# N$^2$Bias: Debiasing Long Inputs One Span at a Time

**Dave Russell** and **Tara Verma**
UC Berkeley School of Information

## Abstract

Recent advancements in Natural Language Processing have achieved extremely strong performance on the detection of social bias in short-form text inputs. However, far less research has been done on accurately identifying bias past the phrase/sentence level. To address this gap, we leverage the Longformer architecture to introduce a scalable, NER-based bias detection approach for long-form text. Our findings indicate that NER is an effective approach for identifying approximate locations of bias in long-form text, but does not achieve the same level of accuracy as a binary classifier trained to identify whether or not input text contains bias.

## 1 Introduction

In recent years, there has been a significant amount of research on the existence and sources of social bias in NLP systems, including identifying bias in embedding spaces, LLMs, machine translation, and toxicity classification (Blodgett et al., 2020). One high-level takeaway from this body of work is that NLP models will reproduce social biases present in their training data. Humanities research has consistently pointed to the interconnectedness of social hierarchy and language (Romaine, 2000), and thus, it is crucial to acknowledge that the epistemic and representational harm perpetuated by biased models can and does translate into physical harm.

The primary complicating factor in this space is that social bias is elusive. We note that the existing body of research does not include a consistent definition, and spans allocational harm, stereotyping, spurious and questionable correlations, and differences in model performance between groups (Blodgett et al., 2020). To this end, Raza et al. proposed a framework for bias detection that treats bias as an entity, allowing for a less rigid definition that achieved state-of-the-art performance on short spans of text. A noted limitation of the Nbias framework however, is its inability to detect bias that spans multiple sentences (2024a). This presents both a problem and an opportunity. The problem is that social bias is not limited to specific harmful words or phrases, and often exists as subtler ideas that develop over the course of several sentences. The opportunity is to develop a scalable bias detection framework that performs well not just on sentence-level inputs, but also on paragraphs, and eventually documents.

This opportunity is important for several reasons. First and foremost, subtler forms of bias are both more common and harder to identify than overt bias, while still perpetuating the same level, if not greater harm (King and Jones, 2016). As humans, overt bias will immediately alert our ethical sensibilities and allow for a critical evaluation of the text. On the flipside, if bias develops over the course of several sentences, it may enter our worldview undetected. If we are able to see ahead of time that a piece of text contains bias, we can prime those same sensibilities and critical thinking skills that come naturally to us for overt bias.

By fine-tuning a transformer model well-suited for long form text on comprehensive bias detection data, we achieve strong performance on paragraph-level text inputs through both binary classification and Named Entity Recognition (NER) approaches to bias detection. We also develop a framework to evaluate NER performance against a binary classification baseline.

## 2 Background

Because bias in NLP is so pervasive, there are several avenues to address. One of the earliest breakthroughs in the field was the detection of bias and subsequent debiasing of word embeddings by removing stereotypical associations that appeared within vector representations of words (such as *receptionist* and *female*) (Bolukbasi et al., 2016).

Another area of focus is understanding how measurements of bias correspond with tangible harms, including stereotyping, disparagement, dehumanization, erasure, and quality of service, allowing researchers to make highly nebulous social tendencies more concrete, and providing a framework with which to measure harm (Dev et al., 2022).

More recent work has built upon the early foundations into more nuanced and elaborate forms of bias. A 2023 study developed models for interpersonal and intergroup relationships and emotions that capture the differences in the ways people speak to others who are members of in-groups and out-groups. For in-groups vs. out-group prediction, their model exceeded human performance (Govindarajan et al., 2023). There has also been significant work identifying bias within complex NLP tasks, including Named Entity Recognition (NER) and Relation Extraction (RE). One study evaluating NER systems found that the models performed differently at identifying names across demographic groups, achieving the highest performance on white male and white female names, and the lowest performance on Black female names. These differences in performance were not mitigated by using debiased embeddings (Mishra et al., 2020). Another study focused on understanding gender bias in RE found performance by gender varied greatly by the type of relationship, and provides a framework for evaluating bias in RE systems (Gaut et al., 2020).

Our approach to long-form bias detection is built primarily on the Nbias framework, the major contribution of which is a massive, semi-autonomously labeled dataset that spans multiple domains, and is well-suited to a broad range of NLP tasks. Since there are countless avenues for bias mitigation in NLP, we will focus, as Nbias does, on bias detection in text. Our work addresses a stated limitation of the Nbias framework, which is its relatively poor performance on long-form text by leveraging a Longformer-based architecture (Beltagy et al., 2020).

## 3 Methods

### 3.1 Architecture

Since state-of-the-art bias detection is currently on sentence and phrase-level inputs, we are defining 'long-form' as paragraph-length text, specifically inputs having 51-512 tokens. We trained a NER model to predict $[BIAS]$ entities spanning one or more tokens using BIO notation ("B" and "I", non-bias tokens as "O"). For our baseline, we are training a binary classifier that predicts whether an entire input contains bias. Both models leverage Longformer, a transformer model that excels on document-length inputs. Longformer is pre-trained on masked language modeling from a RoBERTa checkpoint, with a key modification: the attention mechanism.

One of the primary limitations that machine learning practitioners face when working with long-form text is that the self-attention mechanism operates in $O(n^2)$ time. For each of the $n$ query vectors in our input sequence, we must run computations, usually dot products, against all $n$ key vectors in the input sequence to achieve state-of-the-art performance for many context-dependent NLP tasks. As a result, input sequences are usually either truncated or partitioned. For long-form bias detection, this means that context spread across partitions may be lost, resulting in inaccurate classifications. Longformer can process inputs up to 4,096 tokens, allowing for a much broader context window, by using a modified attention calculation that scales linearly with input length. Longformer uses an "attention pattern" that operates similarly to a CNN filter, by stacking multiple layers of fixed-size attention windows surrounding each token that collectively incorporate information from the entire input, and operates in $O(w \times n)$ time, where $w$ is the window size. To allow for more flexibility, there are also a few task-specific input tokens (the locations of which are pre-selected and fixed) that undergo a global attention calculation, i.e. these tokens attend and are attended to by every other input token. In the case of classification, this is the $[CLS]$ token. The number of global attention tokens does not scale with $n$, and is small relative to $n$, and thus allows for linear attention (Beltagy et al., 2020). Since we are focusing on paragraph-level detection as the natural next step from sentences, we do not strictly need to use Longformer, but we are committed to for the sake of scalability.

### 3.2 Annotations & Data Handling

To train our NER model we undertook substantial data preparation. Here is a sample sentence, with the biased phrases bolded:

> So, yes, **you are criminals**. Bill Clinton on arresting, deporting **illegal aliens draining U.S**. resources. Hilary Clinton

> on **building a wall** and **deporting illegal immigrant children**.

The Nbias dataset (Raza et al., 2024b) provides this data in two forms: (1) the original sentence and (2) the biased words/phrases. We annotated the data in BIO notation, tagging non-entity tokens with “O”, beginning of entities with “B”, and, if applicable, succeeding entity tokens with “I” (see Table 1). We tagged in sync with tokenization to ensure sub-words were appropriately mapped (see *draining* in Table 1).

### 3.3 Baseline

Our baseline model is a binary classifier that determines whether or not the input text contains bias. The binary classifier has two hidden dense layers, with the first and second having 512 and 256 units, respectively, both with ReLU activation. Following each hidden layer is a 0.2 dropout layer for regularization. Finally, the output layer is a single-node dense layer with a sigmoid activation. We compiled the binary classifier using the Adam optimizer, trained with a learning rate of $5 \times 10^{-5}$ and binary cross entropy loss function over five epochs, optimizing for accuracy. Our binary classifier uses similar hyperparameters as the Nbias model, achieving comparable accuracy (93.1%). We can stand by the model as a proxy for state-of-the-art bias detection and therefore a reasonable baseline.

### 3.4 Named Entity Recognition

We trained the NER model in batches of four for a minimum of ten epochs and maximum of 20 epochs with a learning rate of $5 \times 10^{-5}$, employing early stopping with a patience of five epochs (the final training run exited at 16 epochs). To prevent overfitting, we applied a weight decay of 0.01 and incorporated a warm up phase of 100 steps, gradually ramping the learning rate, to prevent gradient explosion. During training we computed F1 score in a custom method and set it as the metric to determine the best model to keep.

Since NER is generally considered a more difficult task than binary classification, we do not expect to classify token-level entities with the same precision and recall as the binary classifier is able to classify the entire input span. This performance disparity suggests a practical implementation of the bias detection pipeline would be a “cascading ensemble,” with the binary classifier first predicting whether an entire input contains any bias. If so, the NER would then return the spans it identified as containing bias.

### 3.5 Experiments

In standing up the NER model we experimented with (1) initialization weights, (2) the number of hidden layers, (3) batch size, (4) learning rate, (5) dropout rate, (6) weight decay, and (7) warm up steps. The most substantial performance improvements came from initializing with weights that better represent the underlying distribution of BIO tokens. The majority of tokens in the dataset are “O”, not containing any bias. When we originally initialized agnostic to the distribution (equally weighting one-third each to O, B, and I) the model simply learned to predict “O” tokens, being the majority class. This led to an unusable model that reported high accuracy, since in fact most tokens were indeed “O.” The Nbias’s dataset is distributed approximately 93, 2, and 5 percent O, B, and I tokens, respectively. We wrote a custom loss class to initialize the class weights with a $[0.93, 0.02, 0.05]$ distribution.

### 3.6 Ablation Study

Our hyperparameter tuning (Table 2) focused on reducing the validation loss and increasing F1 score, which was our primary training parameter. [1]

### 3.7 Evaluation

In order to evaluate the success of our NER model, we leverage the error categories defined in the 5th Message Understanding Conference (MUC-5) (Chinchor and Sundheim, 1993), which are as follows:

- **Correct (COR):** the predicted bias spans match the labeled spans
- **Partial (PAR):** at least one predicted bias span overlaps with the labeled spans
- **Spurious (SPU):** the prediction contains a biased span, the label does not
- **Missing (MIS):** the prediction does not contain a biased span, the label does
- **Incorrect (INC):** the prediction contains bias spans that do not overlap with the labeled spans.

[1] Truncated to 5 of 27 samples run to focus on key learnings; see Notebook in GitHub repository for all samples.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| So | yes | you | are | criminals | Bill | Clinton | on | arresting | deporting | illegal |
| O | O | B | I | I | O | O | O | O | O | B |
| aliens | drain### | #####ing | U.S. | resources | Hillary | Clinton | on | building | a | wall |
| I | I | I | I | I | O | O | O | O | O | O |

Table 1: A tokenized input with BIO labels

| # | **Experiment** | **Val Loss** | **Val Accuracy** | **F1 Score** |
|---|---|---|---|---|
| 1 | N=1,000, 1 epoch, LR=0.00005 | 0.21251 | 0.94965 | 0.00000 |
| 4 | N=1,000, 5 epochs, LR=0.0001 | 0.32785 | 0.94119 | 0.22559 |
| 9 | N=2,000, 5 epoch, LR=0.00005 | 0.82062 | 0.85158 | 0.25722 |
| 22 | N=2,000, 13 epochs, LR=0.0001, initialized O/I/R distribution weights 93/2/5 | 0.67926 | 0.84284 | 0.49112 |
| 27 | N=9,590, 16 epochs, LR=0.00005, initialized O/I/R distribution weights 93/2/5, early stopping | 1.50071 | 0.54065 | 0.57710 |

Table 2: Initializing class weights according to the distribution, which overweights "O" tokens, was pivotal to improve performance.

Once our predictions are labeled, we calculate the following metrics, outlined in the MUC-5 paper, to understand model performance:

$$Error\ Rate = \frac{\frac{PAR}{2} + INC + SPU + MIS}{COR + PAR + SPU + MIS + INC}$$

$$Overgeneration = \frac{SPU}{COR + PAR + SPU + INC}$$

$$Undergeneration = \frac{MIS}{COR + PAR + MIS + INC}$$

Since binary classification and NER are entirely different tasks with different use cases, there is not an established way to evaluate our NER model against the baseline. The downstream applications of the NER model are far more robust and flexible than those of the binary classifier, as it will not only determine whether or not a text input is biased, but will also identify the specific spans within the input that contain bias. However, in order to reasonably leverage the results, we must be able to meaningfully determine how well the model performs at identifying bias. The default accuracy metric is not meaningful on NER, as the overwhelming majority of tokens are labeled “O” and predicted as such. Thus, to meaningfully compare performance against our binary classification baseline and run subgroup analyses, we devised the following accuracy formula, based on the error rate formula from the MUC-5 paper:

$$Accuracy = \frac{COR + \frac{PAR}{2}}{COR + PAR + SPU + MIS + INC}$$

## 4 Results

To understand the performance of our NER bias detection model, we begin by examining the distribution across the MUC-5 categories for our entire test set in Table 3:

| Output Category | **COR** | **PAR** | **SPU** | **MIS** | **INC** |
|---|---|---|---|---|---|
| Count | 758 | 1161 | 408 | 57 | 13 |

Table 3: Distribution of prediction categories

Thus, our derived overall test accuracy is 55.84%, compared to our baseline accuracy of 93.08%. To address our research question, we split our test sets into short and long inputs to determine how model performance changes across our input length categories (Table 4). For the full distribution of error categories by subgroup, see Table 6.

| | Accuracy | |
|---|---|---|
| Dataset | **Baseline** | **NER** |
| **Full** | 93.08% | 55.84% |
| **Short** | 95.78% | 62.70% |
| **Long** | 90.38% | 50.52% |

Table 4: Baseline calculated as traditional accuracy. NER calculated with derived accuracy formula.

Both models performed better on short inputs (50 or fewer tokens) than long inputs (51-512). The delta was wider on the NER model, with long inputs 12% less accurate than short.

Despite the relatively low overall NER accuracy, the model does have strengths. When examining the NER-specific metrics (Table 5), we note that across input lengths, we have extremely low undergeneration, as the model tended towards spurious predictions. The objective of this research is to meaningfully identify bias in paragraph-length text. To that end, undergeneration (the rate at which the prediction completely misses bias in the input text) would decrease model usefulness far more than any of the other metrics.

| | **Error Rate** | **Overgen.** | **Undergen.** |
|---|---|---|---|
| **Full** | 44.16% | 17.44% | 2.44% |
| **Short** | 37.30% | 19.15% | 3.36% |
| **Long** | 49.49% | 16.13% | 1.73% |

Table 5: Evaluation metrics by subgroup

We also note that we chose a conservative scaling factor of 0.5 for partial matches in both our accuracy calculation and our error rate. Thus, across all three categories andbut especially on long inputs (as indicated by the lowest overgeneration and undergeneration rates), our error rates are high because the majority of our predictions fall into the partial category. This suggests that our model is generally able to correctly identify the presence of bias and the approximate location in the input text, but misses the nuance around where the biased spans begin and end, moreso on longer inputs. This pattern is likely due to the Longformer attention mechanism. In order to scale, Longformer uses a reduced attention mechanism that captures high-level context, but largely misses token-level details.

Framed within our research objective, this is a success. NER evaluation occurs at the token level, which is inherently antithetical to the Longformer architecture, so we knew our evaluation metrics were unlikely to be extraordinary. Bias is nebulous, and the task of identifying specific spans of text with bias is immensely difficult, even for humans, as it is highly subjective. Thus, the detection of exact bias phrase start and end boundaries need not be perfect, because the subjectivity inherent in the task means that most users would likely disagree with the exact spans anyway.

## 5 Conclusion

Bias is a subtle and moving target. And detecting its presence spanning multiple sentences, paragraphs, and longer is a challenging and worthwhile endeavor. To this end, we developed a NER model, based on the Longformer framework, that processes up to 512 token-length inputs, which corresponds to multiple paragraphs. Our work contributes to the ongoing efforts to address the pervasive challenge of bias entity recognition by extending bias detection capabilities to long-form text. While our model faces limitations in pinpointing exact span boundaries, it reliably flags approximate locations, which constitutes practical and incremental progress. We hope that our approach inspires others to continue scaling bias-aware systems and implement responsible NLP models.

## Ethics Statement

Our model and results are for research purposes only, and should not be used without human-led review.

## References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings.

Nancy Chinchor and Beth Sundheim. 1993. MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

Sunipa Dev, Emily Sheng, Jieyu Zhao, Aubrie Amstutz, Jiao Sun, Yu Hou, Mattie Sanseverino, Jiin Kim, Akihiro Nishi, Nanyun Peng, and Kai-Wei Chang. 2022. On measures of biases and harms in nlp.

Andrew Gaut, Tony Sun, Shirlyn Tang, Yuxin Huang, Jing Qian, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2020. Towards understanding gender bias in relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2943–2953, Online. Association for Computational Linguistics.

Venkata Subrahmanyan Govindarajan, Katherine Atwell, Barea Sinno, Malihe Alikhani, David I. Beaver, and Junyi Jessy Li. 2023. How people talk about each other: Modeling generalized intergroup bias and emotion. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2496–2506, Dubrovnik, Croatia. Association for Computational Linguistics.

Eden King and Kristen Jones. 2016. Why subtle bias is so often worse than blatant discrimination. `https://hbr.org/2016/07/why-subtle-bias-is-so-often-worse-than-blatant-discrimination`.

Shubhanshu Mishra, Sijun He, and Luca Belli. 2020. Assessing demographic bias in named entity recognition.

Shaina Raza, Muskan Garg, Deepak J. Reji, Syed R. Bashir, and Chen Ding. 2024a. Nbias: A natural language processing framework for bias identification in text. *Expert Systems with Applications*, 237:121542.

Shaina Raza, Mizanur Rahman, and Michael R. Zhang. 2024b. Beads: Bias evaluation across domains.

Suzanne Romaine. 2000. *Language in Society: An Introduction to Sociolinguistics*. Oxford University Press, Oxford, UK.

# A Appendix

|  | COR | PAR | SPU | MIS | INC | Total |
|---|---|---|---|---|---|---|
| Long | 260 | 844 | 214 | 23 | 9 | 1350 |
| Short | 498 | 317 | 194 | 34 | 4 | 1047 |
| Total | 758 | 1161 | 408 | 57 | 13 | 2397 |

Table 6: Error counts by subgroup