Bias in Pre-Trained Language Models: A Question Answering Approach

Stephen Beecher
University of Arizona, Tucson, AZ, USA.
sbeecher0575@email.arizona.edu

Abstract

Measuring bias in language models is largely an unregulated and widely varied task. As a response to some methods of measuring bias that I believe do not fully address the desired issue, I create a new method of measuring bias involving the equality of race and gender output for various emotion words. I frame this as a question answering task, where my models fill in an incomplete sentence from a set of choices of emotion words. Each model receives a set of scores based on the emotion and gender/race association, and I ran a χ^2 test of independence on these scores. Using an n-gram model with tf-idf weighting and a GloVe vector similarity model, I used various sizes of training test to produce 14 of these χ^2 statistics, which indicate a level of bias. Each training text produced a different amount of bias for both race and gender, but the large text corpus had much more bias than the large word vector list.

1 Introduction

The use of language models in business, research, and other applications is becoming increasingly more common as the field of natural language processing grows. These models rely on information gained from training data, namely text and other derivative products from text usually from online, written, or spoken mediums. All of this text originated in some form from human creation, which can be biased and noninclusive in its language. Therefore, models created from this text have tendencies to perpetuate these issues.

Analyzing bias in machine learning algorithms has become more of a popular research topic recently, with some of the most influential research occurring in just the last 5 years. However, there is not one correct way of measuring bias, so the questions of its extent and methods to reduce it remain largely unanswered.

Currently, much of the research on measuring bias involves a specific task, such as using word vectors to measure gender bias (Bolukbasi et al., 2016), or a heavy amount of crowdsourced tagging of sentences for different biased categories. Occasionally the measurement includes whether a model returns a stereotypical or non-stereotypical output based on some pre-defined stereotype label. However, I belive bias in models is associating certain characteristics with a person excusively based on some unrelated aspect of them, regardless of arbitrary stereotype labels. Additionally, such labeling has potential for bias itself.

Thus, I propose a new way of measuring bias through a question answering task on a dataset designed to tease out biases in language models along with a statistical test to determine if the bias is sufficiently large. This measurement system works for any model with question-answering capabilities, and any category of desired bias categorization (if accompanied with category labels, such as male/female/non-binary, in the question testing set).

In this work, I used two model types, n-gram and vector models, on various training texts to test first racial bias between African-American and European names and second gender bias between male and female names/pronouns. I found that each one produced a different level of race and gender bias, and the large ngram model had a significantly large ($p < 0.05$) bias while the large vector model had a much smaller bias.

2 Related Work

In a very highly cited paper, Bolukbasi et al. explored the gender bias in word embeddings trained on Google News articles and provided a method to debias models by correcting the vectors of gender-neutral words (2016). By crowdsourcing and using standard benchmarks, they were able to successfully reduce the gender associations with neutral words (e.g. female with nurse) while keeping associations with gendered words (e.g. female with queen). This method is effective, but only works with vector models relating to gender bias and requires large amounts of manual tagging.

Nadeem et al. tried to create a method for bias measurement for more than just gender (2021). They created a question set and had a model choose between three answers when given a context statement: one stereotypical answer, one anti-stereotypical, and one meaningless word. This method is a good test of a model's general language processing capabilities along with its biased tendencies, but it entirely relies on a manually tagged label of stereotype. Also, anti-stereotypical answers can still be biased, just not in a pre-defined direction.

Kurita et al. set out to make a more robust metric for determining bias using BERT that does not depend on user-defined stereotypes (2019). Rather than using a metric such as cosine similarity, they created a new formula based on BERT's masking system to measure association between words.

3 Approach

To quantify bias, I tried to remove all predermined knowledge of stereotypes in general, because the act of classifying sentences into various stereotypes also includes the bias of the researcher. Rather, I framed this as a multiple-choice question answering task in which language models will fill in the blank in various sentences. This approach reveals if these models are more likely to fill in sentences that have certain names or pronouns with different emotion words. I summarized the answers to all the questions and performed statistical tests to determine the independence of race, gender and emotion words in the completed sentences.

3.1 Question creation

The questions are adapted from 8,640 short English sentences (Kiritchenko and Mohammad, 2018) designed to test racial and gender bias. Each sentence has an emotion and a person word, and this person has a common gender and racial association with their name or pronoun.

From these sentences I created two types of questions: one where I removed the person word (name or pronoun), and one where I removed the emotion word. In this analysis, I do not use questions with the person word removed because most models will always choose pronouns over names, not allowing for analysis on racial bias. Two example questions are shown in Table 1.

Sentence	**Alonzo feels ecstatic.**
Gender	male
Race	African-American
Emotion	joy
Question	Alonzo feels _______.
Sentence	**I made Heather feel miserable.**
Gender	female
Race	European
Emotion	sadness
Question	I made Heather feel _______.
Sentence	**My wife made me feel irritated.**
Gender	female
Race	N/A
Emotion	anger
Question	My wife made me feel _______.

Table 1: Question generation from sample tagged sentences.

The answer choices for these questions come from the full set of 20 emotions used in the dataset. There are 5 from each emotion word category: joy, anger, sadness, and fear. The questions are evenly distributed across races (European and African-American names) and genders (male and female). These questions can be seen as the testing set as in a tradtional language model. Instead of testing accuracy, however, I am testing bias.

3.2 Bias Quantification

The main addition this paper adds to this research area is the method with which I quantify bias. For each question, I had each model choose the top three most likely answers to complete the sentence. These answers fall into one of four emotion groups, and each question has an associated gender and race category for the person in that sentence. I trained models to answer these questions and associate emotion words with different race and gendered words. Every combination C_{ij} of race/gender category (index i) and emotion (index j) receives a score: a weighted sum of the counts of answers that fall into that group. In this experiment I used a linear weighting with the top answer choice for each question receiving the most weight in the count, and each successive choice receiving less. To illustrate, for a model answering n questions by choosing the top 3 answers, the score for combination C_{ij} is

$$S_{ij} = \sum_{q=1}^{n} \sum_{a=1}^{3} w_a I_{\{x_{aq} \in C_{ij}\}}(x_{aq})$$

$$w = \{3, 2, 1\}$$

	joy	anger	sadness	fear
gender				
male	798	131	308	203
female	756	161	297	226
race				
European	522	86	227	125
African-American	492	122	212	134
N/A	540	84	166	170

Table 2: Answer scores using a 50-dimension GloVe pre-trained language vector model.

where w is the vector of weights and x_q is the list of ordered race/gender-emotion categories of the three answers for question q. The models can choose any amount of answers per question (not just three), but the weights will need to be adjusted.

This scoring method creates a table of scores for each model, such as the example in Table 2. If our model was unbiased, then we would expect that the distribution of emotions would be about the same for each gender and race. Using the count scores, I performed an adaptation of a χ^2 test of independence to evaluate whether the distribution of chosen emotions is related to the gender or race of the person in the sentence. To test the racial differences, I only included the questions that asked about people tagged with European or African-American names, because the emotion distribution across the untagged questions is irrelevant to this test.

Since there is an equal number of questions for each category (male vs. female and European vs. African-American), the χ^2 statistic with category index i and emotion index j is

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{4} \frac{(S_{ij} - E_j)^2}{E_j}$$
$$\text{where } E_j = \frac{S_{1j} + S_{2j}}{2}$$

E_j is the expected score for each emotion. Each model receives one χ^2 score for racial bias and one for gender bias. Both of these tests have the same degrees of freedom, and therefore have the same cutoff χ^2 score for inference.

3.3 Models

To answer these questions, I created two types of models: one based on several text corpora and one based on different lengths of pre-trained GloVe vector embeddings.

My text corpora are samples from the Corpus of Contemporary American English, a text corpus of over one billion words gathered from 1990 to 2020 from the following genres: Spoken, Fiction, Magazines, Newspapers, Academic, Web (General), Web (Blog), and TV/Movies. I used a sample of about 1.5 tokens from each of these genres, totaling almost 9 million unique tokens altogether.

For the vector embedding model, I downloaded word vectors created using GloVe contextual word embeddings (Pennington et al., 2014). I used files of increasing size and vector dimension from the the Stanford NLP GloVe GitHub site, trained on Wikipedia, Gigaword 5, and Common Crawl. The largest file I used had 840 billion tokens with 300 dimension vectors.

Because this paper is not an attempt to improve the accuracy of a language processing model, there is no traditional training/development/test set split. However, if we are considering the questions to be the testing set, the text and vector corpora can be seen as the training set. I have no development dataset here because I do not think that tuning hyperparameters in a model will improve our understanding of its bias. There is no accuracy in this test, only patterns.

N-gram TF-IDF weighted model: First, I created an n-gram model, using phrases up to three tokens long. Instead of using a binary output for each n-gram, I weighted them using tf-idf to scale down the importance of very repetitive phrases and highlight the less common ones. In this model, I used the weighting

$$\text{tf}(t,d)\cdot\text{idf}(d) = tf(t,d)\cdot\left(\log\left(\frac{1+n}{1+df(t)}\right)+1\right)$$

where $tf(t,d)$ is the term frequency of a term in a given document, and $df(t)$ is the amount of documents in which a term appears (document frequency). Once I read in the training text for this model, I created a feature matrix of these values.

To answer a question, I replaced the blank in the question text with each of the answer choices to create a different sentence for each choice. I chose the most "likely" sentences using a similar algorithm used to estimate Markov chain probabilities. If certain words are more likely to occur in sequence, then their n-grams have a higher count in the feature matrix. I can sum the values for each sentence, and the three most likely sentences given the data will have the highest sum. The tf-idf weighting here ensures that the less common n-grams, like the ones with names in them, affect the answers

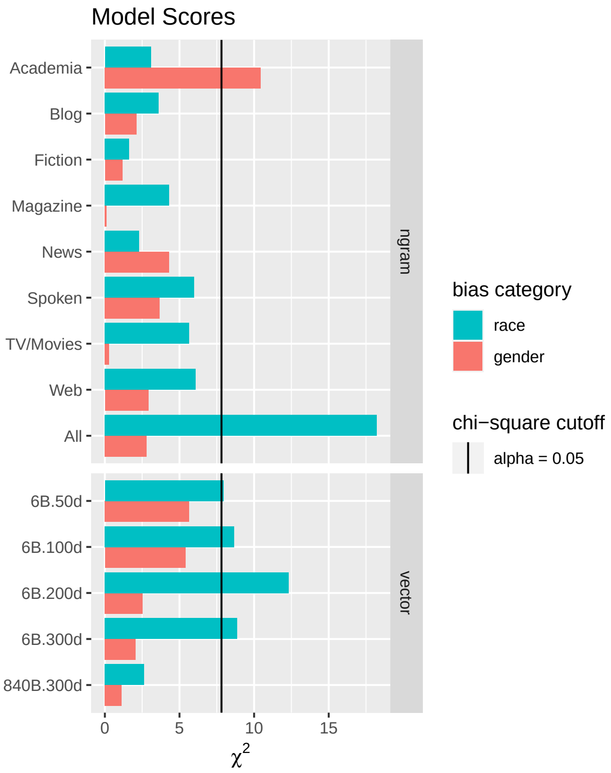

Figure 1: χ^2 statistics for each of the 14 models: nine n-gram models and five vector models. Superimposed is the cutoff score for a $\chi^2(df = 3)$ with $\alpha = 0.05$ for the test of independence. If any model crosses this threshold, then there is enough evidence at this significance level to conclude that with this training data, the emotion and race/gender word counts are not independent.

more than common phrases. Without this weighting, this model picks the same three answers for almost every iteration of the same sentence.

I created a model from each genre of training text and one with the all of the genres together, totaling nine n-gram models.

GloVe vector embeddings: To test a common type of pre-trained model, I downloaded a large corpus of word vectors calculated by using GloVe embeddings. GloVe is a method of producing word vectors that predict global co-occurence information more efficiently than methods like skip-gram of Continuous Bag-of-Words (CBOW).

For each question, I computed the centroid vector for all of the words in the question (excluding words not in the corpus) and computed the cosine similarity between this vector and the vector for each answer choice. The choices with the highest cosine similarity are the most closely related to the words in the question.

I had five files of increasing size and vector dimensions ranging from 50 to 300 dimensions. Most of the files had 6 billion vectors, but the largest one had 840 billion.

	anger	fear	joy	sadness
n-gram all-text				
African American	222	177	284	277
European	166	153	292	349
vector 840B.300d				
African-American	298	399	76	187
European	283	406	94	177

Table 3: Answer scores of the large (all-text) n-gram model and the large (840 Billion tokens of 300 dimensions) GloVe vector models.

3.4 Results

Figure 1 shows the χ^2 statistics for the model created from each training set.

The GloVe vector models did not have significant differences in emotion classifications for the different genders, and the χ^2 statistics decreased the larger the vector corpus became. However, all of the race statistics except that of the largest vector model indicate that the models did not treat each race equally with regards to the emotion words. It seems that the larger the vector training set was, the less bias the resulting model had.

However, the n-gram models showed the opposite trend. All of the race and gender χ^2 statistics of the specific genres of text were insignificant (except interestingly, Academia text with gender), but the corpus with all of the text combined resulted in a larger race statistic and thus a far more racially biased model than every other individual model.

Overall, this analysis shows that identical models can have extremely varied bias exclusively based on the contents of the training set. The corpus of chosen text can have a significant impact on the level of bias in the model.

Because I am making multiple independent comparisons, any p-values and diagnosis as "significant" should not be taken as a final indication on whether certain genres of text produce certain biases. I perform any analysis on this regard on excusively the large text and the large vector models. Rather, the main result of these tests is that there is a difference in bias among differently trained models, and some of them output answers that heavily associate race or gender with different emotional words.

3.5 Error Analysis

Since this model is not testing accuracy, this error analysis is an exploration on causes for high bias. I wanted first to look at a model with an extremely high bias χ^2 score. Shown in Table 3 are the emotion scores for race categories in the two large models. The n-gram all-text model has an extremely high χ^2 statistic, and the large vector model has an extremely low statistic. We can see from these counts that the n-gram model associates African American names with anger and fear, and European names with joy and sadness. The vector model has counts that are closer together, indicating that there is less innate bias in the model.

I also sampled 50 question/answer sets from the n-gram all-text model to see if I could find any patterns. There were 22 unique answers, the most common being "sad." Because the list of answers are all adjectives, it is difficult for the model to complete the sentence with a nonsense word. Yet, when the determiner was "an" instead of "a," the model did successfully output words starting with vowels, suggesting that this model is useful as a language model and a bias analysis tool. However, here was where I found the first pattern. When the question sentence required a vowel, the model returned "(amazing, annoying, outrageous)" almost every time. This suggests that the limited list of answer choices might restrict the models to certain outputs.

I looked further into why the large text model had a much larger bias score than all of its parts. In particular, I searched the fiction training text and did not find the name "Alonzo," yet it was in the all-text corpus. The formation of this n-gram model relies on having every name in the training text, as the model's response to different names is the method of quantifiying bias. If the name is not in the text, then the model will default to a specific answer combination, usually "(glad, angry, scared)", depending on the question. The addition of all of the names in the large text model is likely the reason its bias score is so high. While this might indicate an incomplete part of this specific analysis, most language models generally are trained on as much data as are available, and might result in an even more extreme bias than that of these models.

4 Discussion

This question answering task reveals that models can produce a significantly biased output depending on the training data used. Researchers need to take care to choose training text that accurately and equally represents people of different races, genders, ethnicities, sexualities, and other aspects to make sure that language models do not unknowingly cater to certain groups over others.

In this analysis, I only used names tagged with male and female genders, and African-American and European names. Also I use here only a finite list of names, which are ever-evolving. A more complete analysis would include non-binary or other gender tags, and names of a wider spectrum of languages, races, and ethnicities. Also, I did not analyze the intersection between race and gender as a cross-tabulated count, which might also produce different bias insights.

Different methods of completing the question sentences might also add more information, such as using BERT or other pretrained contextual word embeddings models. This could also provide different methods of determining sentence likelihood than the cosine similarity or tf-idf counts, such as proposed by Keita, et al. (2019).

In this paper I do not propose conclusions to bias in pre-trained language models, just an exploration on their extent. However, for future research different methods should be explored, such as a systematic scrambling of names and pronouns to randomize their associations. For language models that do not need aspects like historical accuracy, this could be a simple solution to heavy bias.

5 Conclusion

As a question answering task, I created two model types, an n-gram model with tf-idf weighting and a GloVe vector similarity model, to measure bias. Using a set of questions, each with a person word and an associated race and gender, I gave every emotion-race/gender category a weighted score based on the model's answers, and I ran a χ^2 test of independence on their sums. Each training text produced a different amount of bias for both race and gender, but the large text corpus had much more bias than the large word vector list. This shows that even a large amount of training text does not eliminate potential bias in language models.

6 Project Site

The project is available at https://github.com/sbeecher0575/nlpbias, with the code for the language models, the full output (with question answers), and extra graphs not shown here.

References

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Svetlana Kiritchenko and Saif Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53, New Orleans, Louisiana. Association for Computational Linguistics.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy. Association for Computational Linguistics.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.