the class distribution and does not negatively impact the resulting labels.

Table 6.6 depicts the number of name sentences placed in the minority class for each base sentence using Random embeddings trained with SGNS and DebiasEmb. We observe a positive mitigation effect for all sentences for both the Reviews (average mitigation effect: 182.45) and the News dataset (average mitigation effect: 713.25). There is a stronger mitigation effect for some base sentences compared to others, though the domain of the sentence does not seem to have an impact as there are higher and lower mitigation values for both movie review and news sentences. For the Reviews dataset two of the base sentences are completely placed into one class, independently of the inserted name.

These results show that DebiasEmb increases the homogeneity of the final downstream class labels for similar sentences containing different names. This effect is independent of the domain that the downstream classifier is trained on. The diversity of the outcomes when using different datasets for training word embeddings shows that the choice of training data impacts not only the bias of word embeddings but also the potential of the debiasing approach.

6.7 Discussion and Conclusion

Bias in word embeddings stems mainly from the training corpora. In the case of textual corpora, depending on how such a corpus is created, it may contain various types of bias, following the theory that language reflects the attributes and norms of a societal group [Hal70]. News in particular are mediators of *ideas, beliefs, ideology* [Fow13]. Thus, for events, political actors, or other event actors (e.g. individuals of a specific community group), a common scheme in news discourse is language that is loaded with subjective

words (i.e. positive or negative words) depending on the take a news source may have on a particular event.

Such insights are validated by several studies [BCZ+16, EG18], and similarly in this work too, we have seen that such bias stemming from the corpus, in our case sentiment bias associated with names, is present in word embeddings (SGNS), trained without taking any precaution in terms of such biases. Furthermore, approaches that rely on predefined lexicons containing gender specific words, against which target words are debiased, do not work due to the fact that such lexicons are incomplete and do not capture their proxies [EG18].

Considering such limitations, we proposed DebiasEmb, an approach for training and debiasing word embeddings based on the established method of skip-gram with negative sampling. Using an oracle sentiment classification model trained on words with explicit positive/negative sentiment, we address the problem of word proxies to consider the weights associated with the embedding dimensions and their values, rather than focusing solely on specific words. This allows us to take into account other word proxies in an automated manner. Apart from learning an accurate representation, DebiasEmb additionally aims at keeping each name embedding from a set of target names *indistinguishable* from both the positive and negative words and their proxies based on the oracle's classifier weights.

The evaluation results on word embeddings trained on varying news sources show that the sentiment bias associated with names is reduced significantly, while at the same time retaining high quality embeddings as measured by standard benchmarking results. Furthermore, on downstream tasks such as determining the sentiment of a text snippet, we showed that depending on the names present, models trained on various embeddings produce highly variable results. By using DebiasEmb the absolute majority of names is treated equally, resulting in a nearly 70% reduction of names that are discriminated by being categorized with a different label compared to the majority of the names.

7 Conclusions and Future Work

Bias on the Web leads to polarization and conflicts between opinion groups and negatively affects users in their free forming of a personal opinion. In this thesis, we addressed the problem of bias by focusing on three central aspects: detecting biased statements, understanding and mitigating worker bias, and debiasing word embeddings.

In Chapter 3, we presented an approach for detecting biased statements in text corpora such as Wikipedia. The approach relies on a set of features, including a bias word list that we obtained by identifying clusters of bias words in the word2vec vector space. Experimental evaluation shows that the model achieves a precision of 74% on a dataset of biased statements, annotated using crowdsourcing.

Based on these results, we introduced an improved, neural-based approach for biased statement detection in Chapter 4 that overcomes the limitations of a purely feature-based, bag-of-words approach. Making use of state-of-the-art recurrent neural networks with word embedding input, we showed that the improved approach is capable of capturing context dependencies in statements and therefore captures occurences of phrasing bias that are hard to identify for a purely feature-based approach. Using attention mechanisms and varying input types, such as Part-of-Speech tags and LIWC word functions, the approach increases the precision up to 92%.

In Chapter 5, we focused on bias being introduced by crowd workers in crowdsourcing tasks. Our study revealed biased behavior among workers with strong opinions on a given topic. We showed that this behavior affects the resulting ground-truth labels, impacting dataset creation for tasks such as bias detection or sentiment analysis. Our presented approaches succeed in mitigating this bias effect by creating awareness among workers and making use of the concept of social projection.

Finally, in Chapter 6, we addressed the problem of bias in word embeddings, focusing on the example of varying sentiments of names. We showed that name sentiment scores depend on the data that the embeddings are trained with and introduced an approach for debiasing word embeddings. An analysis revealed that the approach reduces the bias effect and positively affects the resulting labels of a downstream sentiment classifier. Furthermore,

these results impact the creation of a bias detection model relying on word embeddings.

The work presented in this thesis contributes towards identifying, understanding, and mitigating bias effects on the Web. While we provided solutions for multiple central problems related to bias, there still remain many aspects to be be addressed in future work.

Chapters 3 and 4 focus on identifying phrasing bias as a very common form of bias in text data on platforms such as Wikipedia. Though, other forms of bias do exist. We plan to further extend the bias detection approach on classifying different types of bias, including coverage/gatekeeping and focus/presentation bias. In this context, the inclusion of background information becomes a central aspect. We further plan to expand the understanding of bias and biased user behavior by focusing on the evolution of bias over time rather than viewing bias as a static phenomenon. This might also help to proactively prevent the creation of bias.

Finally, in cases where removing a biased statement is not the preferable option due to e.g. information loss, biased statements should automatically be replaced by unbiased equivalents. This involves the process of content creation. Bias flipping [CWAKS18] could be a first step towards addressing this problem.

Bibliography

[AG18] Alan Aipe and Ujwal Gadiraju. Similarhits: Revealing the role of task similarity in microtask crowdsourcing. In *Proceedings of the 29th on Hypertext and Social Media*, pages 115–122. ACM, 2018.

[BB04] Maxwell T Boykoff and Jules M Boykoff. Balance as bias: global warming and the us prestige press. *Global environmental change*, 14(2):125–136, 2004.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[BCZ+16] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

[Ben16] W Lance Bennett. *News: The politics of illusion*. University of Chicago Press, 2016.

[BEQ+15] Eric Baumer, Elisha Elovic, Ying Qin, Francesca Polletta, and Geri Gay. Testing and comparing computational approaches for identifying the language of framing in political news. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1472–1482, 2015.

[BES10] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204, 2010.

[BG+60] Roger Brown, Albert Gilman, et al. The pronouns of power and solidarity. 1960.

[Bib91] Douglas Biber. *Variation across speech and writing*. Cambridge University Press, 1991.

[BRA18] Dylan Bourgeois, Jérémie Rappaz, and Karl Aberer. Selection bias in news coverage: Learning it, fighting it. In *The International World Wide Web Conference 2018*, number CONF, 2018.

[Bro60] Róger Brown. Gilman. *The Pronouns of the Power and Solidarity*, 1960.

[BSK+13] Alexandra Balahur, Ralf Steinberger, Mijail A. Kabadjov, Vanni Zavarella, Erik Van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. Sentiment analysis in the news. *CoRR*, abs/1309.6202, 2013.

[BY18] Ricardo Baeza-Yates. Bias on the web. *Communications of the ACM*, 61(6):54–61, 2018.

[C+15] François Chollet et al. Keras. https://keras.io, 2015.

[CBN17] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[CH11] Ewa S Callahan and Susan C Herring. Cultural bias in wikipedia content on famous persons. *JASIST*, 62(10), 2011.

[CHMA10] Lydia B Chilton, John J Horton, Robert C Miller, and Shiri Azenkot. Task search in a human computation market. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 1–9. ACM, 2010.

[CIT16] Carrie J Cai, Shamsi T Iqbal, and Jaime Teevan. Chain reactions: The impact of order on microtask chains. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3143–3154. ACM, 2016.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[CWAKS18] Wei-Fan Chen, Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. Learning to flip the bias of news headlines. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 79–88, 2018.

[DCD+15] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 238–247, 2015.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DJL+18] Mark Díaz, Isaac Johnson, Amanda Lazar, Anne Marie Piper, and Darren Gergle. Addressing age-related bias in sentiment analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 412. ACM, 2018.

[DLMI13] Sanmay Das, Allen Lavoie, and Malik Magdon-Ismail. Manipulation among the arbiters of collective intelligence: How wikipedia administrators mold public opinion. In *22nd CIKM*. ACM, 2013.

[EdV13] Carsten Eickhoff and Arjen P de Vries. Increasing cheat robustness of crowdsourcing tasks. *Information retrieval*, 16(2):121–137, 2013.

[EG18] Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 11–21, 2018.

[Eic18] Carsten Eickhoff. Cognitive biases in crowdsourcing. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 162–170. ACM, 2018.

[FAA15] Besnik Fetahu, Abhijit Anand, and Avishek Anand. How much is wikipedia lagging behind news? In *Proceedings of the ACM Web Science Conference, WebSci 2015, Oxford, United Kingdom, June 28 - July 1, 2015*, pages 28:1–28:9, 2015.

[FDNML16] Liye Fu, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Tie-breaker: Using language models to quantify gender bias in sports journalism. *arXiv preprint arXiv:1607.03895*, 2016.

[FHW+02] Gavan J Fitzsimons, J Wesley Hutchinson, Patti Williams, Joseph W Alba, Tanya L Chartrand, Joel Huber, Frank R Kardes, Geeta Menon, Priya Raghubir, J Edward Russo, et al. Non-conscious influences on consumer choice. *Marketing Letters*, 13(3):269–279, 2002.

[FJPT14] Boi Faltings, Radu Jurca, Pearl Pu, and Bao Duy Tran. Incentives to counter bias in human computation. In *Second AAAI conference on human computation and crowdsourcing*, 2014.

[FMNA16] Besnik Fetahu, Katja Markert, Wolfgang Nejdl, and Avishek Anand. Finding news citations for wikipedia. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 337–346, 2016.

[Fow13] Roger Fowler. *Language in the News: Discourse and Ideology in the Press*. Routledge, 2013.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[GCGD17] Ujwal Gadiraju, Alessandro Checco, Neha Gupta, and Gianluca Demartini. Modus operandi of crowd workers: The invisible role of microtask work environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):49, 2017.

[GFH16] Ujwal Gadiraju, Besnik Fetahu, and Christoph Hube. Crystal clear or very vague? effects of task clarity in the microtask crowdsourcing ecosystem. In *1st International Workshop on Weaving Relations of Trust in Crowd Work: Transparency and Reputation Across Platforms, Co-located With the 8th International ACM Web Science Conference*, 2016.

[GFK+17] Ujwal Gadiraju, Besnik Fetahu, Ricardo Kawase, Patrick Siehndel, and Stefan Dietze. Using worker self-assessments for competence-based pre-selection in crowdsourcing microtasks. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(4):30, 2017.

[GG19] Hila Gonen and Yoav Goldberg. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*, 2019.

[Gil05] Jim Giles. Internet encyclopaedias go head to head, 2005.

[GKD14] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. A taxonomy of microtasks on the web. In *25th ACM Conference on Hypertext and Social Media, HT '14, Santiago, Chile, September 1-4, 2014*, pages 218–223, 2014.

[GKDD15] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1631–1640. ACM, 2015.

[GR09] Stephan Greene and Philip Resnik. 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, 2009.

[GS10] Matthew Gentzkow and Jesse M Shapiro. What drives media slant? evidence from us daily newspapers. *Econometrica*, 78(1), 2010.

[GVD+17] Maria Giatsoglou, Manolis G Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, and Konstantinos Ch Chatzisavvas. Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications*, 69:214–224, 2017.

[GYB17] Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. Clarity is a worthwhile quality: On the role of task clarity in microtask crowdsourcing. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, pages 5–14. ACM, 2017.

[GZ12a] Shane Greenstein and Feng Zhu. Collective intelligence and neutral point of view: the case of wikipedia. Technical report, National Bureau of Economic Research, 2012.

[GZ12b] Shane Greenstein and Feng Zhu. Is wikipedia biased? *The American economic review*, 102(3):343–348, 2012.

[Hal70] Michael AK Halliday. Language structure and language function. *New horizons in linguistics*, 1:140–165, 1970.

[Har54] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[HF18] Christoph Hube and Besnik Fetahu. Detecting biased statements in wikipedia. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 1779–1786, 2018.

[HF19] Christoph Hube and Besnik Fetahu. Neural based statement classification for biased language. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 195–203, New York, NY, USA, 2019. ACM.

[HFG18] Christoph Hube, Besnik Fetahu, and Ujwal Gadiraju. Limitbias! measuring worker biases in the crowdsourced collection of subjective judgments. 2018.

[HFG19] Christoph Hube, Besnik Fetahu, and Ujwal Gadiraju. Understanding and mitigating worker biases in the crowdsourced collection of subjective judgments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 407. ACM, 2019.

[HFJ+17] Christoph Hube, Frank Fischer, Robert Jäschke, Gerhard Lauer, and Mads Rosendahl Thomsen. World literature according to wikipedia: Introduction to a dbpedia-based framework. *arXiv preprint arXiv:1701.00991*, 2017.

[HJF18] Christoph Hube, Robert Jäschke, and Besnik Fetahu. Towards bias detection in online text corpora. 2018.

[HKH+14] Tobias Hossfeld, Christian Keimel, Matthias Hirth, Bruno Gardlo, Julian Habigt, Klaus Diepold, and Phuoc Tran-Gia. Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing. *IEEE Transactions on Multimedia*, 16(2):541–558, 2014.

[HLY+18] Jie Hu, Shaobo Li, Yong Yao, Liya Yu, Guanci Yang, and Jianjun Hu. Patent keyword extraction algorithm based on distributed representation for patent classification. *Entropy*, 20(2):104, 2018.

[Ho95] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[Hol68] David S Holmes. Dimensions of projection. *Psychological bulletin*, 69(4):248, 1968.

[Hol78] David S Holmes. Projection as a defense mechanism. *Psychological Bulletin*, 85(4):677, 1978.

[Hol79] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HTF08] Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. The elements of statistical learning. second. isbn-10: 0387848576; isbn-13: 978-0387848570, 2008.

[HTTA13] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Lam Ngoc Tran, and Karl Aberer. An evaluation of aggregation techniques in crowdsourcing. In *International Conference on Web Information Systems Engineering*, pages 1–15. Springer, 2013.

[Hub17] Christoph Hube. Bias in wikipedia. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 717–721, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[IEBGR14] Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *Proceedings of the Association for Computational Linguistics*, pages 1–11, 2014.

[IPW10] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[JOP+] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed <today>].

[JSPW17] Ayush Jain, Akash Das Sarma, Aditya Parameswaran, and Jennifer Widom. Understanding workers, developing effective tasks, and enhancing marketplace dynamics: a study of a large crowdsourcing marketplace. *Proceedings of the VLDB Endowment*, 10(7):829–840, 2017.

[JW19] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[JWN15] Ling Jiang, Christian Wagner, and Bonnie Nardi. Not just in it for the money: A qualitative investigation of workers' perceived benefits of micro-task crowdsourcing. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pages 773–782. IEEE, 2015.

[KKH15] Ece Kamar, Ashish Kapoor, and Eric Horvitz. Identifying and accounting for task-dependent bias in crowdsourcing. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[KM18] Svetlana Kiritchenko and Saif M. Mohammad. Examining gender and race bias in two hundred sentiment analysis systems. *CoRR*, abs/1805.04508, 2018.

[KNB+13] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013.

[KOS11] David R. Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 1953–1961, 2011.

[Kri11] Klaus Krippendorff. Computing krippendorff's alpha-reliability. 2011.

[KSV11] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk. In *AMCIS*, volume 11, pages 1–11, 2011.

[LGG18] Preethi Lahoti, Kiran Garimella, and Aristides Gionis. Joint non-negative matrix factorization for learning ideological leaning on twitter. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 351–359, New York, NY, USA, 2018. ACM.

[LPI12] Qiang Liu, Jian Peng, and Alexander T. Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 701–709, 2012.

[LPM15] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[Lyo70] John Lyons. New horizons in linguistics, volume. 1970.

[LZ12] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

[Mar17] Brian Martin. Persistent bias on wikipedia: Methods and responses. *Social Science Computer Review*, page 0894439317715434, 2017.

[MCQ08] Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403, 2008.

[MDP+11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[MH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[MRZ05] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[MS13] Catherine C Marshall and Frank M Shipman. Experiences surveying the crowd: Reflections on methods, participation, and reliability. In *Proceedings of the 5th Annual ACM Web Science Conference*, pages 234–243. ACM, 2013.

[MSC+13a] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[MSC+13b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[MWB+19] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. On measuring social biases in sentence encoders. *CoRR*, abs/1903.10561, 2019.

[NR16] Edward Newell and Derek Ruths. How one microtask affects another. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3155–3166. ACM, 2016.

[PFB01] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[PNI+18] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[Pre04] Dražen Prelec. A bayesian truth serum for subjective data. *science*, 306(5695):462–466, 2004.

[PSG08] Martin Potthast, Benno Stein, and Robert Gerling. Automatic vandalism detection in wikipedia. In *European Conference on Information Retrieval*, pages 663–668. Springer, 2008.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[R+00] Suzanne Romaine et al. *Language in society: An introduction to sociolinguistics*. Oxford University Press, 2000.

[RCJ+17] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 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, 2017.

[RDNMJ13] Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. Linguistic models for analyzing and detecting biased language. In *ACL (1)*, pages 1650–1659, 2013.

[RKK+11] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. *ICWSM*, 11:17–21, 2011.

[RYZ+09] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 889–896, New York, NY, USA, 2009. ACM.

[Sch99] Dietram A Scheufele. Framing as a theory of media effects. *Journal of communication*, 49(1):103–122, 1999.

[SF88] Gün R Semin and Klaus Fiedler. The cognitive functions of linguistic categories in describing persons: Social cognition and language. *Journal of personality and Social Psychology*, 54(4):558, 1988.

[SHC11] Aaron D Shaw, John J Horton, and Daniel L Chen. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 275–284. ACM, 2011.

[Sie15] Scharolta Katharina Sienčnik. Adapting word2vec to named entity recognition. In *Proceedings of the 20th nordic conference of computational linguistics, nodalida 2015, may 11-13, 2015, vilnius, lithuania*, number 109, pages 239–243. Linköping University Electronic Press, 2015.

[SPW+13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[Str87] Anselm L Strauss. *Qualitative analysis for social scientists*. Cambridge University Press, 1987.

[Sur05] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[TCL15] Oren Tsur, Dan Calacci, and David Lazer. A frame of mind: Using statistical models for detection of framing and agenda setting campaigns. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1629–1638, 2015.

[WGJS15] Claudia Wagner, David Garcia, Mohsen Jadidi, and Markus Strohmaier. It's a man's wikipedia? assessing gender inequality in an online encyclopedia. *arXiv preprint arXiv:1501.06307*, 2015.

[WJ11] Fabian L. Wauthier and Michael I. Jordan. Bayesian bias mitigation for crowdsourcing. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 1800–1808, 2011.

[WQS+15] Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *CoRR*, abs/1510.06168, 2015.

[WRTH15] Morten Warncke-Wang, Vivek Ranjan, Loren G. Terveen, and Brent J. Hecht. Misalignment between supply and demand of quality content in peer production communities. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 493–502, 2015.

[WWB+04] Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Computational linguistics*, 30(3):277–308, 2004.

[YRS10] 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. Association for Computational Linguistics, 2010.

[YYD+16] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[ZWY+19] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. Gender bias in contextualized word embeddings, 2019.

[ZY15] Honglei Zhuang and Joel Young. Leveraging in-batch annotation bias for crowdsourced active learning. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015*, pages 243–252, 2015.

[ZZL+18] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496*, 2018.