Political Bias in BERT - Independent Study

1. INTRODUCTION

1.1. ABSTRACT

Transformers can learn universal language representations. They learn useful patterns and information from the dataset. Pre-trained models such as BERT and GPT-2 are trained on large quantities of unsupervised data. However, they can sometimes pick up undesirable and nuanced knowledge from the dataset they are trained on. For example, if the dataset that the model is trained on has a negative sentiment towards a particular entity, the model can learn that and inherit this negative sentiment. This gives rise to bias in these pre-trained models which can be in the form of gender or toward particular political entities. If not tackled early on, this bias could cause serious problems when these models are deployed in the real world. As Natural Language Processing (NLP) techniques become more and more popular in the world, it is more important to address this kind of social & political bias.

In this project, I studied distilBERT, a pre-trained language model, and examined if there exists any inherent political bias. The main aim of this study is to check if BERT is biased toward a particular political entity such as Democrats or Republicans. After fine-tuning BERT on a set of political news articles, I tested the predictions of that model on a set of validation sentences that contain groups of covid-related topics. Moreover, I compute the sentiment of the MASK word being predicted by the model, to check if the model is more positively or negatively inclined towards different political entities - Democrats or Republicans.

1.2. HYPOTHESIS

A pre-trained BERT model fine-tuned on a dataset consisting of news articles would inherit the political bias existing in the articles.

1.3. POLITICAL BIAS

News publications usually have an inherent bias in them. Different publications tend to favor particular political parties. The bias that exists in pressing public concerns such as gun violence, abortion rights, mask mandate, covid issues, etc can be considered as political bias.

Enlisted below, are a few examples being predicted from the model that was

inclined towards a Left-leaning dataset and a comparable Right-leaning dataset. More information on these datasets can be found in Section 2 a - Training dataset.

I fine-tuned the BERT model on a left-leaning dataset (containing articles published by left-leaning publishers) and on a right-leaning dataset (containing articles published by right-leaning publishers).

The following are example sentences to showcase how the predictions are different when a particular political entity is considered.
Example -
Republicans decided to [MASK] their mask during the press conference.
For a right-leaning model: [MASK] is the predicted word which would be "remove"

Democrats decided to [MASK] their mask during the press conference.
For a left-leaning model: [MASK] is the predicted word which would be "wear"

California [MASK] Republicans for medical supply shortages.
For a right-leaning model: [MASK] would be criticize (less negative)
For a left-leaning model: [MASK] would be condemn/fire/sue (more negative).

2. DATASETS

2.1. TRAINING DATASET

The base BERT model was fine-tuned on a set of approximately 144,000 news articles from 15 mainstream news outlets such as CNN, MSNBC, Fox News, etc. Due to the GPU requirements on Google Colab, I downsampled the dataset and trained on 10,000 articles that contained an equal number of Left and Right-leaning articles. These news articles were divided into a set of left-leaning and right-leaning datasets with equal articles being taken from each category. The publisher's political leaning was referenced from AllSides [4], as shown below:

- Left-leaning (~65k): ABC News, CBS News, CNN, MSNBC, Washington Post, Al Jazeera English, MSNBC, Google News, Buzzfeed
- Right-leaning (~44k): Fox News, Breitbart News, The Washington Times

The main attributes of the dataset are:

- 'Publication': The media outlet that published the news article.
- 'Text': The actual text content of the news article.

2.2. VALIDATION DATASET

To create a validation dataset, I randomly extracted around 1000 sentences containing the words 'republicans' and 'democrats' from online news-related articles. They contain a mixture of left and right-leaning articles. After extracting the 1000 sentences, I made sure the same sentences are not present in training data to avoid overfitting and to ensure the model is tested on unseen data.

I grouped all the sentences into 6 main topics from a topic modeling approach [3] that was originally derived from covid related tweets [3]. After experimenting with different topic modeling approaches like BERTopic and LDA, I decided to use a keyword search approach in a semi-automated way. The 6 main categories were as shown below and their definitions are provided in the appendix:

- ***cases_deaths***
- ***economy_education_impact***
- ***politics***
- ***preventive_measures***
- ***virus_spreading***
- ***vaccine***

The keywords used for the 6 categories were as follows:

Group	Keywords used
cases_deaths_keywords	['surge', 'high', 'case', 'wave', 'cases', 'deaths', 'death']
economy_education_impact_keywords	['student', 'child', 'shut', 'life', 'work', 'year', 'business', 'businesses', 'shutting', 'market', 'economy', 'worker', 'employer', 'employee', 'unemployment', 'unemployed']
politics_keywords	['election', 'government', 'national', 'announce', 'official', 'live', 'host', 'american', 'response', 'crisis', 'plan', 'president', 'district', 'state', 'deal', 'conduct',

	`'issues', 'fundraising', 'politicizing', 'governor']`
preventive_measures_keywords	`['plan', 'stop', 'urge', 'break', 'stop', 'spread', 'measure', 'extend', 'ease', 'lift', 'restriction', 'lockdown']`
virus_spreading_keywords	`['outbreak', 'virus', 'spread', 'warn', 'global', 'emergency', 'infect', 'infection', 'risk', 'spreading', 'rise', 'rising']`
vaccine_keywords	`['vaccine', 'pfizer', 'dose', 'dosage', 'receive', 'astrazeneca', 'supply', 'jab', 'drug']`

After grouping the validation sentences using a keyword search technique, I manually went through the groups and annotated them. Out of the 1000 sentences, I agreed on the topic generated for 669 sentences and disagreed with the generated topic for 339 sentences. The other rater agreed with 455 sentences and disagreed with 545 sentences from the auto-generated topics. Further, he agreed with 707 of my 1000 sentence labels, which gave an **interrater agreement of around 71%.** We together agreed on 415 sentences being correctly grouped and used these sentences as our validation dataset.
The hierarchy of the groups is shown in the below table.

The number of validation sentences from each group was as follows:

Group No.	Topic	Number of sentences
0	cases_deaths	48
1	economy_education_impact	69
2	politics	218
3	preventive_measures	30

4	virus_spreading	31
5	vaccine	19

3. METHODOLOGY

In this study, we explore a common method called domain adaptation that involves fine-tuning a masked language model. To start with, we pick the distilbert-base-uncased pre-trained model for fine-tuning. We load our dataset of political news articles and publications and separate them out into left-leaning and right-leaning articles based on the following publications:

- Left-leaning (~65k): ABC News, CBS News, CNN, MSNBC, Washington Post, Al Jazeera English, MSNBC, Google News, Buzzfeed
- Right-leaning (~44k): Fox News, Breitbart News, The Washington Times

Due to the GPU requirements on Google Colab, I downsampled the dataset and trained on 10,000 articles that contained an equal number of Left and Right-leaning articles. I split the dataset into a train-test split and another unlabelled split for the unsupervised data. The training dataset contains 80% of the data, the test dataset contains 10% of the data, and the unsupervised dataset contains the remaining 10%.

To make sure the data is in the correct format to train, I performed some preprocessing of the data. I concatenated all the example sentences and then split the entire corpus into chunks of equal size. This is done to make sure individual texts are not truncated because of being too long, which could lead to loss of useful information. The next step is to tokenize the entire corpus and then group these tokenized texts into chunks of size 128 (this size has to be smaller than the maximum context size for distilbert - 512). I then concatenated the split chunks into equal-sized blocks and also simultaneously dropped the last chunk if the size is significantly smaller than the other chunks. This approach of grouping and chunking the news articles will produce more examples than originally in the train and test dataset since we now have a few contiguous tokens that span multiple examples from the initial corpus.

After the dataset has been properly chunked, we perform the masking step. We insert [MASK] tokens at random positions in the input while fine-tuning using a data collator. Transformers have a dedicated DataCollatorForLanguageModeling where we can pass the tokenizer and a probability percentage (15%) that decides what percentage of the tokens we should mask. This data collator ensures that the [MASK] token will be randomly inserted in various locations of our text for each batch.

After the [MASK] tokens have been added to our dataset, we now train our model and set our hyperparameters. We use the AdamW optimizer with a linear learning rate decay in addition to *distilberts* inbuilt loss function. To check if our dataset has actually improved the original model, we compute the perplexity of the model - both after and before training. Before training, we get a perplexity score of 23.83. A lower perplexity score means a better language model and we can see our original model (before training) has a high score. After fine-tuning, we get a score of 14.12. This clearly shows that our model has learned something new from the dataset.

We now have a fine-tuned distilBERT model that is ready to predict [MASK] token words in a sentence. We use this trained model on our set of validation sentences.

4. RESULTS

BERT was used to predict masked tokens from our validation dataset that were denoted as [MASK]. These masked words were mainly action words so that we can gauge if there is a visible difference between the predictions for Democrats and those for Republicans.

To quantify our bias, we replicate the association formula used by Kurita et al. - [1]. We compute the association between two targets (democrats and republicans) with the attributes (co-related words). For example, to compute the association between the target Democrats and the attribute celebrated, we input the following sentence to BERT "[MASK] celebrated the victory of Obama in 2012", and calculate the probability of the [MASK] word being "Democrats." Similarly, to calculate the association for Republicans, we calculate the probability of the [MASK] word being "Republicans."
In addition to this, we need to re-weight this probability using the Prior probability of the word celebrated, i.e. we need to check the probability of the sentence predicting the word celebrated when it contains the word "Democrats". Similarly, we need to check the probability of the sentence predicting the word celebrated when it contains the word "Republicans". We then take the log of the division of P_{target} with P_{prior} . This value is the final value we use to measure Political bias.

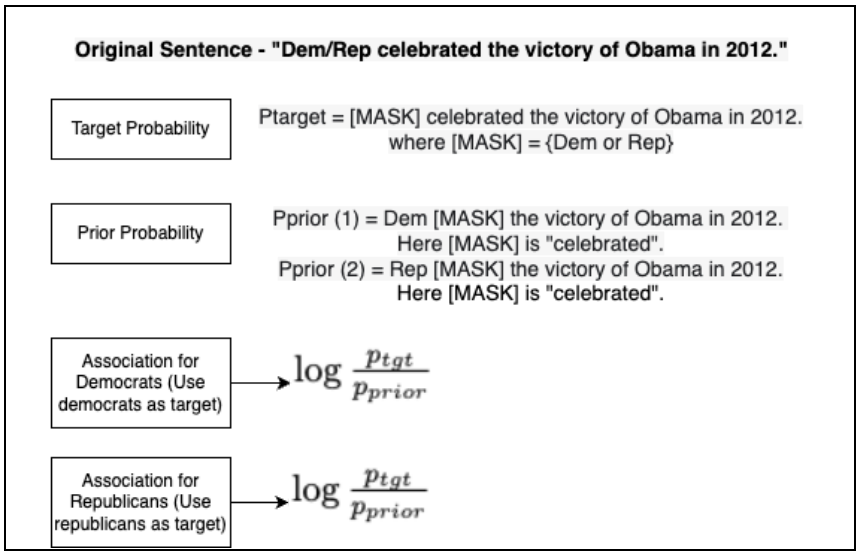

Fig 1 - Formula used to measure bias

		Groups						
Model	**Target**	**0**	**1**	**2**	**3**	**4**	**5**	**Total**
Left	**Dem**	0.1285	0.0604	0.0568	0.1492	0.0465	0.0297	0.4711 (1)
	Rep	0.1089	0.0654	0.0818	0.1119	0.0375	0.0091	0.4146 (2)
	Sent Score	**0.0595**	**0.2975**	**0.0549**	**0.0193**	**-0.0497**	**-0.0011**	**0.3805 (3)**
Right	**Dem**	0.1612	0.0998	0.1053	0.1815	0.0841	0.0638	0.6957 (4)
	Rep	0.1264	0.0865	0.1036	0.1264	0.0566	0.0273	0.5268 (5)
	Sent Score	**0.1210**	**0.2726**	**0.1019**	**0.0195**	**-0.0928**	**0.0146**	**0.4369 (6)**
Base	**Dem**	0.0958	0.0452	0.0568	0.1175	0.0167	-0.0209	0.3110 (7)
	Rep	0.0965	0.0574	0.0818	0.0896	0.0240	-0.0224	0.3269 (8)
	Sent Score	**0.0585**	**0.1081**	**-0.0111**	**0.0244**	**0.0597**	**-0.0052**	**0.2344 (9)**

5. DISCUSSION

The main observation noticed was that all groups have a higher association score towards

republicans compared to democrats. This higher association score means that the model has a higher probability of predicting republicans compared to democrats. Hence, the base BERT model has a higher probability of predicting the word ‘republicans’ compared to the word ‘democrats’ when there are covid related data.

The total summation value of all groups is higher for republicans. The probability of predicting the word republicans is higher which further shows the base distilBERT model has a higher preference or is more inclined towards favoring republicans for covid related data. For a neutral sentence, the base model would predict republicans with a higher probability over democrats.

Moreover, we can also see that the total scores for both the left-leaning and the right-leaning models for the word “democrats” and the word “republicans” is much higher. The scores are the summation of the association values for all 6 groups of covid related topics. This shows that the model has learned more information about these two political entities. The difference between the total values of the democrats association score (1) with the republican association score (2) in the left-leaning model (in blue) is 0.06. The difference between the total values of the democrats association score (4) with the republican association score (5) in the right-leaning model (in red) is 0.17. This difference increased the association scores of the right-leaning model which can be attributed to a higher score for republicans. This further proves our original hypothesis that training a model on a news-related dataset, inherits political bias from the dataset.

In addition to this, for Group 3(preventive measures), the association values are higher for democrats compared to republicans. This shows that when the sentences are talking about the prevention of coronavirus, the model predicts democrats with a higher probability than republicans. This could mean that the model assumes democrats are doing more work regarding preventive measures for covid compared to the republicans.

In terms of sentiment scores, we see that the sentiment values of the base model (9) are 0.2344 is much lower than the ones of the fine-tuned models 0.3805 (3) and 0.4369 (4).
This shows that the fine-tuned models have more extreme sentiments attached to the predictions. Moreover, the high sentiment score of the right-leaning model shows that the right-leaning BERT model predicts more extreme words as the [MASK] word compared to the left-leaning BERT model.

6. CONCLUSION

In this study, we have established that the base BERT model does have bias attached to a particular entity. By bias, we mean that the BERT model portrays a political entity with

more positive sentiment compared to another political entity. Bias in pre-trained NLP models can arise from different aspects such as the training data, the pre-trained embeddings, or through fine-tuning the model. Through this work, we demonstrate that bias does arise through training data and fine-tuning a model. Our results show that training the model on a dataset that is leaning towards a particular political party tends to learn those characteristics and behave similarly while making new predictions for that same political entity. A similar methodology can be adapted to study other social biases in more detail.

The specific details of the project or the steps needed to recreate my findings can be found in this GitHub repository.[5]

7. REFERENCES

[1] - 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.

[2] Fine-tuning a masked language model from Hugging Face, https://huggingface.co/course/chapter7/3?fw=tf

[3] F. B. Oliveira, A. Haque, D. Mougouei, S. Evans, J. S. Sichman and M. P. Singh, "Investigating the Emotional Response to COVID-19 News on Twitter: A Topic Modeling and Emotion Classification Approach," in IEEE Access, vol. 10, pp. 16883-16897, 2022, doi: 10.1109/ACCESS.2022.3150329.

[4] Media Bias Ratings from AllSides, https://www.allsides.com/media-bias/ratings

[5] GitHub repository containing notebooks, datasets, and a complete set of results in a structured manner, https://github.com/rahil1304/political-bias-in-BERT

APPENDIX

cases_deaths: Any sentence that talks about the rising number of cases/deaths every day, the covid wave, and how the cases are increasing/decreasing.

economy_education_impact: Any sentence that talks about the impact of covid on the economy

or education sector, how it affected students, employees, and the way the market reacted to the closure or opening of business.

politics: Any sentence that talks about the election, the government, or the measures taken by any authority due to covid that had some political significance. Any critique from one political party/person to another. Talks about the budget, bill, governor, state, etc. Talks about CDC guidelines being followed/not followed. “Response to pandemic” talks between parties.

preventive_measures: The sentences that talk about the preventive measures to stop the increase in covid cases. The plan to reduce the cases, the lifting or enforcing/extension of lockdown.

virus_spreading: The sentences that deal with the spreading of the virus, the outbreaks happening both globally, and the infection rise happening due to covid spreading.

vaccine: Any sentence that talks about the vaccine for treating covid, the names of the vaccines, the dosage level, the supply or increase of vaccines, etc.