The Behaviors of BERT Attention Heads in Stereotype Detection

Joseph H. Hajjar
Dartmouth College, joseph.h.hajjar.21@dartmouth.edu

Follow this and additional works at: https://digitalcommons.dartmouth.edu/masters_theses

Part of the Artificial Intelligence and Robotics Commons, and the Theory and Algorithms Commons

Recommended Citation
https://digitalcommons.dartmouth.edu/masters_theses/171

This Thesis (Master's) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Master's Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
THE BEHAVIORS OF BERT ATTENTION HEADS IN STEREOTYPE DETECTION

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of
Master of Science

in
Computer Science

by Joseph Hajjar

Guarini School of Graduate and Advanced Studies
Dartmouth College
Hanover, New Hampshire

May 2022

Examining Committee:

Prof. Soroush Vosoughi, Chair

Prof. Bo Zhu

Prof. Hsien-Chih Chang

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies
Abstract

We are living in the age of information, where it has become increasingly easy to share ideas, news, and content which are seen by an increasingly large number of people. This increasing scope of the increasing amount of data that is being shared lends itself to the question: how can we determine whether what we are reading promotes a stereotype? Previous work has applied transformer based models in this domain yielding impressive performance, but few studies exist interpreting the nature of attention heads in this task. Our work explores the feature encoding and extraction behaviors of attention heads in transformer based language models in stereotype detection tasks. We focus our investigation on three stereotype datasets, CrowSPairs, StereoSet, and BUG, and employ two probing mechanisms, gradient probing and leave-one-out probing, to investigate the importance of different attention heads on different subsets of each dataset. Our findings can be leveraged in downstream applications to determine how to boost performance on these tasks as well as generate further understanding of the roles of different attention heads.
Preface

I would like to thank my advisor, Professor Sorough Vosoughi, for his guidance throughout this project, as well as my committee, Professors Bo Zhu and Hsien-Chih Chang, for their review of my work. I would like to thank Weicheng Ma, who guided my research on this project. Lastly, I would like to thank the engineers and researchers who worked on the open-source technologies which I used in my research [22, 9, 26, 36, 19].
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4 Data Partitioning</td>
<td>24</td>
</tr>
<tr>
<td>6 Methodology</td>
<td>26</td>
</tr>
<tr>
<td>6.1 Gradient Based Probing</td>
<td>27</td>
</tr>
<tr>
<td>6.2 Leave-One-Out Probing</td>
<td>28</td>
</tr>
<tr>
<td>6.2.1 Masking</td>
<td>29</td>
</tr>
<tr>
<td>6.2.2 Replacing Words of Interest</td>
<td>30</td>
</tr>
<tr>
<td>7 Results</td>
<td>31</td>
</tr>
<tr>
<td>8 Future Work</td>
<td>37</td>
</tr>
<tr>
<td>9 Conclusion</td>
<td>39</td>
</tr>
<tr>
<td>References</td>
<td>40</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction
With the ever-increasing usership of the Internet, more information flows between people than in any point in history. Individuals consume news whether they intend to or not. The content which they come across can influence their perception of reality and their conception of the world. This can come at the expense of marginalized groups of people if users come across a high volume of content which promotes stereotypes associated with this group of people.

A stereotype is an over-generalization about a group of people, e.g. Women are bad drivers. Even supposedly positive stereotypes, e.g. Asians are good at math, are known to hurt the group about which they generalize [24]. Some stereotypes are well-known, but it is not always clear whether a sentence promotes a stereotype as marginalized groups differ between contexts. Given the amount of information, particularly textual data, which people are exposed to nowadays, it is useful to develop mechanisms of determining whether or not a sequence of words promotes a stereotype. Models with these capabilities could track the proliferation of stereotypes on the Internet, advise users about how much bias may be in a document, or detect hate speech on increasingly popular social media platforms.

The advance of deep learning in the past decade has led to groundbreaking performance on a number of classification tasks [5] which implies a promise of its usability in this domain. Previous work has investigated the applicability of deep models to the task of stereotype and abusive language detection. These models have achieved remarkable results, but few studies have looked “under the hood”, or probed, these models to determine the reasons why models have been able to perform this well.

Our work applies different probing mechanisms to understand the relevance of attention heads in stereotype detection task. We focus our analysis on the popular BERT (Bidirectional Encoder Representation from Transformers) model [6], one of the most competitive language models on Natural Language Processing (NLP)
tasks, and examine the attention heads using both gradient-based and leave-one-out approaches.

The remainder of this paper is structured as follows. In Section 3 we provide background information necessary to understand the contents of the paper. In Section 2 we explain the objective of this task, and we discuss related work in Section 4. In Section 5 we describe the datasets we used in our analysis. In Section 6 we describe the different experimental settings we constructed and dive deeper into the different probing mechanisms we used. Lastly, we discuss results and future work in Sections 7 and 8, respectively.
Objective
The objective of this work is towards a better understanding of the feature extraction and encoding behaviors of attention heads in Transformer models on stereotype detection tasks. We are motivated by the lack of literature on the contributions of attention heads in stereotype detection tasks, in spite of the contrastingly plentiful studies training models to detect abusive language. A more holistic understanding of why these models are able to perform so well, by understanding which attention heads are deemed important by the model, can help boost the performance of these systems.
Chapter 3

Background
Below we describe and define terminology that will be useful in understanding the remainder of this paper, as well as providing general background information.

### Section 3.1 Classification in Natural Language Processing

Natural Language Processing (NLP) is subfield of machine learning research. Researchers in this realm explore how computers can be leveraged to analyze documents of natural language to extract information and insights which can be used to do useful things [4]. The goal of researchers in NLP is to develop models which can achieve human-level understanding of natural language [18].

A subset of NLP problems are classification problems. In this realm, models are tasked with approximating a function which maps a document to a probability distribution which describes a document’s probability of belonging to a certain class. The equation below demonstrates a binary classification problem where \( A \) is a document and \( f \) is the function the function an NLP model would approximate, which maps vectors of dimension \( d \) to a vector of dimension 2. An example of the classification here can be whether the document expresses negative or positive sentiment.

\[
f : \mathbb{R}^d \to \mathbb{R}^2
\]

\[
f(A) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\]

As demonstrated in the equation above, the document must be converted to a vector of a fixed dimension. A popular way to do this is to convert the words in the document to vectors [11, 27, 28, 23]. These vectors are learned by models which are trained on
large unlabeled textual data which are able to encode various semantic and syntactic relationships between words in the vector space.

### Section 3.2

**Deep Learning Models**

The success of deep learning models [5] in the past decade have made them a popular choice for natural language processing tasks. Deep learning models are also referred to as artificial neural networks as their learning mechanisms are inspired by the biological neural networks in the animal brain [21]. In this section, we describe the architecture of a fully connected, feed-forward neural network. This is one of the simpler neural network architectures.

These models consist of an input layer, hidden layers, and an output layer. Each layer consists of a number of neurons, and in fully-connected models, each neuron in layer \(i - 1\) will be connected to each neuron in layer \(i\). The number of neurons can vary by layer, but the input layer should have a number of neurons equal to the dimension of the vector which it expects as input, and the output layer should have a number of neurons equal to the number of classes which the model is being trained to predict.

Connections between neurons represent linear transformations of the output of the source neuron to be fed as input to the destination neuron. For example, the connection \(w_{ij}\) between neuron \(n_i\) in layer \(k - 1\) and neuron \(n_j\) in layer \(k\) can be interpreted as a scaling of the output of neuron \(n_i\) which will be one of the inputs to neuron \(n_j\). Neurons also have activation functions which scale their inputs. Such activation functions include the hyperbolic tangent function (Equation 3.3) and the more recently popularized rectified linear unit (Equation 3.4). Due to the nature of how these deep models are trained, activation functions need to be differentiable.
3.2 Deep Learning Models

Figure 3.1: This image demonstrates a simple neural network architecture. The bright blue neurons represent indices of the input vector, with one neuron per index. In this case, the vector which the model expects as input would be of size 8. The purple and dark blue neurons represent the two hidden layers of the network. These neurons would contain intermediate values which will aid the model in its calculation. The yellow neurons represent the output of the model. If we assume that this model is tasked with classification, the four yellow neurons would represent the four classes as which the input could be classified, and their values would correspond to the probability that the input belongs to each class. The arrows connecting the different neurons represent linear transformations.

\[
\text{tanh}(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3.3}
\]

\[
\text{ReLU}(x) = \begin{cases} 
0 & \text{if } x < 0 \\
x & \text{else}
\end{cases} \tag{3.4}
\]

Figure 3.1 displays a visual of the fully connected neural network architecture\(^1\).

3.2.1. Training and Evaluation of Deep Models

When a neural network is trained, training data is split into three batches, a training set, a validation or dev set, and a test set. The data will be trained on the training set, and validated against the dev set which will consist of examples which the model has

\(^1\)https://openclipart.org/detail/290666/neural-net-graph
not seen in training. Finally, once the model has achieved satisfactory performance on the dev set, it will be assessed against the test set.

The training data is partitioned into **batches** of equal length and these batches are passed as input to the model. In a **forward pass**, the model produces a prediction for each of the samples in the batch and records the difference between its prediction and the ground truth label from the data. A loss is calculated between these two distributions. In classification tasks, the loss is typically the cross entropy between both distributions. The cross entropy $H$ of two probability distributions $P$ and $Q$ is defined as

$$ H(P, Q) = -\sum_{x \in X} P(x) \cdot \log(Q(x)) $$

where $P$ is the target distribution and $Q$ is an approximation of $P$. Intuitively, it is the number of additional bits required to represent an event in probability distribution $P$ using distribution $Q$. Typically, the activations of output neurons in classification tasks are passed through a softmax function which converts the scores to a probability distribution and ensures they sum to one. Then each output neuron’s activation can be interpreted to be the confidence with which the model believes an input belongs to that class. The softmax $\sigma$ for an entry $i$ in a vector $\vec{z}$ of size $K$ is

$$ \sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} $$

Following the forward pass, the model uses an algorithm called backpropagation to update the weights of the model [12]. The **gradient** of the error with respect to each of the weights of the network is calculated, and the weights are updated by adjusting them in the direction of steepest descent. A full pass through the data is called an **epoch**.

At every epoch, a model is evaluated against the validation data set to determine
how well it can perform on data which it did not see in training. For classification tasks, a common evaluation metric is the F1 score, shown in Equation 3.7, which evaluates the balance of the precision and recall of a model. The precision $P$ of a model is the ratio of true positives (samples for which both the model and ground truth annotation is the same) to the total number of samples labeled positive by the model, and the recall $R$ of a model is the ratio of true positives to the number of samples labeled positive by the dataset.

$$F1 = 2 \cdot \frac{(P \cdot R)}{(P + R)}$$  \hspace{1cm} (3.7)

The F1 macro score, useful in multi class classification problems, is an unweighted mean of F1 scores across all labels in the dataset.

### Section 3.3 Transfer Learning

Transfer learning is the process of transferring the weights of a model which has been trained to perform a certain task and using them as an initialization of the model to train for a separate task. The act of training this model to perform the former task is called pretraining, and the act of training to perform the secondary task is known as finetuning the model. In some cases the model architecture may be altered, for example additional layers added, or some model weights will be frozen during the secondary training, i.e. the weights will be resistant to backpropagation.

The intuition behind this process is that models may learn intermediate features in their pretraining which can be useful to a variety of tasks in pretraining. For example, while training a model to determine whether a document expresses positive or negative sentiment, it may learn how to represent emotion, which could be useful
in other tasks such as hate speech detection. Weights in the bottom layers of these models are typically associated with feature extraction and encoding of inputs which they learned in their pretraining. This is in contrast to weights at the top layers, which contribute more directly to the prediction of the model.

### Section 3.4

Transformers

Transformers are a specific architecture of deep models which follow an encoder-decoder architecture [33]. This architecture can be understood to consist of two models, an encoder and a decoder. The encoder model produces an intermediate representation of the input and the decoder model takes this representation as input to produce an output. For the purposes of this paper, we will introduce various transformer concepts in the context of BERT [6].

#### 3.4.1. BERT

BERT is a language model popular in natural language processing due to its ability to be finetuned to perform various tasks. BERT was simultaneously pretrained on two tasks, masked language modeling and next sentence prediction. Masked language modeling consists of passing a sentence to the model with the MASK token over certain words and training the model to determine what words should replace these tokens. In next sentence prediction, two sentences are fed to BERT, and BERT is tasked with predicting whether the second sentence follows from the first sentence. Through various state-of-the-art results, BERT has been shown to learn a rich representation of language. We use the BERT-base model in our research, and subsequent mentions of BERT will be in reference to this model.

The BERT-base architecture is a 12 layer encoder network. Figure 3.2 demon-
3.4 Transformers

3.4.1 Background

Figure 3.2: One of the 12 encoder layers in the BERT-Base model. Each layer has two sublayers, multi-headed attention and a feed forward network. The multi-headed attention sublayer in BERT-Base consists of 12 attention heads.

Strates the architecture of an encoder\(^2\). An input document is “tokenized”, a process which converts the document words into tokens which exist in BERT’s vocabulary. A special [CLS] token is placed at the beginning of the sequences whose hidden state is used to represent the sequence as a whole, and a [SEP] token is placed in between different subsequences within the sequence, such as sentences. BERT uses WordPiece embeddings [37] to convert these tokens to vectors. Positional vectors are added to each input token, and these vectors follow a pattern which the model learns during training to introduce a notion of distance between words in the input sequence and the position of each token in the sentence.

Subsequently, the model employs a multi-headed self-attention mechanism to calculate the relevance of each token relative to other tokens in the sentence. To understand the attention mechanism, consider the matrices \(Q\), \(K\), and \(V\) of dimension

\(^2\text{https://www.researchgate.net/figure/The-Transformer-encoder-structure_fig1_334288604}\)
3.4 Transformers Background

sequence length $n \times 64$ where $n$ is the sequence length. These are the query, key, and value vectors, respectively, for each word stacked into matrices. Then we have the attention calculation

$$\text{Attention}(Q, K, V) = \sigma \left( \frac{QK^T}{d_k} \right) V$$

(3.8)

for each head of the model where $\sigma$ is the softmax function from Equation 3.6 and $d_k$ is the column dimension of the $K$ matrix, 64. In multi-head attention, multiple $Q$, $K$, and $V$ matrices are used, one at each attention head, and the equation becomes

$$\text{Multi-Head Attention}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_n)W^O$$

(3.9)

where each $\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$ and $W^O$ is a learned weights matrix. The product of the $\sigma \left( \frac{QK^T}{d_k} \right)$ has dimension $n \times n$ where $n$ is the sequence length. Each entry $(i, j)$ in this matrix can be considered as the relevance of token $i$ and token $j$. Thus, the output of this intermediate calculation is the attention distribution. The output of the Attention equation has dimension $n \times 64$, and each row is the hidden state of a token at a specific attention head. The BERT model has 12 attention heads in each encoder layer, for a total of 144 attention heads across the model.

It is important to note that the output of BERT is an encoding of each token it processed. For the purposes of this research we are using BERT for classification, so we add additional layers on top of BERT to this end. First we add a pooler layer, which is a linear transformation of the [CLS] token. Next, a linear layer takes the output of the pooler layer and transforms it to a probability distribution for classification. Figure 3.3 demonstrates the full architecture of the BERT model for classification. The weights in the pooler and fully connected layers outside of BERT are not pretrained and will be randomly initialized in the finetuning process.
3.5 Probing Mechanisms

In spite of their ability to achieve very high performance on various tasks, a common criticism of deep models is the lack of interpretability with regards to how these models arrive at a decision. Probing is the act of looking “under the hood” into model parameters with the goal of understanding what goes on in the model’s processing of inputs. We employ two such mechanisms in our research, gradient-based probing and leave-one-out probing.

Gradient-based probing ranks weights based on their gradients. If the gradient on a weight has a large absolute value, it has a greater impact on the loss function. Thus we can say that the weight contributes strongly to the model’s reasoning. Leave-one-out probing relies on input perturbation to determine the importance of different parameters in the model. In NLP tasks, this can look like changing a word in the input sequence and examining how the activation of neurons within the model change. In this paradigm, weights which are altered when a certain word is omitted can be reasoned to encode particular information relevant to that word. Gradient-
based probing identifies parameters within the model which are particularly useful in feature extraction, the ability to identify features, while leave-one-out probing identifies parameters which are useful in feature encoding, the ability to represent features in the inputs.
Chapter 4

Related Work
Work on detecting stereotypical speech and hateful language focus on different languages and different types of biases. These studies leverage labeled datasets to create end-to-end mechanisms for detecting hate speech. A large emphasis has been placed on training models from text generated on social media platforms such as Twitter. [35] construct a dataset of 16k labeled tweets and develop a machine learning approach to classify them as promoting racism, sexism, or neither. [29] introduce MeTwo, a labeled dataset of Spanish tweets with sexist attitudes and expressions, and conduct a survey of experiments aimed at determining the feasibility of using different machine learning techniques to the end of classifying the samples.

[13] incorporate lexical features with the BERT model to detect abusive language. In this study, the researchers experiment with a setting where input sequences are passed both through BERT and a hate lexicon to produce hidden representations before being passed through a dense layer for prediction. They utilize datasets with biases across multiple domains. The work demonstrates that their HurtBERT model with lexical features outperforms the baseline BERT’s performance in these tasks. [31] finetune a BERT model which can be used to detect partisanship in news and sequences which promote stereotypes about immigrants in political speeches. Additionally, they conduct an analysis of the attention distributions on the final layer of BERT, to gain insight into which words are weighted heavily by the model in this task. They find that words such as “irregulars” and “illegals” are deemed important by the model. [1] investigate the applicability of transfer learning in this realm and conclude that the domain shift is not considerable when shifting to detection of abusive language. [3] look at different types of attention mechanisms in Bidirectional Long Short-Term Memory [10] on offensive language detection and determine that contextual attention outperforms self-attention.

[8, 20, 32] analyze the gradients of deep models to improve their interpretability.
Specifically, [20] analyze the contributions of attention heads in transformer-based language models in multi-lingual sequence labelling tasks using their gradients. They conduct experiments to determine which heads to prune in order to improve the model’s performance. [16] introduces the framework for the leave-one-out approach. The researchers remove representation at different levels of an recurrent neural network (RNN), from tokens in the input sequence to hidden units in the model itself. The change in performance across these erasures helps the researchers determine what the model correctly and incorrectly deems important.
Chapter 5

Datasets
5.1 CrowS-Pairs

<table>
<thead>
<tr>
<th>Sent More</th>
<th>Blacks never listen to their parents.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sent Less</td>
<td>Whites never listen to their parents.</td>
</tr>
<tr>
<td>Stereo Antistereo</td>
<td>stereo</td>
</tr>
<tr>
<td>Bias Type</td>
<td>race-color</td>
</tr>
</tbody>
</table>

Table 5.1: A sample pair from the CrowS dataset. The Sent More field denotes a sentence which promotes a stereotype, while the Sent Less field denotes the same sentence with the disadvantaged group replaced with an advantaged group. The stereo Antistereo field denotes the stereotypical direction of the pair and the bias type the domain of the stereotype.

Section 5.1

CrowS-Pairs

The CrowS-Pairs [25] dataset consists of 1,508 sentence pairs which are minimally distant, with one promoting a stereotype about a historically disadvantaged group in the United States and an identical sentence replacing the disadvantaged group with an advantaged group. The former sentence is always promoting a stereotype while the latter is promoting an anti-stereotype. The data was crowdsourced using annotators on Amazon MTurk to generate sentence pairs and each sample pair was validated by 5 annotators.

The project emphasizes that crowdsourced nature of the dataset allows for more diversity in the structure and content of the stereotypes and ensures that the stereotypes are all ones which are collectively understood to be stereotypes. The samples are categorized into nine subcategories based on which stereotype the stereotypical sentence promotes: race, gender, sexual orientation, religion, age, nationality, disability, physical appearance, and socioeconomic status. In addition to the bias type, the dataset includes a field indicating the stereotypical direction of the pair, stereo or anti-stereo. A stereo direction indicates that the pair demonstrates a stereotype about a historically disadvantaged group, while an anti-stereo direction means that the pair violates a stereotype about a historically disadvantaged group. A sample
5.2 BUG Datasets

Table 5.2: A data sample from the BUG dataset. The sentence consists of a profession and a pronoun which refers to it. The Profession First Index field and the G first index field denote the index of the profession and pronoun in the sentence token list, respectively. The Stereotype field denotes whether the sentence promotes a stereotype or not, and the Distance field denotes the difference between the Profession First Index and G First Index.

<table>
<thead>
<tr>
<th>Sentence Text</th>
<th>A librarian shares her views on information literacy.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profession</td>
<td>librarian</td>
</tr>
<tr>
<td>G (pronoun)</td>
<td>her</td>
</tr>
<tr>
<td>Profession First Index</td>
<td>1</td>
</tr>
<tr>
<td>G First Index</td>
<td>3</td>
</tr>
<tr>
<td>Predicted Gender</td>
<td>Female</td>
</tr>
<tr>
<td>Stereotype</td>
<td>1</td>
</tr>
<tr>
<td>Distance</td>
<td>2</td>
</tr>
<tr>
<td>Num of Pronouns</td>
<td>1</td>
</tr>
<tr>
<td>Corpus</td>
<td>pubmed</td>
</tr>
</tbody>
</table>

The BUG dataset [15] was created with the intention of mitigating shortcomings of previous work attempting to quantify bias [14, 38, 30]. These work used artificially constructed datasets which did not accurately represent the natural language training distribution and used datasets which were too small to substantially train or finetune a model to mitigate gender bias.

[15] sampled three large corpora, PubMed abstracts, Wikipedia, and Covid19 Research, using syntactic matching to extract sentences with 14 different syntactic patterns ensuring that each sentence mentions a human entity and an associated pronoun. The sentences were then labeled with whether or not they promoted a stereotype associated with gender-role. This resulted in a diverse dataset with 105,687 sentences.
Section 5.3

StereoSet

The StereoSet dataset [24] contains both inter- and intra-sentence stereotype examples. We focus on the intersentence subset of the data, of which there are 6,369 samples. In this subset, each sample contains two sentences. The first is a context sentence introducing a target demographic. The second is a descriptive operative sentence which either promotes a stereotype regarding the target demographic, promotes an anti-stereotype regarding the target demographic, or is unrelated to the context sentence. The data is split into four domains, profession, race, gender, and religion, and as indicated previously, the pairs are either classified as stereotypical, anti-stereotypical, or unrelated.

[24] emphasized realistic sentences in their dataset. One annotator was tasked with generating both context and operative sentences, and these were then labeled by five separate annotators as stereotype, anti-stereotype, or unrelated. An example

Table 5.3: An example of a sample in the StereoSet dataset. The operative sentence can be interpreted as either a stereotype, an anti-stereotype, or as being unrelated to the context sentence. Each context sentence has three corresponding sentences, each corresponding to one of the mentioned labels. The sample also includes a target word, which indicates the group about which the sentence pair generalizes.

<table>
<thead>
<tr>
<th>Context</th>
<th>My Professor is a hispanic man.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence</td>
<td>He is a legal citizen.</td>
</tr>
<tr>
<td>Bias Type</td>
<td>race</td>
</tr>
<tr>
<td>Label</td>
<td>anti-stereotype</td>
</tr>
<tr>
<td>Target</td>
<td>Hispanic</td>
</tr>
</tbody>
</table>

We used the Balanced BUG subset of the data sampled from the full 105,687 sentences to ensure equal male and female entities and balanced stereotype and anti-stereotype samples. This reduces the size of the dataset to 25,504 sentences. The full schema of the data can be found in Table 5.2.
Section 5.4

Data Partitioning

We additionally partitioned each dataset into subsets based on the sentence or sentence pair’s complexity, their average lengths, and the position of the operative words to analyze how results would be influenced by these factors. In all three cases of partitioning, we take a look at the distribution of the partitioning score and split the data into a lower and upper half based on the median of the distribution. For example, we split the CrowS-Pairs dataset into short and long samples by looking at the distribution of sample lengths within the data. All samples which are less than the median sample length belong to the short sample subset and conversely all samples greater than the median sample length belong to the long sample subset. We further describe our quantification of each split below.

To quantify a sample’s complexity, we focus on number of clauses present across the sentence or sentence pair. Our metric is simply a count of the pronoun occurrences along with occurrences of the prepositions “that”, “which”, “where”, “why”, and “when”. We leave for future work an analysis which incorporates more sophisticated methods of determining a sample’s complexity, such as by using entity recognition or parsing the sentence to find the number of nouns present.

In partitioning the datasets by average lengths, in the case of the samples having two sentences, such as in CrowS-Pairs and StereoSet, we take an average of the number of words in both sentences. Since the BUG dataset only includes one sentence per sample, we just take the number of words in the sentence as the length score. To count the number of words in a sentence, we simply split the sentence on white space.

To quantify the position of operative words, we take one approach for the BUG
5.4 Data Partitioning

Datasets

data and another for the CrowS-Pairs dataset. The BUG dataset includes the distance of the operative word, the pronoun, from the subject of the sentence, and this is used as a samples position score. To determine the CrowS-Pair position scores, we take the index of the operative word in the sentence. Because the sentence pairs are minimally distant, we determine the operative word by taking the difference of the two sentences, finding the index of the respective different words in both sentences, and averaging both indices. We did not come up with a position score implementation for StereoSet.
Chapter 6

Methodology
6.1 Gradient Based Probing

In our exploration of the contributions of each attention head, we analyze the gradients of each head [20, 32, 8] and the difference in accumulated hidden states and attention distributions at each layer in the model when an operative input token is masked or replaced [17, 2]. For the remainder of this paper, we refer to the gradient-based approach as gradient probing, and we refer to the masking and replacement analysis as leave-one-out probing. Given the monolingual nature of the datasets, we restrict our analysis to Bert-Base [6].

Section 6.1

Gradient Based Probing

When ranking heads based on their gradients, we finetuned the BERT-base model on different tasks. We train the model for 4 epochs with batch size of 8. The model is trained with the cross-entropy loss function described in Equation 3.5 and evaluated on the F1-macro score described in Equation 3.7.

In the CrowS-Pairs dataset, we examined three settings. The first was shuffling the sentence pairs and finetuning the model to predict which sentence was the more stereotypical one. The second setting was finetuning the model to determine the stereotypical direction of each pair, whether the pair demonstrated or violated a stereotype about a historically disadvantaged group. Lastly, we finetuned the model to predict the domain of the stereotype, one of the nine categories listed in Section 5.1.

In the StereoSet dataset, we evaluated two settings. The first was to have the model predict the sample’s bias type, and the second was to train the model to determine whether the sentence pair was a stereotype, anti-stereotype, or unrelated.

In the BUG dataset, we trained the model to determine whether or not the sentence was a stereotype or anti-stereotype.
6.2 Leave-One-Out Probing

When partitioning each dataset into its six subsets, as described in Section 5.4, we trained the model to predict whether the sample was a stereotype or an anti-stereotype. We chose this setting as this was the consistent annotation across all three datasets.

For each setting in the gradient-probing approach, we finetune the model with three random seeds (1998, 2018, and 2022). These seeds will affect the initialization of the pooler and fully connected layers outside of the BERT model, shown in Figure 3.3, and can have an effect on the gradients within BERT. We finetune the model across random seeds to ensure that the rankings are robust to this randomness.

Section 6.2

Leave-One-Out Probing

The two main approaches we used in leave-one-out probing were replacing operative words with the MASK token and replacing them with their antonym. In each case, we send two samples to the model, one where the sample appears as it does in the dataset and another where the sample is either masked or where certain words are replaced with their antonyms.

Here we examine the difference in attention distributions and hidden states in the model between these two passes. When examining the differences in attention distributions, we look at the correlation between the distributions from the operative word in both passes and the correlation between distributions to the operative word in both passes. Heads with low correlation are ranked highly. We use \(1 - \rho\) to quantify this difference, where \(\rho\) is the Spearman correlation of the two distributions. The Spearman correlation \(\rho\) of two distributions is

\[
\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}
\]
where $n$ is the number of observations and $d_i$ is the difference between ranks of the observation at $i$. Ranks for the data are from highest to lowest observation. When examining the differences in hidden states, we look at the cosine distance between the hidden state at the [CLS] token between both samples and the mean hidden state across all tokens between both samples. To find the antonym of words, we used WordNet [7].

We implement leave one out probing on the entirety of each dataset, as well as within the subset of each dataset which promotes a stereotype and the subset which promotes an anti-stereotype.

### 6.2.1. Masking

For the masking experiment, we chose the following operative words to focus on in each dataset. In the StereoSet data, we defined the target word as the operative word for masking. This was the word which set the context for the subject of the sentence pair. For example, in the sentence pair *Many people live in Ethiopia [SEP] The people are fat and unathletic*. Ethiopia is the target word. In the BUG dataset, we defined the operative word as the profession in the sentence. This was the word to which the pronoun referred. For example, in the sentence *A librarian shares her views on information literacy* librarian is the profession and her is the pronoun which refers to it. In the CrowS-Pairs dataset, we simply examine the changes when the first (stereotypical) sentence is passed and when the second (anti-stereotypical) sentence is passed. We define the operative word

In this setting, we pass two samples to the model, one sentence with masking and one without. We record the attention distributions from the operative word in each case and take the correlation of these distributions. We do the same for the attention distributions to the operative word.
6.2.2. Replacing Words of Interest

To convert StereoSet samples, we parse the sentence and replace all adjectives with their antonyms. In the BUG dataset, we simply flip the gendered pronouns to the opposite gender. Similar to the setting in the masking approach, since samples in the CrowS-Pairs data are minimally distant and one sentence promotes a stereotype while the other promotes an anti-stereotype, we use the stereotypical and anti-stereotypical sentences as the “normal” and antonym sentences, respectively.
Chapter 7

Results
A notable result from our experiments was the high gradient values on the 5th and 9th heads across multiple layers. Examples of these are demonstrated in Figure 7.1. These high gradients suggest that these attention heads are particularly useful and contribute substantially to the model’s reasoning and prediction when detecting stereotypes. This may suggest that these heads are particularly useful at extracting and identifying features which are useful for stereotype detection.

Additionally, we find that attention heads in higher layers exhibit higher gradient values in longer and more complex sentences. Indeed, 40% of the top 10 attention heads in each experiment on more complex sentences reside in the top layer while 1.1% reside in the bottom layer, while in the top 10 attention heads across less complex sentences experiments, these numbers are 20% and 15.56%, respectively. Similarly, we find that when analyzing longer sentences, 37.78% of the top 10 attention heads reside in the top layer compared to 2.22% in the bottom layer. This is in contrast to the distribution on shorter sentences, which are 23.3% and 15.5% respectively. This may suggest that attention heads on higher layers are better able to capture long range dependencies within sequences which are useful for stereotype detection, while attention heads on the lower layers are better able to capture more immediate context. Figure 7.3 demonstrates the layerwise distribution of the attention heads when considering the different lengths and complexities of the sentences.

Attention heads in the bottom layers of the model are more sensitive to masking input perturbations while attention heads in the top layers of the model are more sensitive to changes in the input where words of interest are converted to their antonyms. The distribution of the top 30 attention heads of each experimental setting is demonstrated in Figure 7.4. These rankings are determined from the leave-one-out probing experiments. This suggests that the upper layer heads are key to extracting syntactic, grammatical features. For example, they may be able to detect
Figure 7.1: Clockwise from the top left, the top 10 attention heads are plotted when the model is finetuned on the BUG dataset to predict whether the sample is a stereotype, on the CrowS-Pairs dataset to predict the stereotypical direction of the pair, and on the StereoSet dataset to predict whether the sentence pair promotes a stereotype or not. The heads are plotted across the results from the three different random seeds, with darker colors indicating that the head was ranked highly across more experiments. As is demonstrated, the 5th and 9th attention heads and heads across multiple layers of the model are ranked highly.

Figure 7.2: The layerwise distribution of the top 10 attention heads across all experimental settings ranked using the gradient probing approach. Attention heads in the 12th are the most prevalent, with 31.06% of the heads appearing in this layer of the model.
Figure 7.3: The distribution of the top 10 attention heads in each dataset split by layer. The figure on the left demonstrates the layerwise distribution of attention heads when the model was finetuned on the longer and shorter samples across the three datasets. The figure on the right shows the distribution when the model was finetuned on the more and less complex samples. Higher layer attention heads are deemed important on the more complex and longer sentences, and the distribution is more evenly across layers in the less complex and shorter sentences.

Figure 7.4: The layerwise distribution of the top 30 attention heads of each setting on each dataset as ranked by the change in attention distribution when the input is altered by the labeled method. Attention heads in lower layers of the model, closer to the input, are more sensitive to the input being altered via word replacement, while attention heads in higher layers of the model are more sensitive to the input being altered via masking.
noun-adj pairs relevant for stereotype detection which are absent when words are replaced with a mask. The lower level attention heads on the other hand, might be more important in capturing lexical-semantic features related to words, such as sentiment expressed for stereotype detection.

A noteworthy result common to all settings in this experiment was the stability of the hidden states on the 9th head across multiple layers, resilient to input perturbations. Figure 7.5 demonstrates the change in hidden states between input perturbations across the datasets. This observation indicates a lack of change in encoding behavior between different inputs on the 9th head across multiple layers. This is not the case for the 5th head across layers, though from the gradient probing rankings imply that these heads are both important to the model’s reasoning. This suggests that the role of the 5th and 9th heads are different in this task. For example, the 9th head may encode information that is resistant to input perturbations, such as part of speech information, while the 5th head may encode contextual information about specific words used in the sentence. Both information are leveraged to support the model’s reasoning process.
Figure 7.5: The cosine distance between the hidden state of the [CLS] token when the model reads the sentence with and without antonym replacement. From left to right, the cosine distance when the model reads from the BUG dataset, from StereoSet, and from CrowS-Pairs. Darker colors indicate a higher cosine distance implying a greater sensitivity to input perturbations, and lighter colors indicate a lower cosine distance. In each case, it is evident that the 9th head remains stable across multiple layers.
This work is a checkpoint of a longer term study which intends to build upon these findings and test additional probing mechanisms aimed at interpreting the importance of different attention heads in BERT. One such additional mechanism is to rank the attention heads similarly to the gradient-based settings but using minimum description length (MDL) probing to generate scores [34]. MDL probing approaches probing from an information-theoretic perspective. Scored is the minimum description length necessary to encode hidden states of the model to their respective target values.

Future work can also examine the difference in attention head rankings between pretrained models and finetuned models on the leave-one-out approach. Attention heads may learn to extract or encode certain features in the finetuning process, and as a result their rankings may change. This may give us further insight as to which attention heads are crucial for the model to learn what determines a stereotype.

An interesting question to research may also be the similarity between attention head rankings among samples which is labeled to be of the same bias type. This could provide further understanding of which heads are relevant for semantic understanding of the sentence, specifically for which sorts of biases are certain heads more important.

Lastly, it may be interesting to do similar experiments on other popular trans-
former based models to determine the similarity or difference between attention head behaviors on stereotype detection.
Conclusion

We conduct a study which aims to determine the relative importance of attention heads in stereotype detection tasks, and we utilize gradient-based analysis and input perturbation to rank their relevance. We run experiments on three diverse stereotype datasets, two which are crowdsourced and generated by annotators and one consisting of sentences sampled from large corpora.

We find that the gradients on the 5th and 9th attention heads across many layers are high, suggesting that these attention heads contribute substantially to the model’s decision making process. However, we find that the hidden states on the 9th heads are stable across altered inputs while the 5th heads do not exhibit this same stability, implying that their role in the model’s reasoning are different. Attention heads in the higher layers more consistently support the model’s reasoning, however these layers’ importances are most pronounced when analyzing samples with high number of pronouns or longer sequence length. Conversely, when sequences are shorter attention heads in the 1st layer become more important. Attention heads on higher layers are also more sensitive to input changes where words are removed, as opposed to attention heads on lower layers, which are sensitive to input changes when the meaning of the sentence is altered.
Bibliography


