Tuning for Bias:
An Analysis of Political Bias Through Title Generation

Li-Heng Pan, Isaac Wecht, Sagar Raichandani

CSCI 2470: Project Report

1 Introduction

As businesses and social organizations are drawn to the vast potential of A.I., language models of tomorrow are likely to be fine-tuned versions of LLMs developed by specialized institutions today. Trained on tailored datasets that echo the biases of the institutions developing these tuned models, we believe that such selective development poses a significant challenge to local and global communities. Through this project, we hope to study how language models trained on two divergent media houses' data–conditioned on each organization's political view–can exacerbate bias. More specifically, we develop a transformer-based language model that predicts an article's title, conditioned on a liberal vs. conservative viewpoint. Bearing that title generation is a form of text summarization, we hope to highlight that sensationalized news headlines of media houses split across political views can present different interpretations of the same content/information.

2 Methodology

Our data is sourced from two of the largest media houses in the US–The New York Times, a leading left-leaning news organization, and Fox News, which boasts the highest viewership among conservative audiences. For NYT, each article's URL and its title is sourced from a public API; a Beautiful Soup scraper then fetches the article's content from the published webpage. With Fox News, data collection was a bit more involved. Since Fox does not have a public API, we built a Selenium and Beautiful Soup scraper to collect relevant article links and subsequently scrape the content and title. In total, we collected over 10,500 New York Times articles and 16,000 Fox News articles.

Next, as a part of pre-processing, we begin with data cleaning, i.e., removing encodings, standardizing text to lowercase, and removing all but a small set of special characters. We condition each article to the organization's view by prepending a stance "liberal" for NYT articles and "conservative" for Fox articles. This conditioning will prove to be useful later. Then, we split the data into training and test sets (0.95-0.05 by default), prepend and append special tokens '<start>' and '<end>', followed by padding/truncation to ensure a consistent length of 256 tokens for article content and 16 tokens for the article title. Finally, we restrict

the vocabulary to the top 15,000 tokens for titles and the top 100,000 tokens for content. Now, for each word in the title and content vocabularies, the word is mapped to its corresponding vector in GloVe, such that the word's index in vocabulary matches that of a 2D array. This array (or matrix) serves as an initializer to the embedding layers of the model.

2.1 Model Architecture

In the project, we leverage the Transformer architecture (see Fig. 1) to build our model. For the input embedding layer, we initialize the weights with GloVe's 100-d embeddings. Next, positional encoding is applied to incorporate the order information of the sequence. For the encoder block, there are two main components: the first is a multi-head self-attention mechanism, and the second is a fully connected feed-forward layer. We apply a residual connection after each component, followed by layer normalization. The decoder block follows a similar composition with some variation. We incorporate a "multi-head attention layer with causal masking." Causal masking is a mechanism that prohibits certain positions from attending to subsequent positions. The "cross-attention layer" performs multi-head self-attention on the encoder output and the decoder input. In our project, we use 2 encoder blocks, 2 decoder blocks, and 8 attention heads. After the encoder-decoder components, we include multiple linear layers and a Softmax activation layer.

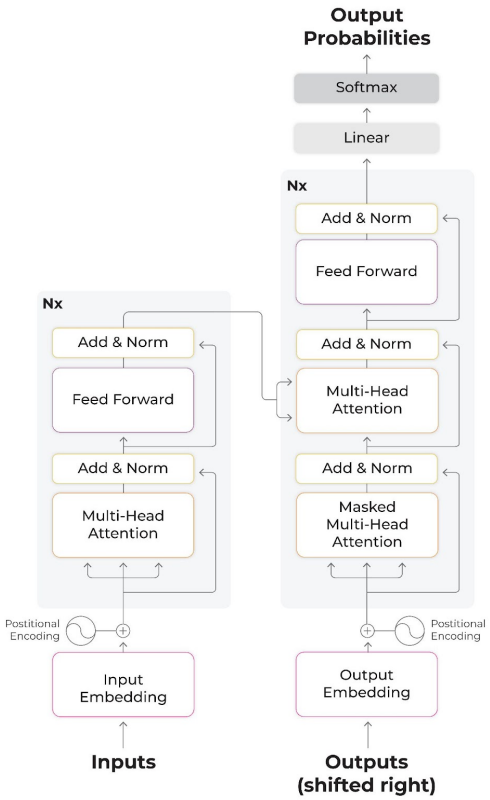

Figure 1: Transformer Architecture

2.2 Training & Testing

In the training stage, we train the model for 30 epochs with a batch size of 200 articles, considering a combined 25,000 articles from both liberal and conservative news outlets. We consider a modified version of sparse categorical crossentropy–masked loss–as the primary loss function. In masked loss, we essentially ignore the padded (index can be specified) tokens in the true labels, incentivizing the model that predicted words matter more than padding. Likewise, in metrics we consider the masked accuracy, which checks for matching words, disregarding padding.

In the testing phase, we first validate our model's capacity to generate coherent titles. We test it on a cumulative 1300 articles from both news outlets. Further, to manifest the inherent bias in the news articles, we modify each article in the test set by reversing the original view and concatenating it into the original test set. That is, each article is labeled with its original viewpoint and its opposing viewpoint in the modified test set. Then, we conduct a quantitative examination of coherence through BLEU and a qualitative examination of bias.

3 Results

Masked Accuracy (Validation)	Masked Loss (Validation)	mean BLEU (Test)
[0.49, 0.57]	[1.736, 2.316]	[0.287, 0.350]

Table 1: Metrics

Of the metrics discussed above, we train our model multiple times, and Table 1 shows the metrics for our model. Every metric is shown in a range.

Original title	**Predicted title**
NY Gov. Hochul tests positive for COVID-19: 'I'm asymptomatic'	new york gov hochul tests positive for covid 19
Iowa signs ban on gender transition surgery for minors, says it's 'in the best interest of the kids'	iowa gov laura kelly signs bill banning gender treatments for minors
Wisconsin Republicans block chickenpox, meningitis vaccine mandates in schools	wisconsin gov evers vetoes schools vaccine mandate
Biden admin dragged for blaming botched Afghanistan withdrawal on Trump: 'Disgraceful and insulting'	gop congressmen slam biden admin over afghanistan withdrawal

Table 2: Example coherent titles

The table above demonstrates that the model generates coherent titles well. While we do usually see relevance, there are several instances where the model's output, although coherent, falls short on relevance– that is, the model's predicted title is not strongly related to the content

of the article. While we are unsure of the exact reasons for this, we posit that limiting the number of article tokens to 256 restricts the model's ability to learn a deeper and more relevant context.

Original title	**Predicted title (Original View)**	**Predicted title (Opposite View)**
Adams lifts COVID vaccine mandate for most NYC workers, but thousands fired won't automatically get jobs back	nyc mayor adams to offer covid 19 vaccine mandate	new york city to end covid 19 vaccine mandate
More city, state leaders make decisions on lifting COVID-19 restrictions	new york city to require masks in covid 19	new york city to lift mask mandate for covid 19
Congress poised to repeal covid vaccine mandate for troops in military bill	poised to repeal covid bill mandate for troops in military bill goal looms looms soldiers would	lawmakers to resume vaccine mandate for troops in military bill goal soldiers soldiers soldiers denies

Table 3: Example biased titles

In Table 3, we highlight the model's ability to generate titles as per the "liberal" vs. "conservative" conditioning we talked about in methodology. In each of the above examples, it is easy to see that the model's predicted title shifts with switching the condition from "liberal" to "conservative" for originally left-leaning articles and vice-versa. While such a strong indication of bias was absent in most results, the model does pick up on biased themes occasionally– especially around topics that are divisive such as COVID-19 policies, climate change, and technology.

4 Challenges

This project highlights the complexity of language models. While we now have a newfound appreciation for subtleties with Transformers, attention layers, and masking, our project was (and remains to be) challenging with respect to:

1. **Building a consistent workflow with scraping**. Each website is different, and thus requires special considerations to scrape. While we had initially hoped for more information from a diverse group of media houses, it was simply too cumbersome in the scope of this project.
2. **Data cleaning and preprocessing is a delicate balance** with a high impact on downstream tasks. Incorporating facets of tokenization & vectorization that match assumptions of the pre-trained embedding layer's training philosophy is crucial to extract best results.
3. **Nimble model architecture**: with more flexibility comes the power of easy, organized experimentation, which is crucial to test new ideas and hyperparameter tuning.

4. **Conditioning for bias**: this continues to be a challenge– while we demonstrate bias in some test samples, with the model differentiating "liberal" vs. "conservative" tendencies (especially with articles related to climate change), we haven't realized a comprehensive strategy to quantify or incentivize the model to produce left vs. right leaning title consistently on cue.

5 Reflections

Our base goal is to build a model that can produce coherent and relevant titles. It meets our expectations with a mean BLEU score of 0.31 on the test set. For the target goal, we are able to find some paired titles that manifest the inherent bias in the data. While the majority of titles are coherent, many fail to demonstrate relevance, and most do not show bias. Eventually, we can only devote limited time to our stretch goal of quantitative evaluation of bias, yet we believe we can show bias quantitatively once the model is capable of consistently generating meaningful prejudiced titles.

In the beginning, we did not include computed masking in our encoder or causal masking in our decoder. The former prevents the model from learning the semantical representation of padding, and the latter hinders it from attending to subsequent sequences in the title. Hence, our model performed poorly and only repeatedly output the same word or the padding token. After integrating masking into our model, it performed as we expected, and the BLEU score also improved.

If more time is given, we would first consider collecting more data. On the one hand, Transformer is data-demanding, i.e., a huge amount of data is needed for decent performance. On the other hand, the inherent bias in news articles is subtle, so it takes a vast amount of data for it to surface. Additionally, we would consider doing finer data preprocessing, such as lemmatization, to improve our data quality. We would also experiment with GloVe embeddings of 200 dimensions since it may be capable of capturing a richer semantical representation than the 100-dimension version. Lastly, we would experiment with different model architectures, e.g., adjusting the number of encoder blocks, decoder blocks, and attention heads.

Our biggest takeaway from this project is that the training of deep learning models is a cumbersome task. It includes the collection of unstructured data, data preprocessing, model specification, model training, and cross validation. Every step has its nuances that we need to conquer, and none of them itself is an easy task. It takes a lot of effort and time to build an impressive deep neural network. That being said, we are extremely amazed by machines' ability to perform human-like jobs, such as generating news titles, and we look forward to further applications of deep learning in the future.