CHARM: Multivariate Headline Tuning Along Predefined Feature Structure

Sodi Kroehler
University of Pittsburgh, PA, USA
sek188@pitt.edu

Abstract

Algorithmically reversing or neutralizing bias in written media has many uses, yet has historically only been done in a polar fashion. However bias is almost always multifaceted, and there may be certain types of bias that are more acceptable than others. Thus there is a potential value in being able to create generative algorithms that generate text along pre-defined bias parameters. Building off the work of (Lu et al., 2022), I make available several modifications tho their model that helps to tune sentences along these predefined parameters while retaining the high performance that LLM's exhibit. Additionally, I show that this method also works to develop other stylistic changes, such as average word length.

1 Introduction

Along with the very high performance of LLM's in the recent years has been a growing trend to explore fine-tuning them. Previous attempts have shown successful results in reducing bias (Zhang et al., 2024), introducing author style (Bhandarkar et al., 2024) and reducing toxicity (Lu et al., 2022).

However, along with reducing bias holistically, there has been some research in altering rather than attempting to remove bias. The motivations for doing so are two-fold. Firstly, there are numerous different forms of bias, and it may be difficult to tune a model to reduce all forms equally. There may be use cases wherein a certain type of bias - say political leaning - is much more acceptable than another type (e.g. racist ones).

Secondly, there are some situations where entirely removing a certain kind of bias is actually not preferred. Political leanings are a prime example - studies have shown many people actually prefer to consume media from news outlets that are traditionally heavily biased (Liu et al., 2021b). Unilaterally reducing bias may incidentally also reduce viewership. Liu et al. identify changing political polarity as a possible method for reducing the effects of echo chambers and exposing news consumers to alternate viewpoints/stories. A use case thus emerges for news to be expressed in different viewpoints concurrently, a task where an algorithmic solution would be ideal. This was a primary argument for Chen et al. who created a model to "flip" the bias of news headlines, prompting numerous developments since.

To further this effort, I utilize the fine tuning method developed by Lu et al.. They showed successful results in minimizing toxicity and other tasks by training the model based on pre-pending tokens to generated utterances depending on a reward function. I extend their method by introducing a new reward function CHARM to express multiple features and show that this method is capable of maximizing model performance along different metrics satisfactorily. This enables tuning across N-axes, resulting in a more fine-grained control of the biases that a generative LLM might produce in it's generations. Given that it can adjust any number of features, it also allows for various personality or other stylistic flairs to be developed at the same time.

2 Related Work

2.1 Reducing Political Polarity

Chen et al. is certainly a forerunner in this space. They trained a transformer style model on ALLSIDES headlines, and were able to fairly consistently "flip" the bias of news headlines. However, they did notice this was somewhat at the expense of the content descriptions (Chen et al., 2018).

Liu et al. extended this with a very in-depth study of types of bias as well as the ability to translate entire news articles instead of just headlines. They also utilize a different model structure which they claim is able to encode both stylistic and content information, resulting a more accurate bias "flipping"(Liu et al., 2021a).

Lu et al. is also useful to mention, given that my method is an extension of this work. This is also the first work that I could find that tunes a LLM instead of attempting to build a transformer outright. This has many advantages in the content space, but also leverages the abilities that the LLM implicitly has in other spaces(Lu et al., 2022).

I could not find any other papers that dealt with this space directly.

2.2 Bias Definition

Bias in news media, and elsewhere, is difficult to define. Groeling categorizes media bias into selective bias and presentation bias, where selective bias skews the choices of what events to cover, and presentation bias skews the style of how to report the news.

Eberl et al. extends this to a three point scale: visibility, tonality and agenda. Visibility bias covers the political actor's salience, with higher-profile actors receiving greater attention than lower, on average. Tonality bias here covers the style of the media's author - roughly corresponding with Groeling's "presentation bias" - and finally agenda bias covers "the extent to which parties address preferred issues in media coverage" - roughly corresponding to Groeling's "selective bias". Other studies have displayed other categorizations(Puglisi and Snyder, 2015) (Lichter, 2014). However, there seems to be a general agreement that there is a bias involved in style, and it is this form of bias that this paper seeks to mitigate.

More recently, (Wessel et al., 2023) introduces a media bias benchmark, and separates media bias into a 9 element scale. With a rough similarity to Groeling, "presentation" bias is further subdivided into "linguistic", "text-level context bias", "reporting-level context bias", "hate speech", "racial bias" "gender bias" and "political bias", and "fake news" and "cognitive bias" falling more into "selective bias". However, Wessel et al. does not rely on Groeling or Eberl et al.'s classifications, so this distinction is not entirely accurate.

In this paper, I do not specifically measure along any of these scales, although I would have liked to. Instead I rely entirely on a political polarity measurement, which could somewhat be grouped under Wessel et al.'s "political bias" bias type, and deal only with that. Future developments on this paper would surely benefit by measuring the CHARM model on the rest of their benchmarks to further elucidate its performance in reducing other types of bias other than political.

2.3 Stylistic Tuning

Research has also gone into tuning LLM's to affect various types of stylistic changes, which in this case is very similar to what I am trying to do. In 2022, Syed et al. trained an transformer model on the author corpus as well as additional general data, and achieved mildly successful results. However, this required a very large amount of author-written material in order to successfully train the model. In 2024, (Bhandarkar et al., 2024) extended this work by prompt engineering on several state-of-the-art LLMs, again with some success. Bhandarkar et al. also state that no further research on tuning LLM's with reference to generated style, and I could not find anything to refute this.

Both of these, and others (see (Fu et al., 2018) (Lample et al., 2019)) rely on the assumption that style is distinctly separate from context. This is not shared universally. (Shen et al., 2017) incorporate adversarial networks into cross-aligned autoencoder architecture, encouraging the system to learn the separate style and the content distribution. (John et al., 2018) artificially divide the latent representation into style and content space, and design auxiliary multi-task loss and adversarial loss, enforcing the separation of style and content latent spaces when training an encoder-decoder network. However, in this paper I leave content similarity measurement to future work, thus relying on an assumption that the style is disentangled from the context.

3 Method

3.1 Data

- **REALTOXICITYPROMPTS** In accordance the (Lu et al., 2022) method, I used their released dataset from REALTOXICITYPROMPTS for development. However, since I do not work on minimizing toxicity in this paper, I did not use it for any testing.

- **ALLSIDES** I mainly use the dataset released by Chen et al., collected from the Allsides web portal [1].

[1] available from Chen at https://webis.de/data/corpus-webis-bias-flipper-18

I had hoped to measure the toxicity reduction using the REALTOXICITYPROMPTS dataset, to show comparable results to the unmodified Quark model, but the dataset was too large and my computer was unable to train the model in a satisfactory amount of time. Thus, I am only mentioning this dataset as a courtesy since I used it heavily in development. However, I would like to note that the PerspectiveAPI, which was utilized as the reward function in the original Quark model, is absolutely compatible as a feature in the CHARM model, and I see no reason why an experimenter would not achieve exactly the same performance levels as did the Quark team if they were to use it as such.

As the ALLSIDES data was formatted sufficiently different, I did some manual conversions to allow the Quark model to be able to read it. Specifically, I started by setting the headline to be the "text" of the initial prompt, and the summarized body as the "continuation". The original dataset also had span annotations, which didn't make sense given the ALLSIDES data, so I just set them manually to be the lengths of the headline. In accordance with typical NLP standards, I also split 10% of this data out for a test set. The full data set had 6448 rows, leaving 5803 rows in the training set and 645 in the test set.

3.2 Target Characteristics Encoding

For this experiment, I focused mainly on political belief encoding. However, to show that this could extend to an arbitrary amount of features, I also worked two other simple features - "customWord" and "wordLength".

I represent a specific combination of these features in a CHARMLIST , a list of tuples [VALUE, LEVEL]. Each feature that the particular tuning will accomodate recieves one tuple in the CHARMLIST. VALUE ranges from -1 to 1, and parameterizes the features reward function. LEVEL ranges from 0 to 5, and represents the desired weighting of that particular feature in the overall model tuning.

VALUE is intentionally open-ended, to allow for a wide selection of features. With linear features, -1 corresponding with the least amount of that feature, and 1 the most. In this case, a VALUE of -1 would try to "tune out" the feature of the model's generation, 0 would leave it unchanged, and 1 would promote it. In other cases this can also correspond to a multi-poled scale. In the political bias example,a value of -1 represents relatively right-leaning text, 0 relatively central and 1 relatively left-leaning. This openness to interpretation is intentional and seemed to work very well in my experiments.

level represents the importance that the particular feature has upon the final reward that is given to the model, and thus is used simply as a weight in the CHARM reward function. A CHARMLIST with a [1,5] encoding for feature A and a [1, 1] for feature B will see a much greater amount of feature A (or a very left-leaning result using our political bias example) when compared to feature B.

3.3 Features

In order for CHARM to work, each feature must also provide a reward function, that generates a measurement based on the input text. Features can utilize any sort of reward function, and may return values along any scale. In this model, I've used both a highly linear reward function (customWord) and more complicated model (political), to demonstrate that either of these is effective. Additionally, I used differing scales, with the customWord feature using a reward function that vaired between 0 and 1 and with the equally linear wordLength feature, who's reward function varied among all reals.

For my experiments, I settled on the following three features, and explored different tunings with different values of each.

3.3.1 Political

As mentioned before, the primary feature that I focused on was political leaning. As I wanted to compare to Chen et al., I used their data set to provide the inital prompts for the model to work with. However, they did not provide a analytical measurement method for leaning detection and as such I wasn't able to think of a way to create a reward function based on their work. Instead, I used the model developed by (Baly et al., 2020) , which is a finetuning of a BERT model that expresses polarity of the input text. The model outputs a value from -1 to 1, with -1 corresponding to very right-leaning, 0 relatively centrist, and 1 being very left-leaning. Thus I could then set the reward function for the political feature to be the output classification provided by their model, weighted by the VALUE variable in the CHARMLIST. In doing the weighting I was careful to ensure that the output of the reward for this feature would follow the VALUE, i.e. if a value of 0 was given for variable, the reward function would output a lower reward if the inputted text was very right or left leaning, and higher if it was more centrist, rather than the close to zero value it would have otherwise.

3.3.2 Custom Word

Secondarily, to show the effect of a linear feature, I also developed the CUSTOMWORD feature. It is remarkably uncomplicated - it measures the occurrence of the inputted word(s) and returns that count normalized by the length of the sentence. Like political, it is also weighted by the VALUE variable, to ensure that it maximizes the given level of the parameter. However, unlike political, it utilizes a completely linear scale, with -1 penalizing the model based on the number of occurences, and 1 rewarding it. No special consideration was given to the case where VALUE was 0; in this case the reward function will be negative and grow smaller as VALUE approaches 0. Thus, a VALUE of 0 is equivalent to a LEVEL of 0, or not including the feature at all.

3.3.3 Word Length

Finally, I also tried a word length feature. This feature simply returned the mean length of the words given an sentence as input. As mentioned earlier, I did not do any normalization here, so the mean range arbitrarily. I did however perform the same multiplication as I did for the CUSTOMWORD feature, so a negative VALUE will give a negative reward, thus prioritizing shorter words, and a positive VALUE will reward longer word lengths. As with wordCount, no special attention was paid to a VALUE of 0, it is equivalent to not including the feature.

3.4 The Charm Function

To combine the features, I created a CHARM function,

$$R = \sum_{0}^{N} \frac{1}{(1 + e^{-}(n.reward(text) * n.LEVEL)} \tag{1}$$

where N is the number of features. Notice that the impact of each feature is modulated by the provided LEVEL in the CHARMLIST. It also utilizes a sigmoid function, so that the final reward provided to the Quark model ranges between -1 and 1. I added this as the toxicity reward function that the unmodified model had also ranged in this way, and I wasn't sure what kind of adverse effects might result from changing this. However I also dont think that removing it or altering it would generate a large performance improvement, as in my understanding of the model, the reward function's value is purely relative. A larger reward might cause the model to show more dramatic results with less training, but this would also likely be tempered by the KL divergence penalty that is already in place. In any case, I did not experiment with changing it so I cannot empirically state one way or the other.

Results for all of these are summarized in Table 1.

3.5 Model Setup and Training

The Quark model functions by iteratively fine-tuning the model based on it's own generations and a pre-defined reward function R. It is split into three stages:

1 Exploration: the model is used to generate new prompts.

2 Quantization: the resultant data is divided on its performance by the reward function

1 Learning: the data is then pre-pended with a token and used as the fine-tuning train set

I did not modify this procedure in my modifications.

It also utilizes a KL-divergence penalty, to ensure that the model does not stray too far from its original parameters. Although they experiment with different values of the coefficient, I did not have the bandwidth to do the same, Thus, in line with their findings, I use their default value of 0.05 for the coefficient. The use an Adam optimizer, with a default learning rate of 0.00005. I again used this default and did not experiment with alternate values.

In fact, hardly anything was changed in any part of the model. I tried my best to use the code directly as it was released, except, of course,for the reward section.

The only exceptions to this were largely due to utilitarian constraints. The first was a necessary code change that arose from what I believe to be the use of an older version of the transformers library. The "_get_logitcs_warper" function was called implicitly with parameters, which best I could find was outdated, being replaced instead by a GenerationConfig instance. However, in this replacement, I used the same parameters and did not change their order or structure.

I also changed the underlying model from "gpt2-large" to the smaller "gpt2". I had done a good bit of testing with the "gpt2-large" but kept running into hardware constraints. Moving to this smaller model also enabled me to run the tuning process on my home computer.

I ran all training runs on my home computer (no GPU/i5 processor/16gbram). In order to faciliate that, I also had to reduce the batch size from the default level of 128 in the original Quark model to 2 in my model, and the total episode count from 30000 to 30. I would have very much liked to be able to experiment with changing these values, but was not able to in the time given.

However, I was able to see marked results, and I do not think that the reductions in the training process that I had to take affected that. Since this is a tuning and not a training, additional tuning passes are likely to only improve the features rewards, at least up to a certain point where the language becomes non-sensical. My intent in this paper was just to show positive change according to the CHARMLIST, and I have showed that. Before any use case, an exploration of the extents of this system, using upgraded hardware and much more time, would be highly advisable.

I trained on the entire dataset provided by Chen et al., save of course for the test set which I set aside at the beginning. However, in this model structure, these texts are used as prompts to a LLM, and the real training comes from the reward functions of the generations. Thus, increasing the total episode count would likely have similar results to increase the training dataset size, at least up to a certain threshold.

3.6 Evaluation

To evaluate the various tuning tests, I follow Lu et al. and simulate responses from the tuned model, measuring them on the rating from the CHARM function. For ease of measurement, I also lowered the number of samples per prompt to 2 from the Quark default of 25.

I evaluated using the same set of reward functions that I used to tune the model, but also recorded the individual feature reward values and not just the CHARM value. I've displayed the results of my tests in the following section.

4 Results

I ran the tuning method, with the parameters explained the "Model Setup and Training" section, on the following different options for the CHARMLIST.

A Default Right: A baseline model showing maximum right political leaning, with no other features

B Default Left: A baseline model showing maximum right political leaning, with no other features

C Default Right with Feature: The same political tuning as the baseline models, but now introducing another feature, specifically the customWord function with "smart" as the maximized word.

D Default Left with Feature: The same as C, but this time with the opposite political stance.

E Smart: The same features as all the prior ones, but this time the political feature is set far inferior in LEVEL compared to the customWords feature.

F Long Words: Here, the customWords feature is replaced by wordLength feature, set to reward longer words. The political model is kept at baseline levels, and set to be left-leaning.

G Small Words: The same as F, but this time rewarding shorter words.

The results for the different values of CHARMLIST are summarized in Table 1.

4.1 Discussion

My initial goals in this paper were as follows:

- Basic Replicate Chen et al., and be able to convert from right leaning to left leaning.
- Moderate Be able to adjust polarity levels at a more fine-grained level, i.e. be able to generate far right, somewhat right, somewhat left, far left, etc. utterances.
- Ambitious Train a model using alternate features. [2]

[2] Originally, this was meant to be from (Piotrkowicz et al., 2017), however I did not have the ability to implement all of these features

Test Case	Feature	Value	Level	Result	CHARM
A	politicalLeaning	-1	5	-0.229	0.952
B	politicalLeaning	1	5	0.260	1.056
C	politicalLeaning customWord("smart")	-1 1	5 2	-0.217 0.001	0.956
D	politicalLeaning customWord("smart")	1 1	5 2	0.255 0.000	1.051
E	politicalLeaning customWord("smart")	1 1	1 5	0.047 0.000	0.559
F	politicalLeaning wordLength	1 1	3 5	0.149 23.263	0.559
G	politicalLeaning wordLength	1 -1	3 5	0.175 -23.014	0.540

Table 1: Test Cases for the CHARM model

4.2 Feature Performance

From A and B, we can see that I have successfully reproduced the work of Chen et al. within the Quark framework, and thus met the basic goal. Specifically, when the VALUE was set to -1, we observed a more right-leaning mean polarity level, and when VALUE was set to 1, we observed more left-leaning generations. Additionally, F and G show that we are able to tune this political polarity to a highly granular degree, thus meeting our moderate goals. Finally, C,D,E,F,and G show that we are also able to learn alternate features, to varying degrees of success, thus partially meeting our ambitious goal.

I do not understand why model B was able to create generations that were consistently labeled as being farther right than model A's generations were left. It would be surprising to me that gpt2's unmodified generations would be classed as more left leaning than right in the general run of cases - a more likely observation would be that perhaps the classifier provided by Baly et al. might be the cause of this more likely. Otherwise, it is possible that this discrepancy is simply due to my model not being tuned for long enough, and with greater processing time it would go away.

Notice from C and D that the customWord feature only minimally effected the polarity switching (0.012 for the right leaning models and 0.005 for the left leaning). Additionally, the generations fared fairly consistently despite whatever the other feature was, in F and G the models vary very similarly with respect to the political feature, despite the wordLength feature directly flipping. This seems to indicate the CHARM will be able to satisfactorily tune a LLM to a variety of features, at least with enough tuning time.

I would have liked to experiment with different features, as I am sure various levels of correlation would have adversely affected performance. Even still, it is interesting to me that right-leaning C had slightly higher performance with the customWord feature than did the left-leaning. I am unclear if this was just a fluke of my training examples of it is indicative of some correlation between the two in the data or else gpt2 itself(which I would find unlikely)? This would likely only be visible with significant more testing, however it does bring up the possibility of working with these very flexible subjects in an interesting - and quantifiable - way.

I am as of yet unclear as to why model G performs the way that it does. I would have thought that a negative reward provided by the wordLength reward function would have penalized the model and led to shorter words, instead it seems to have had the same effect as it did in F, but yet the reward is still being returned negative. There is a high chance it was simply caused by a mistake on my end, while setting this to run late the night before, but it also may have something to do with the structure of my reward function for this feature - I am sure there is yet a more excellent way to measure this characteristic.

An important distinction with the customWord feature was the presence of the word in the initial prompt set. I chose the word "smart" randomly, just to illustrate the possibility of the feature, and because I thought it might reasonably be expected to show up in all leanings of news media. For reference, the word "smart" appeared 6 times in the training prompts, and zero times in the test prompt, although again Quark is tuned not on these prompts but rather on the generations from these. However, I did not experiment all that much with word choice and there is still the possibility if not likelihood that it would show up too rarely to meaningfully impact the model. An interesting future research would have been to see if differing word choices, or else including synonyms, would have increased performance in this feature.

4.3 Contextual Performance

Following the example ofLu et al.'s paper, I did measure the distribution of the various datasets, and they are summarized in Table 2. They characterize these values as "sequence-level repetition, defined as the proportion of repeated n-grams (rep-n)". Their distributions ranged from 0.43 to 0.80 for Dist-2 and 0.66 to 0.84 for Dist-3, so we seem to still be in the same ballpark.

Again, in accordance with the Quark model we also do some human validation on the model results. However since I am the only human here, I reliquinsh this work to the readers, including some examples. For model A (baseline right-leaning model), the prompt "There's another controversial Hollywood racial decision" resulted in "There's another controversial Hollywood racial decision that's been exploding in Hollywood for some time now.", while model B(baseline left-leaning model) generated "There's another controversial Hollywood racial decision that has compelled federal officials to disclose bias in federal and state law against Donald Trump: Information reported...". These results scored the highest in the political bias measurements of the samples generated.

We can see from these results that model comprehension hasn't been affected, and also that the model is capable of generating responses that contain named entities that are of the opposite polarity (e.g. Donald Trump, a known right-leaning politician), thus suggesting that it has retained the abilities of the underlying LLM in knowledge comprehension and has largely focused the tuning on stylistic choices.

5 Future Work

In line with the common themes expressed so far, the most likely avenue for future work on this subject would be to train this model with more robust models structure and better feature depictions. I have saved all my model checkpoints, which might help in that endeavour somewhat, both to determine weak points and to develop insights for future modifications.

Test Case	Dist-1	Dist-2	Dist-3
A	0.600	0.874	0.865
B	0.603	0.872	0.864
C	0.604	0.879	0.866
D	0.604	0.879	0.866
E	0.604	0.877	0.865
F	0.600	0.878	0.864
G	0.600	0.878	0.864

Table 2: Text Diversity measures for the CHARM model

Future work could easily extend this to any number of other features, including contradictory ones, to examine how these features interact on a larger scale. It would be very interesting to see if model performance continues to degrade as features are added, and what combinations of features perform better together than others, as discussed briefly in the discussion section. Additionally, as shown in the results section, even with a very high level the model did not achieve a very high polarity measure. I am sure there are ways to improve this, although in this situation this is effectively introducing bias and I was not comfortable with doing that. If future work could introduce other features, maximization beyond the levels I show here could be very useful.

Furthermore, in my CHARM model, all weights are combined, and the model weights the generations and adds the pre-pended token strictly on that scale. It am curious as to whether measuring and appending a different token for each feature would work better than this.

Finally, as discussed in the findings section, the low performance on the custom word feature is likely due to poor representation in the training set. Feature introduction would be an interesting avenue to explore in future work, and would likely result in performance enhancement in this feature.

6 Limitations

Likely the biggest limitation in this paper was hardware. In order to be able to run it, I had to significantly minimize the training time. In addition, just getting the Quark model to run took most of my time throughout the semester, and so I ended up only having the last few days to do any real experimentation, which was far less than ideal.

I was able to get access to the CRC computing cluster, but since by that time I had a few successful training runs done, I thought it would be a better use of my time to expand on this paper and try other experiments rather than focus on getting the code to run on the cluster. In hindsight, it would have been much more wise to have worked on the cluster from the beginning, and likely would have saved a great deal of time. I had originally planned on using google colab's, and spent most of my time trying to get it to run, but (somewhat strangely) it kept running out of memory. The only way I could get it to run is to reduce it to the current level anyways, and at that point it was easier (and about the same speed) to just run it locally.

The dataset from Chen et al.'s paper was not the largest. As mentioned earlier, this may have impacted the model quality less than my small batch size and low episode count, but it still would likely benefit from more training prompts.

7 Ethical Concerns

The nature of these experiments are sensitive ethically, and the reality of that was not lost on the author. The CUSTOMWORD feature has no restriction on what word is fed in, and could be easily used to reward derogatory or otherwise unsavory words. Furthermore, other reward values could be easily added to promote negative effects in generated texts. However, these dangers are still present without using the CHARM model, if at a lesser extent. It is the hope of the author that future users will engage with the model ethically and safely.

Somewhat less severe, there is a question of the ethicality of "maximizing" bias, even if it is political in nature like it is in this paper. Consuming media from only one outlook is damaging and can result in "information silos" that can contribute to hate and other unhealthy mannerisms. However, as I mentioned earlier, it is possible that implementing a model similar to this might work against this, by automatically exposing media consumers to alternate viewpoints to the same factual content.

8 Conclusion

In this paper, I have described by CHARM addition to the QUARK model developed by (Lu et al., 2022), and shown that it performed successfully in a headline polarity translation task. Additionally, I showed that it could be expanded to multiple other features with little performance drawbacks. I also release my CHARM reward function and the model checkpoints after fine-tuning on several different CHARMLISTS for future study.

References

Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020. We Can Detect Your Bias: Predicting the Political Ideology of News Articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP '20, pages 4982–4991.

Avanti Bhandarkar, Ronald Wilson, Anushka Swarup, and Damon Woodard. 2024. Emulating Author Style: A Feasibility Study of Prompt-enabled Text Stylization with Off-the-Shelf LLMs. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 76–82, St. Julians, Malta. Association for Computational Linguistics.

Wei-Fan Chen, Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. 2018. Learning to Flip the Bias of News Headlines. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 79–88, Tilburg University, The Netherlands. Association for Computational Linguistics.

Jakob-Moritz Eberl, Hajo G. Boomgaarden, and Markus Wagner. 2017. One Bias Fits All? Three Types of Media Bias and Their Effects on Party Preferences. *Communication Research*, 44(8):1125–1148. Publisher: SAGE Publications Inc.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style Transfer in Text: Exploration and Evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). Number: 1.

Tim Groeling. 2013. Media Bias by the Numbers: Challenges and Opportunities in the Empirical Study of Partisan News. *Annual Review of Political Science*, 16(Volume 16, 2013):129–151. Publisher: Annual Reviews.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled Representation Learning for Non-Parallel Text Style Transfer. ArXiv:1808.04339 [cs].

Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. MULTIPLE-ATTRIBUTE TEXT REWRITING.

S. Robert Lichter. 2014. *Theories of Media Bias*, volume 1. Oxford University Press.

Ruibo Liu, Chenyan Jia, and Soroush Vosoughi. 2021a. A Transformer-based Framework for Neutralizing and Reversing the Political Polarity of News Articles. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–26.

Ruibo Liu, Lili Wang, Chenyan Jia, and Soroush Vosoughi. 2021b. Political Depolarization of News Articles Using Attribute-Aware Word Embeddings. *Proceedings of the International AAAI Conference on Web and Social Media*, 15:385–396.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable Text Generation with Reinforced Unlearning. ArXiv:2205.13636 [cs].

Alicja Piotrkowicz, Vania Dimitrova, and Katja Markert. 2017. Automatic Extraction of News Values from Headline Text. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 64–74, Valencia, Spain. Association for Computational Linguistics.

Riccardo Puglisi and James Snyder. 2015. Empirical Studies of Media Bias. pages 647–667.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style Transfer from Non-Parallel Text by Cross-Alignment. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bakhtiyar Syed, Gaurav Verma, Balaji Vasan Srinivasan, Anandhavelu Natarajan, and Vasudeva Varma. 2020. Adapting Language Models for Non-Parallel Author-Stylized Rewriting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9008–9015. Number: 05.

Martin Wessel, Tomás Horych, Terry Ruas, Akiko Aizawa, Bela Gipp, and Timo Spinde. 2023. Introducing MBIB - The First Media Bias Identification Benchmark Task and Dataset Collection. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2765–2774, Taipei Taiwan. ACM.

Zheng Zhang, Fan Yang, Ziyan Jiang, Zheng Chen, Zhengyang Zhao, Chengyuan Ma, Liang Zhao, and Yang Liu. 2024. Position-Aware Parameter Efficient Fine-Tuning Approach for Reducing Positional Bias in LLMs. ArXiv:2404.01430 [cs].