Experiment Design for Hypotheses About How NLP Models Work

Sofia Serrano

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2024

Reading Committee:
Noah A. Smith, Chair
Yulia Tsvetkov
Sheng Wang

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Experiment Design for Hypotheses About How NLP Models Work

Sofia Serrano

Chair of the Supervisory Committee:
Noah A. Smith
Computer Science and Engineering

In the last few years, Natural Language Processing models have come a long way. However, for all the work that continues to report performance improvements, we still see different lines of work identifying problems and undesirable behavior in our most current modeling approaches. Why is this? We argue that this is a symptom of not putting enough emphasis on understanding our models, since knowledge of our models' functioning expressed beyond test-set performance helps us speak to how models might generalize, as well as their limitations. To facilitate discovering this knowledge about how our models work, establishing trustworthy, precise methods for *how* we go about testing such hypotheses is of critical importance. Here, we focus specifically on the design of experiments for hypotheses about, or explanations of, models' observed behavior.

We discuss three projects that have examined different kinds of questions in this space. The first two demonstrate experiment design for different granularities of hypotheses about the functioning of NLP models, while the last project investigates which aspects of a particular experimental design choice can skew findings.

The first two of these projects pose questions about whether a model as a whole exhibits a particular trait and whether a certain mechanism within many NLP models can be interpreted as instance-level explanations, respectively. Specifically, we first model the design of an experiment to investigate whether lexical

correlations in the training data transfer to models finetuned on that data. Using the designed method, we find bias in the models reflecting that in the training data, even when that training data has been rebalanced to mitigate those biases. This offers further implications regarding the strong ability of contemporary NLP models to leverage higher-order features. The second of these projects, in contrast, investigates whether a particular component of many NLP models, the attention mechanism, functions as a descriptor of which information was most important in producing the model's output for a particular input, finding gaps between models' calculated attention distributions and the corresponding importance of inputs to the attention module.

Meanwhile, the third project we discuss instead tests the impact of varying a key part of a common experimental design in choosing (or advocating for new) methods that explain which input information models use. In particular, for experiments that test the ability of an explainability method to recover a known ground truth about which input information *must* have been used to make a downstream decision, we examine the impact of the kind of known-ground-truth test sets used on such an experiment's results.

Finally, we close with a discussion of future work centered on examining the impact of other kinds of common experiment-structuring choices in this space.

Acknowledgements

Goodness, this has been a long road. This thesis would not exist without the help of so many people. By rights, you should each get your own individual essay about how you've helped me through the last few years. And naturally, I've waited to write the acknowledgements until a point where I really have to get these written and my brain is mostly fresh out of words. Please do me a favor and interpret the relative brevity of what follows as me throwing my hands up and just pushing a pint of ice cream at you partway through my speech.

Noah: You've been the best advisor anyone could ask for. Frankly, it's outrageous how good you are at this job. I'm going to be spending my career trying to emulate your advising, and I've made my peace with that. Thank you for everything.

Fellow ARK members: I've been so fortunate to be surrounded by such brilliant, kind people during my time at UW. Sam, Dallas, Jesse, Swabha, Sarah, Emily, Nelson, Phoebe, Maarten, Liz, Kelvin, Roy, Hao, Rik, Rahul, Suchin, Leo, Mourad, Lavinia, Xiaoning, Ananya, Zhaofeng, Nikos, Judit, Jungo, Nikita, Ofir, Yizhong, Alisa, Wenya, Yanai, Hila, Phillip, Weijia, Oreva, Leon, Zander, Andrey, Margaret, Sachin, Yushi, Victoria, and Guang, your wise words about research will stick with me.

Lucy: Thank you for being a voice of reason through this whole thesis-writing process. I'm honored to have been privy to the very best weekly gossip about what some actors in NYC have been up to. Thanks for injecting some delight into my Friday mornings.

Lolo, Mom, and Dad: Thanks for the Mozy photos, the Real Madrid updates, and the pokégifts. You guys are the best, and as grandma would say, I'm totally objective!! (Incidentally, you are also the only people who will know what I mean when I describe my thesis-writing process as OLACOCOLA.)

Maryam: Thanks for the camaraderie, and for having my back through the PhD process. It makes a real

difference for me to know you're out there rooting for me in Chicago.

Anni, Ben, Jinhong, Lauren, Leah, Madi, Natalie: You all hold the distinction of being friends I talk to about anything. I expect to have the impulse of "The ferns must know" for a very, very long time to come. Thanks for humoring me as I talk your ear off about anything from minor league baseball mascots to musings about roombas.

Callie and Dolma, I would also be remiss not to acknowledge your help in writing the thesis, even though I know it was sometimes frustrating that thesis-writing cut into your belly-rub time.

And finally, Gabe, Willie, Chris, Erin, Matt, Tyler, Tal, Kim (and Rose): I've been so, so lucky to be a part of the best D&D group in the world. There's no one else with whom I'd rather get crushed by a hut. If anyone ever gets around to inventing teleporting, please know that my first thought will be of restarting weekly in-person potlucks and game sessions. You're the ones who have made Seattle feel like home over the last few years.

Much love to you all.

DEDICATION

To my family, ferns, and the D&D crew

Contents

List of Figures

List of Tables

Chapter 1

Introduction

NLP models have improved dramatically over the last several years for a staggering variety of tasks [Läubli et al., 2018; Liu et al., 2019a; Brown et al., 2020, e.g.]. In many ways, they seem to produce more reliably high-quality outputs than ever. However, after the introduction of new modeling approaches, we often see subsequent work identifying issues in these models such as bias, degeneration of output generations, and problems shifting between seemingly similar domains [e.g., Sheng et al., 2021; Li and Bamman, 2021; Liu et al., 2022; Lahnala et al., 2022]. Why is this? We argue here that this staggered identification of model issues is a symptom of not sufficiently understanding our models.

In some ways this can be framed as a problem of evaluation. What we ultimately want when we develop an NLP model is for that model to perform well in some sense, whether we take that to mean achieving high accuracy, demonstrating usefulness, or something else. If only we could perfectly articulate everything we wanted to be true about a model and express it in held-out test data and some evaluation metric(s) over that held-out dataset, then we could simply apply a model to that data, calculate our metrics, and know all its benefits and shortcomings. But as we move to increasingly complex tasks and applications, it becomes infeasible to exhaustively enumerate *all* our desiderata, weight them correctly, and translate those into data truly representative of the overall task space. Indeed, gaps between performance as computed on test sets and "real-world settings" those test sets were intended to simulate are well known [Marcus, 2018]. How can we compensate for these gaps, then? It's instructive to look at a relatively early argument that Doshi-Velez and Kim [2017] made about the importance of interpretability in machine learning: that it accounts

for "incompleteness in the problem formalization." In other words, understanding of how a model works can allow us to say something about how that model operates beyond its performance on our particular test inputs. This, in addition to other benefits such as increased transparency of models' decision-making processes and insights into the kind of information in the input that is helpful for solving the assigned tasks, is what makes the ability to understand our models so important.

There are many different aspects to understanding models, but in this work we focus specifically on investigating the fit of a hypothesis to an underlying complicated model. Note that this is in contrast to focusing on, say, users' comprehension of an explanation that sits between them and such a model; this second direction is also important and is often grouped together with the first under the term "interpretability" or "explainability," but it is a separate line of work.[1] The distinction between these two research directions also divides the kinds of experiments associated with them (with the exception of some work explicitly comparing or combining results from the two branches, such as Nguyen [2018] or DeYoung et al. [2020]). Jacovi and Goldberg [2020] highlight that certain experiment setups, namely those involving human evaluation, are useful for gauging what humans do with explanations of models but *not* for assessing the faithfulness of those explanations to the underlying model being explained. This point applies more generally even when the intermediary between us and a model is a hypothesis not packaged as an explanation.

When designing an experiment for a hypothesis about an NLP model, what kinds of challenges do we face? In contrast to many experiments that we run in NLP more broadly, we often initially lack a clear answer as to how we would verify whether a model might have a particular trait (for example, "this model displays gender bias"). This means that a key challenge we face with every experiment, to a greater extent than many other areas within NLP, is *precise problem articulation*: "explainability" encompasses many different things, so which aspect(s) of explainability does the particular question we posed require us to focus on? How are we defining each term in our work, how does that differ from the aspects of that term on which past authors have focused, and how does that affect our ability to use elements (metrics, datasets, human evaluations, etc.) from their existing work? The design of these experiments also foregrounds the perpetual challenge of evaluation in NLP. Even once we've posed our research question, how will we convince ourselves of a

[1] There are still other lines of work that fall under the umbrella of explainability; for example, Hooker et al. [2019] frame validity of an explanation as being tied to the underlying data, *not* a specific instantiation of a model trained on that data, but here we focus specifically on connections to models.

particular answer?

Testing how well a particular hypothesis fits a given model, while already distinct from certain questions under the umbrella of explainability as described above, still encompasses multiple branches of research. Some of these branches are focused on describing models globally, others focus on developing explainability methods, and still others focus on meta-analysis of the methods we use to examine models. Here, we focus on three projects that represent different kinds of questions within this area. One argues for a particular method of testing intersecting biases a model may have picked up; another constructs different experiments to test multiple facets of whether a model component could be construed as "interpretable"; and the other tests the impact of a particular facet of experiment design.

In more detail, the first project we discuss (chapter 2) examines broader patterns in model bias and the traits of the training data that might contribute to that bias. We consider the conditional distributions of labels associated with different features that should ideally be uninformative on their own (specifically, lexical features), using a new strategy for detecting this kind of bias to determine that uneven conditional distributions of this sort are reflected in models trained using this data. We then apply a simple optimization approach to reweight our training data and examine the resulting difference in detected bias on models newly trained on that reweighted data. Surprisingly, although we confirm that we largely do successfully reweight the data according to our features of interest, we still see a significant presence of the original data's bias in models only trained on the reweighted version. To explain this, we show that the bias mitigation strategy we demonstrate has adverse effects on higher-order features. We close this section with some discussions about the implications of our results on what it means to "debias" data, and the capability of modern NLP models to pick up biases from data even in cases where they've been mitigated.

We then turn to a second project where the underlying hypotheses about a model take a different form: instead of focusing on a high-level hypothesis about which information a model generally uses, the project in chapter 3 concentrates on the relation of an internal module's outputs to the determination of which information in a data instance was most important to making a model's decision, which is a common variety of hypothesis found in interpretability research. Specifically, we focus on attention mechanisms, which have boosted performance on a range of NLP tasks over the past few years. Because attention layers explicitly weight input components' representations, it has often been assumed that attention can be used to identify

information that models found important (e.g., specific contextualized word tokens). We test whether that assumption holds by manipulating attention weights in already-trained text classification models and analyzing the resulting differences in their predictions. While we observe some ways in which higher attention weights correlate with greater impact on model predictions, we also find many ways in which this does not hold, i.e., where gradient-based rankings of attention weights better predict their effects than their magnitudes. We conclude that while attention noisily predicts input components' overall importance to a model, it is by no means a fail-safe indicator.

Then, in chapter 4, we shift from designing experiments to test a hypothesis about a model to testing aspects of a common experiment design themselves. Instead of walking through the design of experiments for a hypothesis and illustrating the rationale behind them, we test the design of experiments directly. Specifically, we consider the common case where an NLP researcher or practitioner is deciding which previously proposed explainability method to use for investigating why a model is producing certain outputs. One way of vetting explanation methods in this situation is to use test data annotated with known ground truth about which information is *necessary* for producing a particular label or token downstream. (These test datasets must be carefully constructed, and focus on settings where determining such a ground truth about necessary information is possible.) The candidate explanation methods are then all deployed on a particular model performing inference on that test data, and whichever explanation method is best at tracing back to the known evidence in the input is the one to use. One question that's easy to overlook, however, is what *kind* of ground-truth-annotated data to use for conducting these tests. In this project, we specifically consider whether using in-distribution versus out-of-distribution test data matters, as well as the impact of using a test set where the nature of the annotated ground truth is syntactic, rather than semantic.

Finally, we discuss planned future work that concentrates on the impact of other kinds of common experiment-structuring choices in this space. While there is of course quite a bit of variety in how experiments in this space are structured as mentioned earlier, most experimental designs are not (and do not need to be) entirely unique. This means that there *are* commonalities between many experiments, and where such commonalities exist, it's important to develop guidelines for the details of how these experiments are implemented. Developing better practices for which kinds of data we use to test models, how the choice of one metric over another affects results, how to better simulate eventual deployment settings, etc. will all be

key to ensuring that the conclusions we draw about models are trustworthy, and that we know whether and how those conclusions generalize.

Chapter 2

Measuring the Impact of Data Bias on Downstream Models

We begin with work focusing on characterizing a model broadly, by looking at patterns in a model's behavior in aggregate over many instances at inference time. This chapter includes materials originally published in Serrano et al. [2023].

Machine learning research today, including within NLP, is dominated by large datasets and expressive models that are able to take advantage of them [Brown et al., 2020; Hoffmann et al., 2022; Touvron et al., 2023; Groeneveld et al., 2024]. At the same time, as the scale of training data has grown, this explosion of data has come at the expense of data *curation* [Penedo et al., 2023]; for many of the datasets currently in use today, human oversight of the full breadth of their contents has become unrealistic. This makes it more likely that training datasets contain undesirable associations or shortcuts to learning intended tasks. One key question is **whether these unintended biases in the training data propagate to models trained on that data.** Considerable research has posed similar questions of undesirable associations in data manifesting in models, whether through spurious correlations between lexical features and labels [Tsuchiya, 2018; Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019] or through gender or racial bias [Waseem and Hovy, 2016; Rudinger et al., 2018; Stanovsky et al., 2019; Davidson et al., 2019; Sap et al., 2019].

Overall, we see mixed results from the existing literature. Caliskan et al. [2017] determine that trained word vectors do pick up societal biases from their training corpora. Likewise, Rudinger et al. [2018] find

evidence of gender bias in coreference resolution systems, Stanovsky et al. [2019] find gender bias in machine translation systems, and Sap et al. [2019] find racial bias in hate speech detection models. However, whether *multiple* attributes' biases in data transfer to models is less clear. For example, Steed et al. [2022] find that both pretraining data and finetuning data have an effect on biases having to do with gendered pronouns and identity terms that are learned by occupation and toxicity classifiers, but that certain forms of bias reduction in either pretraining or finetuning data don't necessarily overcome bias that the model might pick up from the other. This is possibly explained by the results of Zhou and Srikumar [2022], who find that data used for finetuning largely distances clusters of textual representations by label without significantly changing other properties of the underlying distribution of data. In a similar vein, Joshi and He [2022] find that counterfactually augmented training data can actually exacerbate other spurious correlations in models.

For all the different results reported in this body of literature, there are some typical characteristics of the bias evaluation methodology they apply. It is common for this work to test for a *single* undesirable form of behavior (e.g., biased use of gendered pronouns). For example, Belinkov et al. [2019] focus on whether NLI models ignore input instances' premise, an important problem, but this also simplifies their evaluation, as they doesn't need to consider the potentially disparate impact of their adjusted model on intersecting biases. Another common characteristic is the creation of new and separate test data [McCoy et al., 2019; Zhang et al., 2019], on which decreased performance is taken to indicate bias [Tu et al., 2020; Wu et al., 2022]. A concern regarding this strategy, though, is that such test sets very likely still contain (undetected) biases of their own. Due to the complicated nature of natural language and the highly intertwined features that occur together in text, it is very likely that this will be true regardless of the test set created.

To get a sense of how models do or don't absorb biases in practice, designing a scenario where many such biases naturally intersect is crucial. Therefore, we structure our experiments to facilitate those intersections. Specifically, we need:

1. To operationalize bias in a way that makes testing our hypothesis possible (section 2.1).
2. A way of testing whether a model exhibits hypothesized biases (section 2.2).
3. A way of dramatically lessening those biases in the training data while keeping everything else as equal as possible (section 2.3).

4. To apply the test for bias to two types of models: those trained on data found to contain many intersecting biases, and those trained on the version of that data to which our bias mitigation method has been applied.

We first lay out our definition of bias, which is designed to allow for automated (and therefore scalable to small training set size) identification of many different kinds of potential intersecting biases.

We then introduce an approach to testing for undesirable model biases that can operate using existing held-out data, even though that data might itself have spurious correlations. In particular, we repurpose the classic permutation test to examine whether observed differences in model performance between instances exhibiting more common feature-label pairings and those exhibiting less common feature-label pairings are statistically significant.

To simultaneously cut bias associated with our many intersecting features of interest, we apply an optimization-based approach to reweighting the training instances. The approach brings uneven label distributions closer to uniform for thousands of different intersecting lexical features, many more than we use for our model bias evaluation, and still manages to have a strong effect on the most initially biased features despite our reweighting approach not focusing on those in particular.

For our experiments, we focus on the simplest kind of feature-label association: correlations between lexical features and task labels. We select two tasks (natural language inference and duplicate-question detection) for which any such lexical feature should be uninformative on its own. Finding strong evidence that models finetuned on three different datasets have at least some of the same lexical biases that exist in their training data, we then examine the extent to which those biases are mitigated by lessening biases in the training data, using our optimization-based method. We then finetune new models on those (reweighted) datasets. We find that although model bias lessens somewhat when we do this, we still see strong evidence of bias. Surprisingly, this holds even when we consider models that make use of no pretraining data.

We close with a discussion of possible factors contributing to these results. We first note that perhaps the continued relative lack of variety of minority-class examples containing certain features hinders the reweighted models' ability to generalize their recognition of those less-common feature-class pairs, even though the combined weight given to those few instances in the loss function is increased. However, when we examine the effect of our reweighting on higher-order features (namely, bigrams), we see another prob-

lem: the same reweighting that mitigates associations between unigrams and any particular label actually strengthens associations between bigrams and certain labels in data. Based on this observation, we arrive at two conclusions: (1) simultaneously reducing bias across features of different levels of granularity for natural-language data is likely not feasible, and (2) even if we aim to mitigate model bias *only* with respect to simple features, if we do so by reweighting the data, the high-capacity models used in modern NLP are still capable of learning the spurious correlations of the original unweighted data through associations that remain encoded in more complex features even after reweighting. We conclude that bias reduction in NLP cannot be cast purely as a "data problem," and solutions may need to focus elsewhere (e.g., on models).

2.1 What Do We Mean by Bias?

The term "bias" is polysemous, having been adopted by different communities to mean different things, from historically rooted social inequity to skewed model evaluations [Mehrabi et al., 2021] to techniques that help with supervised class imbalance in labels [Chen et al., 2018]. In our work, we use "bias" to mean correlations between individual input features and task labels.

This framework is fairly general, but our focus in this work is natural language data. Therefore, as an example to illustrate our definition of bias, we will refer to correlations between the presence of individual word types in the input (unigrams) and a given label in a classification task.

More formally, consider a task of mapping inputs in $\mathcal{X}$ to labels in $\mathcal{Y}$. We assume a training dataset $\mathcal{D} = \langle (x_i, y_i) \rangle_{i=1}^{n}$, each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. We are particularly interested in a designated collection of d binary features on $\mathcal{X}$, the jth of which is denoted $f_j : \mathcal{X} \to \{0, 1\}$. For example, f_j might be the presence of the word "nobody" in an instance. Let $f_{j,i}$ be shorthand for $f_j(x_i)$ (e.g., whether instance x_i contains the word "nobody" ($f_j(x_i) = 1$) or not ($f_j(x_i) = 0$)).

Introducing random variable notation, we can characterize $\mathcal{D}$ by its empirical conditional distribution over labels given each feature, such that for all $y \in \mathcal{Y}$,

$$\hat{p}(Y = y \mid F_j = 1) = \frac{\sum_i \mathbf{1}\{f_{j,i} = 1 \wedge y_i = y\}}{\sum_i \mathbf{1}\{f_{j,i} = 1\}}.$$

If the conditional distribution of output labels given the presence of a particular lexical feature is very

different from the overall label distribution in the data, we consider that feature to be biased in the training data.

Importantly, note that as long as the presence of a particular feature can be determined automatically in an instance, this definition of bias can also be calculated automatically. This will allow us to determine biases at scale for many, many features.

2.2 Measuring Bias in Model Performance and Data

Running experiments to determine the effect of a particular training dataset on model bias requires a reliable way to identify whether that model bias is present. In this section, we describe our method for doing so.

Recall that when $\hat{p}(Y = y \mid F_j = 1)$ is close to 1, it means feature j is correlated with label y in a given dataset. Let us denote the set of examples that contain feature j and have the label most strongly associated with feature j in $\mathcal{D}$ by $\mathcal{U}_j$, which we call the "usual-labels" set. Then, denote the examples that contain j but have a *different* label by $\mathcal{N}_j$, which we call the "unusual-labels" set.

To build intuition, the accuracy of the model on instances which contain feature j is the accuracy over the union $\mathcal{U}_j \cup \mathcal{N}_j$. However, to measure if the model is picking up bias from the data, we will measure accuracy over $\mathcal{U}_j$ and $\mathcal{N}_j$ separately. To maximize accuracy on $\mathcal{U}_j \cup \mathcal{N}_j$ the model would be justified in disproportionately labeling instances containing f_j with y, so we can't use accuracy by itself to measure model bias. Instead, the key idea here will be to look for differences in error rates between instances whose labels align with features' training biases (the "usual-labels" set), and instances whose labels do not.

If the model has learned a biased representation of the data, we expect it to have higher accuracy on the "usual-labels" set, $\mathcal{U}_j$. On the other hand, if the model hasn't learned that bias, we would expect the correct predictions to be uniformly distributed between $\mathcal{U}_j$ and $\mathcal{N}_j$. We use this as the basis for a hypothesis test: the null hypothesis H_0 is that the accuracy of model is the same on both sets $\text{ACC}(\mathcal{U}_j) = \text{ACC}(\mathcal{N}_j)$, and the alternative hypothesis H_1 is that $\text{ACC}(\mathcal{U}_j) > \text{ACC}(\mathcal{N}_j)$. That is, if the errors are distributed uniformly at random, how likely is it that $\mathcal{U}_j$ would have *at least* its observed number of correct instances?

2.2.1 Permutation Test

Given a model's accuracy on $\mathcal{U}_j$ and $\mathcal{N}_j$, and the size of the two sets, we can calculate the p-value for this hypothesis test exactly using the permutation test [Phipson and Smyth, 2010]. Our null hypothesis is that the errors are uniformly distributed between $\mathcal{U}_j$ and $\mathcal{N}_j$, so the permutation test calls for randomly shuffling whether a given instance is correctly labeled, while not changing the number of instances in each category *or* the model's overall accuracy on the set union, both of which change the shape of the distribution of correct instances that we'd expect to see, but neither of which is the property for which we're testing. As there are finitely many ways to shuffle whether a given instance is correctly labeled, this test also has the benefit of having a closed form, giving us an exact p-value.[1]

2.2.2 Calculating Bias over Multiple Features

In the previous section we described how we could use a permutation test for a single feature f_j. Here we describe how to apply this to the full dataset. We define $\mathcal{U}$ as $\cup_j \mathcal{U}_j$ and $\mathcal{N}$ as $\cup_j \mathcal{N}_j$ for 50 features f_j per distinct label (namely, those that demonstrate the highest association with that label in the training data), so 100 or roughly 150 features f_j total depending on whether the dataset is 2- or 3-class ("roughly" because some features are among the most associated for two classes in 3-way classification). Given that each example x_i includes multiple features (e.g., $f_{j,i} = 1 \wedge f_{k,i} = 1$) it's possible for example x_i to have label y, which is the "usual-labels" for f_j but an "unusual-labels" for f_k. When this happens, we add it to both sets $\mathcal{U}$ and $\mathcal{N}$, meaning that their intersection is not necessarily empty. Pooling examples in this way allows us to run a single hypothesis test for whether or not the model learns bias from the dataset, avoiding the multiple-comparisons issue of running one hypothesis test for each feature. This procedure is described in Figure 2.1.

[1] For simplicity, we assume here that the model has an equal likelihood of guessing any of the output classes. In practice, this is approximately accurate for the data on which we experiment, though this assumption could be removed in principle by multiplying each permutation by a corresponding probability.

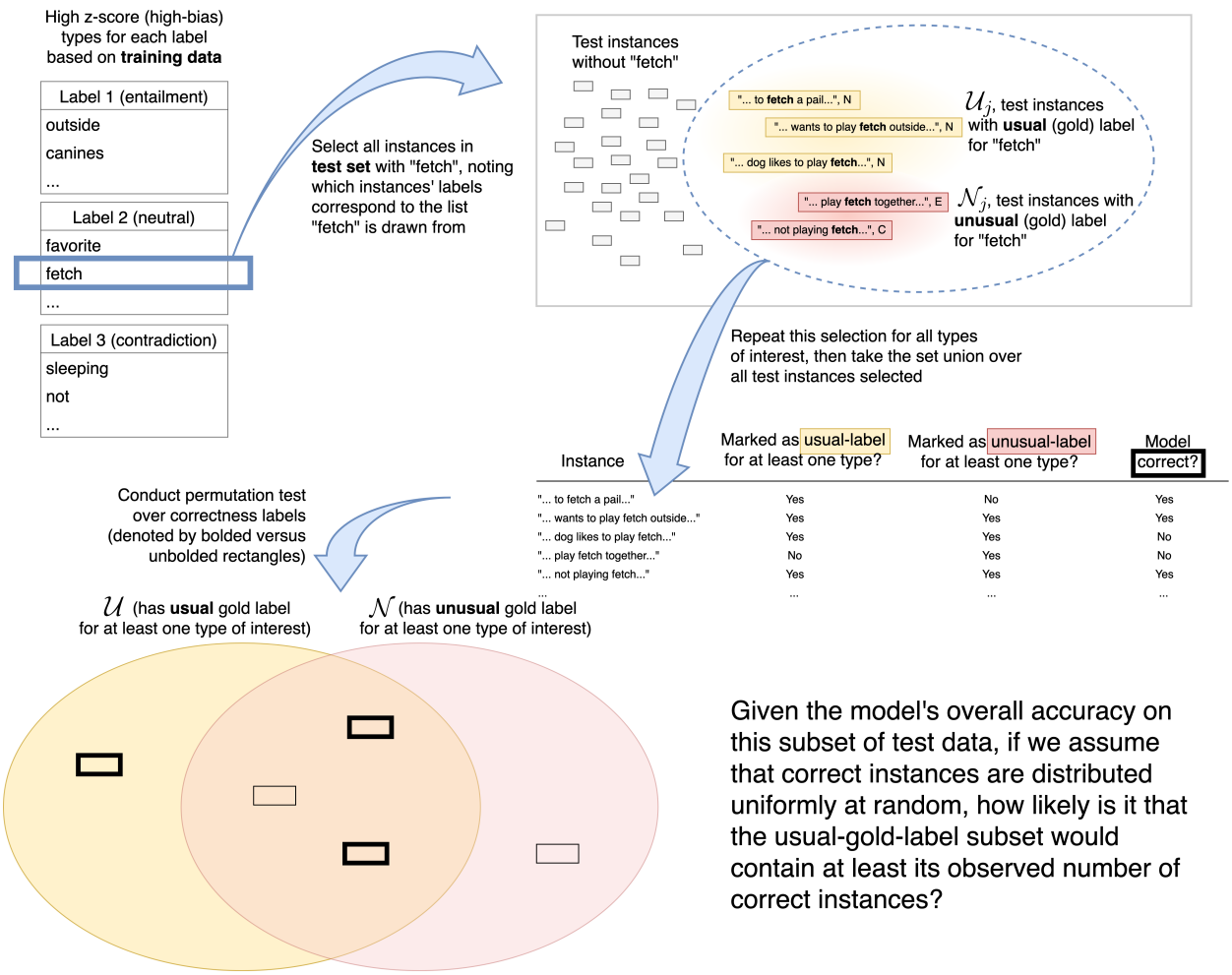


Figure 2.1: The setup of the permutation test that we use to test for bias in models trained on biased data, which in this figure uses word types as features and natural language inference as the underlying task.

2.3 Intervening on the Data by Balancing It

Now that we have a method for testing a model for particular biases in aggregate, we need to establish the control and test settings for our experiment. The difference between these two settings hinges on having access to two versions of the same dataset, each with all the same informational content as the other, except for one of the two lacking all the many intersecting forms of bias we enumerate.

Our strategy to produce that second version of a dataset is to intervene on the data to lessen lexical bias.[2] While modifying the data is only one family of approaches towards reducing eventual bias of a

[2]Note, we do not describe our approach as "removing bias," as natural language data in general is biased to some extent; see the argument made by Schwartz and Stanovsky [2022].

learned model (see, for example model-based strategies such as those proposed by Clark et al., 2019, or Karimi Mahabadi et al., 2020), recall that our goal here is to investigate the effect of the finetuning data on the rest of the training setup, so for our purposes we keep the rest of the training procedure the same.

Prior work has explored different ways of intervening on data, such as manual data augmentation [Zhao et al., 2018; Zhang and Sang, 2020; Gowda et al., 2021; Lee et al., 2021], or occluding bias in the original data [Feldman et al., 2015], but along very few different axes of bias. Other work augments minority-class data for the purpose of addressing class imbalance [Chawla et al., 2002]. Yet others have taken the approach of generating new data to augment the existing data in ways that counteract certain biases [Wu et al., 2022]. However, this last work relies on model-generated text, which, as Wu et al. [2022] themselves acknowledge, could differ from human-generated text in ways that aren't immediately obvious [Zellers et al., 2019].

In order to avoid potential new artifacts introduced by using machine-generated training data, and to improve the label balance in aggregate for a large volume of features simultaneously, we reweight existing training data such that in expectation, the disproportionate association of lexical features with certain labels is decreased. Reweighting data to remove bias is not a new idea—Kamiran and Calders [2012] do this through downsampling—but typically such approaches have considered at most a handful of different axes of bias. Some existing work, namely Byrd and Lipton [2018] and Zhai et al. [2023], has pointed out the limitations of approaches based on reweighting data, but again based on reweighting along comparatively few axes (in the case of the former) or on simpler model architectures than we consider here (in the case of the latter), so in the absence of a viable alternative meeting our requirements, we proceed with reweighting as our form of intervention for our experiments.

Typically, training datasets like $\mathcal{D}$ are treated as i.i.d., representative samples from a larger population. Formally, we instead propose to *weight* the instances in $\mathcal{D}$, assigning probability q_i to instance i, such that, $\forall j, \forall y \in \mathcal{Y}$,

$$\frac{\sum_i q_i \cdot \mathbf{1}\{f_{j,i} = 1 \wedge y_i = y\}}{\sum_i q_i \cdot \mathbf{1}\{f_{j,i} = 1\}} = \frac{1}{|\mathcal{Y}|} \tag{2.1}$$

From here on, we denote the lefthand side of Equation 2.1 as $q(y \mid F_j = 1)$. Note that, for simplicity, we assume a uniform distribution over labels as the target, though our methods can be straightforwardly adapted to alternative targets.

Given an algorithm that produces a weighting $q_1, \ldots, q_n$ for dataset $\mathcal{D}$, we quantify its absolute error with respect to Equation 2.1 as

$$\text{Err}(q) = \frac{1}{(\text{number of features}) \cdot |\mathcal{Y}|} \cdot \sum_j \sum_{y \in \mathcal{Y}} \left| q(y \mid F_j = 1) - \frac{1}{|\mathcal{Y}|} \right|$$

How do we choose these q_i values? We can state the general problem as a constrained optimization problem.[3] We seek values $q_1, \ldots, q_n$ such that:

$$\sum_{i=1}^{n} q_i = 1 \tag{2.2}$$

$$q_i \geq 0, \ \forall i \tag{2.3}$$

$$q(y \mid F_j = 1) - \frac{1}{|\mathcal{Y}|} = 0, \ \forall j, \forall y \in \mathcal{Y} \tag{2.4}$$

(The constraints in the last line are derived from Equation 2.1; strictly speaking one label's constraints are redundant and could be removed given the sum-to-one constraints.)

Using this setup, we seek a vector q that satisfies the constraints. We do this by minimizing the sum of squares of the left side of Equation 2.4; the approach is simplified by a reparameterization:

$$q_i = \frac{\exp z_i}{\sum_i \exp z_i}$$

This is equivalent to optimizing with respect to unnormalized weights (z_i) that are passed through a "softmax" operator, eliminating the need for the constraints in Equations 2.2 and 2.3. Once we have q, we multiply each x_i's contribution to the loss during training by $q_i \cdot |\mathcal{D}|$.

We apply this algorithm to reweight the following training datasets: SNLI [Bowman et al., 2015], MNLI [Williams et al., 2018], QNLI [Wang et al., 2018], and QQP.[4] In contrast to the <200 features per dataset that we use for evaluation of bias in models, when reweighting data, we used all types that appeared at least 100

[3]The slightly simplified formulation we present here for ease of reading only takes into account cases where feature j appears somewhere in our data, but Equation 2.4 can be straightforwardly modified by multiplying it by the denominator of $q(y \mid F_j = 1)$ to account for this.

[4]Quora Question Pairs dataset (QQP): `data.quora.com/First-Quora-Dataset-Release-Question-Pairs`

	$\lvert\mathcal{D}\rvert$	# Features	$\lvert\mathcal{Y}\rvert$	Err(Uniform) ($\downarrow$)	Err(Adjusted q) ($\downarrow$)
SNLI	549,367	3866	3	0.057	0.040
MNLI	392,376	6854	3	0.022	0.084
QNLI	104,743	3770	2	0.042	0.012
QQP	363,831	4386	2	0.154	0.047

Table 2.1: The average absolute difference between the empirical fraction of label y in instances with any particular unigram feature j and the total weight given to label y in the full training data, computed over all features and all their label values. Lower is better.

times in their corresponding training data as features, and we denoted an "instance" as the concatenation of a paired premise and hypothesis (or, for QQP, the concatenation of the two questions). We removed features from consideration if they did not have at least one document in the dataset for each of their labels.[5]

We see in Table 2.1 that by solving for distributions q over the different datasets as described, we successfully reduce Err(q) compared to the initial uniform weighting for all datasets except MNLI.[6] This leaves us with three successfully reweighted datasets with lessened unigram bias overall, and we can use these to investigate possible reduction of lexical bias compared to their original, uniformly-weighted counterparts. We confirm that for the high-z-score features used for model bias evaluation for each of these three, their label balance in the data either improves (often dramatically) or stays comparable as a result of our reweighting q. (Here and elsewhere, we use "label balance" of a feature to refer to the average absolute difference between its empirical label distribution in the training data and the overall label distribution of the training data, averaging elementwise over each possible label.) For example, see Figure 2.2 for the change that our reweighted q makes in improving the label distributions of our original high-z-score features from SNLI that we use for evaluation.

2.4 Running the Experiments

With our test for bias and our method of producing debiased versions of training datasets both established, here we shift our focus to particular tasks and datasets, in order to apply our test from section 2.2 in practice.

[5]This was not the case for any features in MNLI or QNLI, but applied to the word "recess" for SNLI, and the words "gobi" and "weakest" for QQP.

[6]MNLI is unusual among the datasets we studied in its remarkably low degree of lexical-feature bias to begin with, so it is perhaps not surprising that further lowering that bias across thousands of features proves difficult.

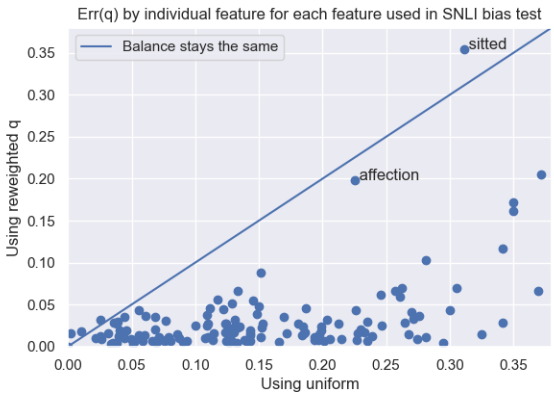


Figure 2.2: Label balance of the 137 lexical features used in our *model* bias evaluation for SNLI (since a handful of the highest z-score features in the training data didn't appear in the test set), using a uniform weighting and reweighed using q. q produces a lower Err(q) for most of these features and is comparable for most of the remaining few, even considering that the reweighting was with respect to all 3,866 features. We have labeled the only two features that go against this pattern.

2.4.1 Determining Biased Features (and Tasks)

For our experiments, we want a large volume of features that should ideally exhibit no correlation with labels. In order to get a large number of features, we'd like them to be simple and easy to automatically detect, so unigram features again come to mind, guiding our selection of tasks and datasets for experiments.

When is the association of unigram features with a particular label a problem? While previous work has argued that the presence of an individual word type in a given instance, by itself, does not provide enough information to predict the label for *any* ideal task that requires an understanding of natural language [Gardner et al., 2021], in this work we consider this argument only as it relates to two tasks where such a position is relatively uncontroversial: natural language inference, and duplicate-question detection.

Consider the task of natural language inference (NLI), where the input consists of two sentences (premise and hypothesis), and the correct label is a human annotation indicating whether the premise entails the hypothesis, contradicts it, or neither. Continuing our example from section 2.1, if $f_{j,i} = 1$, then the word "nobody" appears somewhere in example x_i (premise, hypothesis, or both). Given these definitions of the task and the features, $f_{j,i} = 1$ by itself is uninformative for predicting y_i (intuitively, we don't learn any information about whether or not the premise entails the hypothesis by knowing that the word "nobody" appears somewhere in the input). However, it has been shown that in the SNLI dataset [Bowman et al., 2015]

$f_j = 1$ almost perfectly predicts the label, in both the training and test sets (for example, in the training set, 2368 instances with $f_j = 1$ have a label of "contradiction" and only 13 don't). Thus, this is an example of a "spurious correlation" (or, bias in the data).

2.4.2 Applying the Bias Test to Models Finetuned on Biased Training Data

We now apply the permutation test described in section 2.2 to finetuned models. For each of SNLI [Bowman et al., 2015], QNLI [Wang et al., 2018], and QQP, we finetune three pretrained RoBERTa-large models [Liu et al., 2019b] with different random seeds on their training sets. We use a learning rate of 2×10^{-6} and finetune for 15 epochs using a single GPU with 12GB memory.

Following the argument by Gardner et al. [2021] that unigram features for these kinds of theoretically complex tasks should ideally be uninformative in isolation, we use lexical types as our bias evaluation features. For the purpose of this calculation, each label will contribute the 50 features that have the strongest correlation with it (as calculated by z-score, again following Gardner et al., 2021) in the lowercased training data, excluding stop words, since they tend to receive high z-scores due to appearing in such an overwhelming number of instances. We then select all test instances with one or more of those types present as our evaluation set for our permutation test. For models finetuned on SNLI and QQP, we find p-values of at most 2.3×10^{-17} (see "Trained on uniform" rows of Table 2.2), indicating very strong evidence that—as expected—these models reflect the bias associated with types with high z-scores in the training set. For QNLI, we see mixed results depending on our random seed, with p-values of 0.0057, 0.024, and 0.053 for our three finetuned models. (Worth noting is the fact that, as we will see later in Section 2.3, QNLI has the lowest overall feature-label bias of any of these three datasets.) Still, we see enough of these models demonstrating bias to merit investigating why this occurs.

Having established that there is often similar bias in the finetuning data and models trained on that data, we consider that the finetuning data is not necessarily the source of the bias in the finetuned RoBERTa-large models. For example, the bias could come from the pretraining data as well. Therefore, in the following section, we also include smaller models (LSTMs) trained from scratch on both versions of our datasets for comparison purposes.

2.4.3 Impact when Finetuning on Reweighted Data

We now consider what happens when we finetune models on data with dramatically lessened bias. We finetune RoBERTa-large models using new random seeds and all the same hyperparameters as before, only this time on training data reweighted using the new q distributions. We see similar validation accuracies (a point or so of difference), indicating that this reweighting has a small effect on overall performance, even though the validation sets may contain similar biases to their corresponding training sets and therefore benefit models that leverage those biases.

			p-value(s) for permutation test
Finetuned RoBERTa	SNLI	Trained on uniform	$1.9 \times 10^{-35}, \{1.1, 2.2\} \times 10^{-23}$
		Trained on adjusted q	$\{1.2, 1.7, 3.2\} \times 10^{-14}$
	QNLI	Trained on uniform	$5.7 \times 10^{-3}, \{2.4, 5.3\} \times 10^{-2}$
		Trained on adjusted q	$\{3.7, 7.6, 2.6\} \times 10^{-1}$
	QQP	Trained on uniform	$2.4 \times 10^{-26}, 2.6 \times 10^{-20}, 2.3 \times 10^{-17}$
		Trained on adjusted q	$7.6 \times 10^{-20}, 5.9 \times 10^{-7}, 1.2 \times 10^{-5}$
From-scratch LSTM	SNLI	Trained on uniform	$5.9 \times 10^{-83}, 7.4 \times 10^{-69}, 9.5 \times 10^{-88}$
		Trained on adjusted q	$2.0 \times 10^{-75}, 8.5 \times 10^{-64}, 7.3 \times 10^{-54}$
	QNLI	Trained on uniform	$3.1 \times 10^{-61}, 1.3 \times 10^{-42}, 5.7 \times 10^{-55}$
		Trained on adjusted q	$1.6 \times 10^{-10}, 1.4 \times 10^{-37}, (1.0 - 5.6 \times 10^{-11})$
	QQP	Trained on uniform	Approx. $10^{-638}, 10^{-168}, 10^{-209}$
		Trained on adjusted q	Approx. 10^{-762}, underflowed for other seeds

Table 2.2: Exact p-values for permutation tests conducted on different models, which check the probability that the usual-gold-label subset of the test data would have at least its observed accuracy if the instances guessed correctly by the model were distributed uniformly at random across the usual and unusual gold-label test subsets. The pretrained model used to initialize each finetuned transformer was RoBERTa-large, and for each pairing of a dataset and a uniform or adjusted weighting of its data in finetuning a transformer, we ran three separate random seeds to observe variance. For each dataset-weighting pairing in training LSTMs from scratch, we also ran three separate random seeds.

The results of rerunning our model bias evaluation are listed in the top half of Table 2.2. While we do see an increase in p-values, indicating weaker evidence of bias than for models trained on the uniformly-weighted training data, for both SNLI and QQP, we are still left with very strong evidence of bias (p-values of at most 1.2×10^{-5}). A natural question that we might ask is whether we can attribute this remaining bias to the pretraining data.

To test whether we see the same patterns in the absence of any other training data, we also train bidirectional three-layer LSTMs per dataset from scratch (i.e., no pretraining and no pretraining data). For each

dataset, we trained three separate seeds of biLSTMs using uniform weighting of the training data, and three others using q-reweighted.[7]

As we can see in Table 2.2, while there generally continues to be a rise in p-value with the switch to the reweighted q, the higher p-value is still vanishingly small except in one case. **Almost all the models trained from scratch are biased.** The one exception to this is interesting: on one of the random seeds of LSTM trained on the adjusted q for QNLI, the p-value is actually very close to 1, while the other random seeds of LSTMs trained on that same reweighted data still display *very* strong evidence of the original dataset's bias. However, the p-value for this one exception model is *so* high that it indicates that if anything, the model has learned biases that favor correctness for the prior *minority* labels for each feature. Even in that one exception model's case, *there is still significant evidence against the null hypothesis;* that evidence just goes the other way, towards an improbable distribution of model correctness in favor of the previous minority labels. This indicates that the LSTMs are still able to work around the reweighted training data to pick up on (versions of) biases associated with features in the original data.

Also of particular interest is the fact that most LSTMs trained on QNLI display strong evidence of bias, while the pretrained transformers that were finetuned on either version of QNLI (reweighted or not) were the only models that did not display strong evidence of bias. This further indicates that at least in QNLI's case, bias has entirely separate causes than the finetuning data; for QNLI, it's only the models trained from scratch that display significant evidence of bias. This, along with the tiny p-values for the other LSTMs, indicates that there are still factors even in the reweighted data that contribute to bias.

	Err(Uniform)(↓)	Err(Adjusted q)(↓)
SNLI	0.059	0.122
QNLI	0.134	0.173
QQP	0.215	0.224

Table 2.3: The average absolute difference between the empirical distribution of label y (in the data) for instances with a **bigram** feature j and the overall distribution of label y given the full data (we perform this difference elementwise). The calculations over any row in this table are performed over 200 randomly selected bigrams j from that dataset, which are kept consistent across columns. Lower is better.

[7] To ensure no leaked signal from any other data, we initialized the word embeddings of the LSTMs to continuous bag-of-words embeddings [Mikolov et al., 2013] trained using their respective q-weighted training sets. We use a word embedding dimension of 128, a hidden size as input to the second LSTM layer of 256, and a hidden size as input to the third LSTM layer of 512. That third layer outputs a 128-dimensional vector, to which a linear projection projecting it to the appropriate number of output dimensions is then applied.

At first, this is surprising. Given that the LSTMs trained with the reweighted q distributions over data were exposed to no other data, why do they still almost always exhibit bias? One possibility is issues of quality inherent to some unusual-label data. For example, consider the word "favorite" in SNLI, which has one of the highest z-scores for the "neutral" label. Even though nothing about the task of determining whether one sentence entails another inherently suggests an association between "favorite" and a particular label, since SNLI was constructed based on photographs (without any additional data about their subjects' mental states) as the underlying source of data for written premises, we expect the term "favorite" to occur mostly in hypotheses that are neither entailed nor contradicted by this data. Even though the reweighted q gives more weight to unusual examples, those examples could sometimes be of lower quality due to details of how the data was collected.

Furthermore, even though the total contribution to the loss function during training is approximately the same across labels using the reweighted q, the model still sees a wider variety of instances for types' "usual" labels, which perhaps allows it to generalize better in that regard. In other words, the characteristics of less common (f_j, y) pairings aren't inherently easier for a model to learn than the characteristics of more common pairings, so models' generalization to new examples with the less common (f_j, y) pairing would still be hurt by seeing a smaller variety of examples representing those kinds of instances, even if that smaller variety received greater total weight in the loss function.

While it's tempting to consider changing the data distributions so that diversity is matched across instances with (pre-reweighting) majority and minority labels, keep in mind that whether an instance is "majority"- or "minority"-label is feature-specific, and that changing the diversity of the data itself would require either removing or adding instances. If adding instances, it quickly becomes necessary to do this with machine-generated text for reasons of scale, which introduces the potential confounds of machine-generated text differing in substantive ways from the rest of the human-written data; Ross et al. [2022] performed an experiment similar to this. If removing instances, the feature-specific nature of the "majority" vs. "minority" labels means that even if an instance receives a weight less than 1 overall, indicating that it's perhaps worth leaving it out, it's still possible that that instance, with its weight less than 1, is still "upweighted" as a minority label for another of the features it contains, compared to even lower-weighted instances. The downsampling that adjusting the data diversity (without adding machine-generated text) would require is

therefore substantially complicated.

2.5 Effects of Rebalancing Data on Higher-Order Features

We have found that rebalancing labeled data doesn't typically remove bias in a downstream model. Another possible explanation is that rebalancing also affects higher-order features' effective correlations with labels, and such bias may carry over into models (whether it was originally present or not). We consider bigrams, as they represent only a slight additional level of complication.

To get a sense of how bigrams overall are affected, we randomly sample 200 bigrams for each of the three successfully rebalanced datasets, selecting uniformly at random among the set of bigrams that appear in at least one instance of each label. We then examine the effect of our (unigram-based) rebalancing of data from table 2.1 on associations in the data between bigram features and labels. Table 2.3 shows that in all cases, the average gap between the overall label distribution in the data and the empirical distribution of labels given a bigram *worsens*, despite unigrams' label distributions better reflection of the data's overall label distribution (Table 2.1) that results from the same reweighted q.

This analysis provides a possible explanation for how rebalancing the data with respect to biased unigram features fails to prevent models from learning bias: the rebalancing didn't correct for biased bigram features, which mislead the model, effectively "bringing the unigram features" along with them so that unigram-bias gets learned anyway. This is a troubling sign for approaches to bias reduction that focus on data alone, pointing to the need for methods that focus on other aspects of model learning as well.

2.6 Limitations

One of the limitations of this work is that we restrict ourselves to examining datasets for supervised learning that contain relatively short instances of text. This likely facilitated the reweighting of data that we wished to perform as an intervention to produce the reweighted data that we study, as the short length of each text effectively capped the number of different lexical features that could cooccur in the same instance. The results we present here might not be representative of lexical feature bias in data with much longer units of text. Also, the fact that the datasets that we used are all in English means that our lexical features

were premised on simple whitespace tokenization with punctuation removal; for other languages with a larger variety of reasonable tokenization schemes at varying levels of granularity, the distribution of lexical features, and the resulting conclusions, might look very different.

In addition, apart from the issues we have raised in transferring reduced bias in data to models, we note that an exhaustive list of *all* features that are present in particular data is extremely impractical (and in some cases impossible); any set of features will inevitably leave out some trait of the data, making the reweighting procedure we follow in this work inherently incomprehensive. For those features not included in the problem setup, the measured quality of a returned q distribution will not reflect any changes relevant to those features, although the balance of those features has likely also changed. Even among the features included in the problem input, shifting q's probability mass to improve the balance for one set of features' labels may simultaneously hurt the balance for another.

2.7 Relating This Analysis to Other Findings

Results using our permutation testing framework indicate the difficulty of removing or mitigating bias from data in a way that corresponds to the mechanisms by which models absorb that bias in practice. This is reminiscent of results from, for example, Gonen and Goldberg [2019] or Elazar and Goldberg [2018], who note that certain ways of seemingly covering up bias still leave traces of that bias in models, and is in line with arguments made by, for example, Eisenstein [2022] and Schwartz and Stanovsky [2022]. Further development and testing of hypotheses about how models acquire bias will be important to ensuring that they truly perform the tasks that we intend, and not versions that rely on biased shortcuts in the data.

Here, we explored how lexical bias in labeled data affects bias in models trained on that data. To structure our experiment, we used a procedure based on the permutation test for analyzing biased associations between given features and model predictions, in test data that might itself contain biases, as well as an optimization-based method for mitigating a given list of biases in data. Our empirical finding is that, in cases where a dataset can be rebalanced to remove most lexical bias, the resulting models remain biased. This may be related to our observation that the correlations of higher-order (bigram) features with labels actually get *worse* after rebalancing. We conclude that reducing bias in NLP models may not be achievable by altering existing training data distributions.

Chapter 3

Assessing the Faithfulness of Attention Distributions

Producing instance-level explanations of which information from an instance of data was important to a model producing its corresponding output is a key challenge in model interpretability and analysis. This is a harder problem than querying whether a model generally leverages a particular kind of information, as we did in the previous chapter. If we could determine precisely why a model made its particular decisions at the instance level, then given enough instances, we would have enough information to draw broad conclusions about how the model generally works, but not vice versa. For example, it's easier to conclude that a model is biased than it is to pinpoint precisely those instances for which the model improperly made a decision using biased features. Therefore, there has been a considerable amount of research into many different kinds of strategies for producing these instance-level model explanations [Ribeiro et al., 2016; Lundberg and Lee, 2017; Li et al., 2016; Sundararajan et al., 2017, *inter alia*].

One branch of this literature [Andreas et al., 2016a,b; Lei et al., 2016] focuses on models designed to be "inherently interpretable." This idea refers to models that produce decisions or computed quantities as byproducts during inference that offer a conceptual interpretation of some aspect of how it produced its output. This is very attractive for multiple reasons. For one thing, this paradigm requires no additional computation to explain a decision, provided that inference has been run. For another, because these interpretable components are baked into the model itself, they could theoretically avoid stated limitations of post-hoc

interpretability methods (like an approximately linear local decision boundary for Ribeiro et al. [2016], or an assumption that feature importance is additive as in Lundberg and Lee [2017]). However, due to the proposed explanation method being an inherent part of the model, it has not always been obvious that additional verification of its explanation faithfulness was needed, and such verification could be complicated. As a case study, this chapter will focus on the interpretability of the attention mechanism [Bahdanau et al., 2015], borrowing materials originally published in Serrano and Smith [2019].

Part of recent years' development of NLP models has been the incorporation of attention mechanisms into models for a variety of tasks. For many different problems—to name a few, machine translation [Luong et al., 2015], syntactic parsing [Vinyals et al., 2015], reading comprehension [Hermann et al., 2015], and language modeling [Liu and Lapata, 2018]—incorporating attention mechanisms into models has proven beneficial for performance. While there are many variants of attention [Vaswani et al., 2017], each formulation consists of the same high-level goal: calculating nonnegative weights for each input component (e.g., word) that together sum to 1, multiplying those weights by their corresponding representations, and summing the resulting vectors into a single fixed-length representation.

Since attention calculates a distribution over inputs, prior work has used attention as a tool for interpretation of model decisions [Wang et al., 2016; Lee et al., 2017; Lin et al., 2017; Ghaeini et al., 2018]. The existence of so much work on visualizing attention weights is a testament to attention's popularity in this regard; to name just a few examples of these weights being examined to understand a model, work conducted shortly before this project focused on goals from explaining and debugging the current system's decision [Lee et al., 2017; Ding et al., 2017] to distilling important traits of a dataset [Yang et al., 2017; Habernal et al., 2018].

Despite this, when this project was conducted, existing work on interpretability had only begun to assess what computed attention weights actually communicate. In an independent and contemporaneous study, Jain and Wallace [2019] explored whether attention mechanisms can identify the relative importance of inputs to the full model, finding them to be highly inconsistent predictors. In this work, we apply a different analysis based on *intermediate* representation erasure to assess **whether attention weights can instead be relied upon to explain the relative importance of the inputs to the attention layer itself.** We find similar cause for concern: attention weights are only noisy predictors of even intermediate components' importance, and

should not be treated as justification for a decision.

3.1 Testing for Informative Interpretability

We focus on five- and ten-class text classification models incorporating attention, as explaining the reasons for text classification has been a particular area of interest for work in interpretability [Yang et al., 2016; Ribeiro et al., 2016; Lei et al., 2016; Feng et al., 2018].

In order for a model to be interpretable, it must not only suggest explanations that make sense to people, but also ensure that those explanations accurately represent the true reasons for the model's decision. Note that this type of analysis does not rely on the true labels of the data; if a model produces an incorrect output, but a faithful explanation for which factors were important in that calculation, we still consider it interpretable.

We take the implied explanation provided by visualizing attention weights to be a ranking of importance of the attention layer's input representations, which we denote $\mathcal{I}$: if the attention allocated to item $i \in \mathcal{I}$ is higher than that allocated to item $j \in \mathcal{I}$, then i is presumed "more important" than j to the model's output. In this work, we focus on whether the attention weights' suggested importance ranking of $\mathcal{I}$ faithfully describes why the model produced its output, echoing existing work on explanation brittleness for other model components [Ghorbani et al., 2017].

3.1.1 Intermediate Representation Erasure

We are interested in the impact of some contextualized inputs to an attention layer, $\mathcal{I}' \subset \mathcal{I}$, on the model's output. To examine the importance of $\mathcal{I}'$, we run the classification layer of the model twice (Figure 3.1): once without any modification, and once after renormalizing the attention distribution with $\mathcal{I}'$'s attention weights zeroed out, similar to other erasure-based work [Li et al., 2016; Feng et al., 2018]. We then observe the resulting effects on the model's output. We erase at the attention layer to isolate the effects of the attention layer from the encoder preceding it. Our reasoning behind renormalizing is to keep the output document representation from artificially shrinking closer to $\mathbf{0}$ in a way never encountered during training, which could make subsequent measurements unrepresentative of the model's behavior in spaces to which it *does* map inputs.

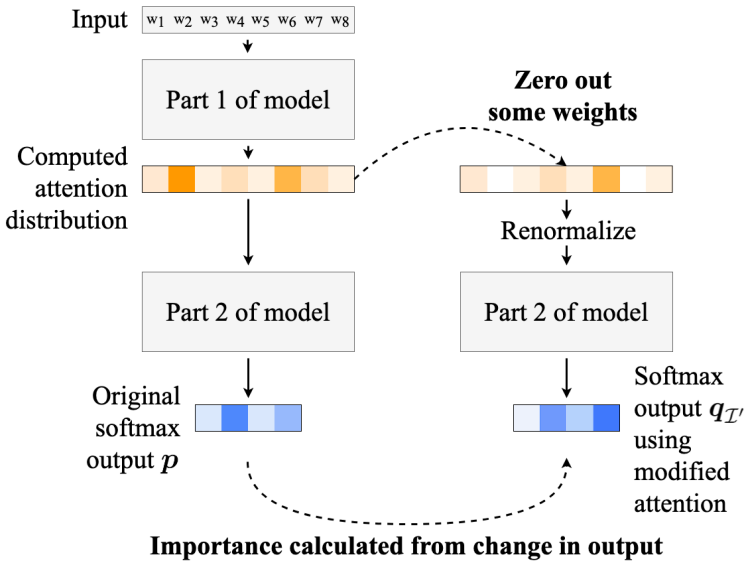


Figure 3.1: Our method for calculating the importance of representations corresponding to zeroed-out attention weights, in a hypothetical setting with four output classes .

One point worth noting is the facet of interpretability that our tests are designed to capture. By examining only how well attention represents the importance of intermediate quantities, which may themselves already have changed uninterpretably from the model's inputs, we are testing for a relatively low level of interpretability. Other contemporaneous work to this project looking at attention examines whether attention suffices as a holistic explanation for a model's decision [Jain and Wallace, 2019], which is a higher bar. We instead focus on the lowest standard of interpretability that attention might be expected to meet, ignoring prior model layers.

While starting with this most conservative hypothesis about attention's interpretability– that NLP models indicate via their attention mechanisms the relative importance of *immediately preceding earlier representations*– makes sense as a starting point for verifying the explanatory power of attention, the nature of this hypothesis also introduces a difficulty in our experiment design. Experiments designed to verify an explanation method often deploy that explanation method in a setting with a known ground truth as to which input information is "important" (that is, which input information enables producing the correct output). In such a setup, the explanation method's success rate at recovering the known important input information suffices to argue for or against that method's validity for identifying important input. However, in our case, we have no way of knowing how a model has combined or shifted separate input tokens' information between the

Dataset	Av. # Words	(s.d.)	Av. # Sents.	(s.d.)	# Train. + Dev.	# Test	# Classes
Yahoo Answers	104	(114)	6.2	(5.9)	1,400,000	50,000	10
IMDB	395	(259)	16.2	(10.7)	122,121	13,548	10
Amazon	73	(48)	4.3	(2.6)	3,000,000	650,000	5
Yelp	125	(109)	7.0	(5.6)	650,000	50,000	5

Table 3.1: Dataset statistics.

different *intermediate* representations being weighted via attention. In most model configurations, we therefore lose the ability to compare to any "ground truth," as that ground truth could potentially be localized in any of the intermediate representations. Our experiments will therefore require us to proceed without such a ground truth; we lay out strategies for alternative experiment design in sections 3.3 and 3.4.

We denote the output distributions (over labels) as $\boldsymbol{p}$ (the original) and $\boldsymbol{q}_{\mathcal{I}'}$ (where we erase attention for $\mathcal{I}'$). The question now becomes how to operationalize "importance" given $\boldsymbol{p}$ and $\boldsymbol{q}_{\mathcal{I}'}$. There are many quantities that could arguably capture information about importance. We focus on two: the Jensen-Shannon (JS) divergence between output distributions $\boldsymbol{p}$ and $\boldsymbol{q}_{\mathcal{I}'}$, and whether the argmaxes of $\boldsymbol{p}$ and $\boldsymbol{q}_{\mathcal{I}'}$ differ, indicating a decision flip.

3.2 Data and Models

We investigate four model architectures on a topic classification dataset (Yahoo Answers; Zhang et al., 2015) and on three review ratings datasets: IMDB [Diao et al., 2014],[1] Yelp 2017,[2] and Amazon [Zhang et al., 2015]. Statistics for each dataset are listed in Table 3.1.

Our model architectures are inspired by the hierarchical attention network [HAN; Yang et al., 2016], a text classification model with two layers of attention, first to the word tokens in each sentence and then to the resulting sentence representations. The layer that classifies the document representation is linear with a softmax at the end.

We conduct our tests on the softmax formulation of attention,[3] which is used by most models, including the HAN. Specifically, we use the additive formulation originally defined in Bahdanau et al. [2015].

[1] downloaded from `https://github.com/nihalb/JMARS`

[2] `https://web.archive.org/web/20171208174731/https://www.yelp.com/dataset/challenge`

[3] Alternatives such as sparse attention [Martins and Astudillo, 2016] and unnormalized attention [Ji and Smith, 2017] have been proposed.

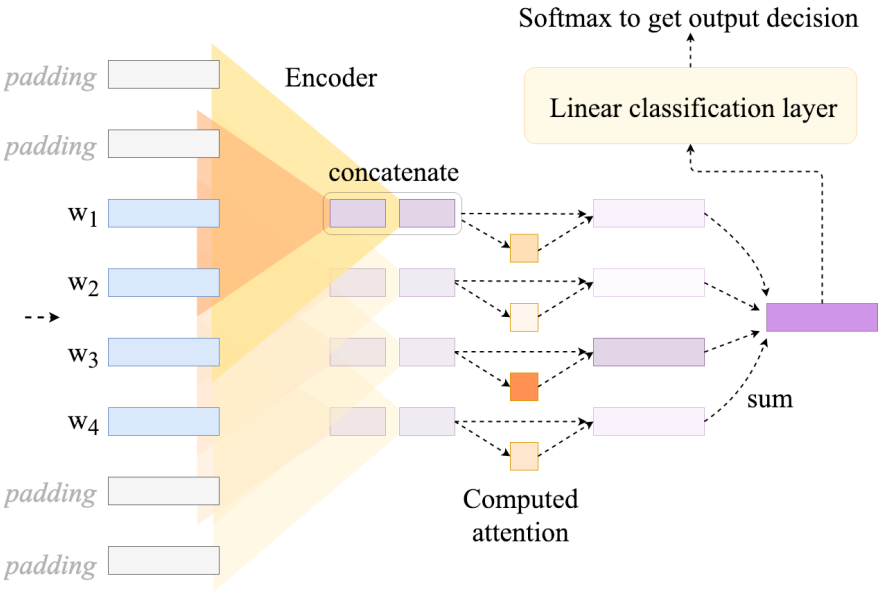


Figure 3.2: Flat attention network (FLAN) demonstrating a convolutional encoder. Each contextualized word representation is the concatenation of two sizes of convolutions: one applied over the input representation and its two neighbors to either side, and the other applied over the input representation and its single neighbor to either side.

Given attention layer ℓ's learned parameters, element i of a sequence, and its encoded representation $\mathbf{h}_i$, the attention weight α_i is computed using ℓ's learned context vector $\mathbf{c}_\ell$ as follows:

$$\mathbf{u}_i = \tanh(\mathbf{W}_\ell \mathbf{h}_i + \mathbf{b}_\ell)$$
$$\alpha_i = \frac{\exp \mathbf{u}_i^\top \mathbf{c}_\ell}{\sum_i \exp \mathbf{u}_i^\top \mathbf{c}_\ell}$$

We evaluate on the original HAN architecture, but we also vary that architecture in two key ways:

1. Number of attention layers: besides exploring models with a final layer of attention over sentence representations (which we denote with a "HAN" prefix), we also train "flat" attention networks with only one layer of attention over all contextualized word tokens (which we denote with a "FLAN" prefix). In either case, though, we only run tests on models' final layer of attention.

2. Reach of encoder contextualization: The original HAN uses recurrent encoders to contextualize input tokens prior to an attention layer (specifically, bidirectional GRUs running over the full sequence). Aside from biRNNs, we also experiment with models that instead contextualize word vectors by

convolutions on only a token's close neighbors, inspired by Kim [2014]. See Figure 3.2 for a diagram of the FLAN architecture using a convolutional encoder. We denote this variant of an architecture with a "conv" suffix. Finally, we also test models that are trained with no contextualizing encoder at all; we denote these with a "noenc" suffix.

3.3 Single Attention Weights' Importance

As a starting point for our tests, we investigate the relative importance of attention weights when only one weight is removed. Let $i^* \in \mathcal{I}$ be the component with the highest attention and let α_{i^*} be its attention. We compare i^*'s importance to some other attended item's importance in two ways.

3.3.1 JS Divergence of Model Output Distributions

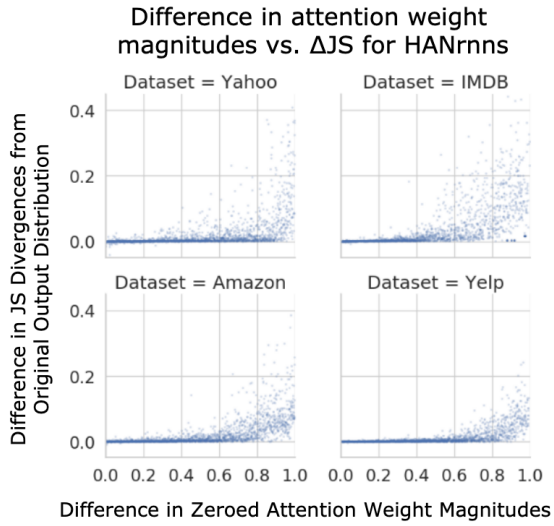
We wish to compare how i^*'s impact on the model's output distribution compares to the impact corresponding to a random attended item r drawn uniformly from $\mathcal{I}$. Our first approach to this will be to calculate two JS divergences—one being the JS divergence of the model's original *output* distribution from its output distribution after removing only i^*, and the other after removing only r—and compare them to each other. We subtract the output JS divergence after removing r from the output JS divergence after removing i^*:

$$\Delta\text{JS} = \text{JS}(\boldsymbol{p}, \boldsymbol{q}_{\{i^*\}}) - \text{JS}(\boldsymbol{p}, \boldsymbol{q}_{\{r\}}) \tag{3.1}$$

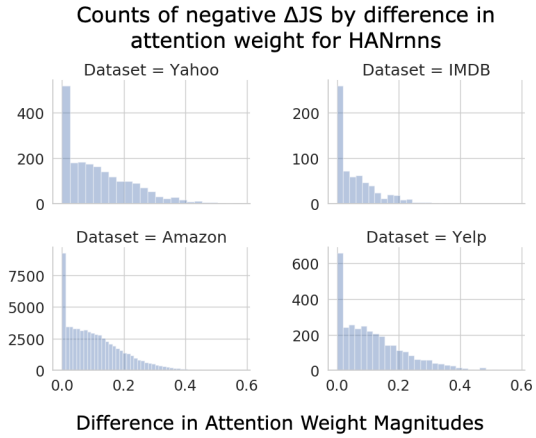
We plot this quantity against the difference $\Delta\alpha = \alpha_{i^*} - \alpha_r$ in Figure 3.3a. We show results on the HANrnn, as the trends for the other models are very similar.

Intuitively, if i^* is truly the most important, then we would expect Eq. 3.1 to be positive, and that is what we find the vast majority of the time. In addition, examining Figure 3.3a, we see that nearly all negative ΔJS values are close to 0. By binning occurrences of negative ΔJS values by the difference between α_{i^*} and α_r in Figure 3.3b, we also see that in the cases where i^* had a smaller effect, the gap between i^*'s attention and r's tends to be small. This is encouraging, indicating that in these cases, i^* and r are nearly "tied" in attention.

However, the picture of attention's interpretability grows somewhat more murky when we begin to



(a) Difference in attention weight magnitudes versus ΔJS for HANrnns, comparable to results for the other architectures.



(b) These are the counts of test instances for the HANrnn models for which i^*'s JS divergence was smaller, binned by $\Delta\alpha$. These counts comprise a small fraction of the test set sizes listed in Table 3.1.

Figure 3.3: Plots reporting on ΔJS

consider the magnitudes of positive ΔJS values in Figure 3.3a. We notice across datasets that even for quite large differences in attention weights like 0.4, many of the positive ΔJS are still quite close to zero. Although we do finally see an upward swing in ΔJS values once $\Delta\alpha$ gets even larger, indicating only one very high attention weight in the distribution, this still leaves many open questions about exactly how much difference in impact i^* and r can typically be expected to have.

3.3.2 Decision Flips Caused by Zeroing Attention

Since attention weights are often interpreted as an explanation for a model's argmax decision, our second test looks at another, more immediately visible change in model outputs: decision flips. For clarity, we limit our discussion to results for the HANrnns, which reflect the same patterns observed for the other architectures.

Table 3.2 shows, for each dataset, a contingency table for the two binary random variables (i) does zeroing α_{i^*} (and renormalizing) result in a decision flip? and (ii) does doing the same for a different randomly chosen weight α_r result in a decision flip? To assess the comparative importance of i^* and r, we consider when exactly one erasure changes the decision (off-diagonal cells). For attention to be interpretable, the

blue, upper-right values (i^*, not r, flips a decision) should be much larger than the orange, lower-left values (r, not i^*, flips a decision), which should be close to zero.[4]

Although for some datasets in Table 3.2, the "orange" values are non-negligible, we mostly see that their fraction of total off-diagonal values mirrors the fraction of negative occurrences of Eq. 1 in Figure 3.3b. However, it's somewhat startling that in the vast majority of cases, erasing i^* does *not* change the decision ("no" row of each table). This is likely explained in part by the signal pertinent to the classification being distributed across a document (e.g., a "Sports" question in the Yahoo Answers dataset could signal "sports" in a few sentences, any one of which suffices to correctly categorize it). However, given that these results are for the HAN models, which typically compute attention over ten or fewer sentences, this is surprising.

Remove random: Decision flip?

Remove i^*: Decision flip?

Yahoo

	Yes	No
Yes	0.5	8.7
No	1.3	89.6

IMDB

	Yes	No
Yes	2.2	12.2
No	1.4	84.2

Amazon

	Yes	No
Yes	2.7	7.6
No	2.7	87.1

Yelp

	Yes	No
Yes	1.5	8.9
No	1.9	87.7

Table 3.2: Percent of test instances in each decision-flip indicator variable category for each HANrnn.

Altogether, examining importance from a single-weight angle paints a tentatively positive picture of attention's interpretability, but also raises several questions about the many cases where the difference in impacts between i^* and r is almost identical (i.e., ΔJS values close to 0 or the many cases where neither i^* nor r cause a decision flip). To answer these questions, we require tests with a broader scope.

3.4 Importance of Sets of Attention Weights

Often, we care about determining the *collective* importance of a set of components $\mathcal{I}'$. To address that aspect of attention's interpretability and close gaps left by single-weight tests, we introduce tests to determine how

[4] We see this pattern especially strongly for FLANs, which is unsurprising since $\mathcal{I}$ is all *words* in the input text, so most attention weights are very small.

multiple attention weights perform together as importance predictors.

3.4.1 Multi-Weight Tests

For a hypothesized ranking of importance, such as that implied by attention weights, we would expect the items at the top of that ranking to function as a concise explanation for the model's decision. The less concise these explanations get, and the farther down the ranking that the items truly driving the model's decision fall, the less likely it becomes for that ranking to truly describe importance. In other words, we expect that the top items in a truly useful ranking of importance would comprise a minimal necessary set of information for making the model's decision.

The idea of a minimal set of inputs necessary to uphold a decision is not new; Li et al. [2016] use reinforcement learning to attempt to construct such a minimal set of words, Lei et al. [2016] train an encoder to constrain the input prior to classification, and much of the work that has been done on extractive summarization takes this concept as a starting point [Lin and Bilmes, 2011]. However, such work has focused on approximating minimal sets, instead of evaluating the ability of other importance-determining "shortcuts" (such as attention weight orderings) to identify them. Nguyen [2018] leveraged the idea of minimal sets in a much more similar way to our work, comparing different input importance orderings.

Concretely, to assess the validity of an importance ranking method (e.g., attention), we begin erasing representations from the top of the ranking downward until the model's decision changes. Ideally, we would then enumerate all possible subsets of that instance's components, observe whether the model's decision changed in response to removing each subset, and then report whether the size of the minimal decision-flipping subset was equal to the number of items that had needed to be removed to achieve a decision flip by following the ranking. However, the exponential number of subsets for any given instance's sequence of components (word or sentence representations, in our case) makes such a strategy computationally prohibitive, and so we adopt a different approach.

Instead, in addition to our hypothesized importance ranking (attention weights), we consider alternative rankings of importance; if, using those, we repeatedly discover cases where removing a smaller subset of items would have sufficed to change the decision, this signals that our candidate ranking is a poor indicator of importance.

3.4.2 Alternative Importance Rankings

Exhaustively searching the space of component subsets would be far too time-consuming in practice, so we introduce three other ranking schemes.

The first is to randomly rank importance. We expect that this ranking will perform quite poorly, but it provides a point of comparison by which to validate that ranking by descending attention weights is at least somewhat informative.

The second ranking scheme, inspired by Li et al. [2015] and Feng et al. [2018], is to order the attention weights by the gradient of the decision function with respect to each calculated attention weight, in descending order. Since each of the datasets on which we evaluate has either five or ten output classes, we take the decision function given a real-valued model output vector to be

$$d(\boldsymbol{x}) = \frac{\exp\left(\max_i\left(\boldsymbol{x}_i\right)\right)}{\sum_i \exp \boldsymbol{x}_i}.$$

Unlike the last two proposed rankings, our third ranking scheme uses attention weights, but supplements them with information about the gradient. For this ranking, we multiply each of our calculated gradients from our previous proposed ranking scheme by their corresponding attention weight magnitude. Under this ordering, attended items that have both a high attention weight and a high calculated gradient with respect to their attention weight will be ranked most important.

We introduce these last two rankings as an attempt to discover smaller sets not produced by the attention weight ranking. Note, however, that we still do not take either as a gold-standard indicator of importance to the model, as with the gradient in Ross et al. [2017] and Melis and Jaakkola [2018], but merely as an alternative ordering method. The "gold standard" in our case would be the minimal set of attention weights to zero out for the decision to change, which none of our ordering methods will necessarily find for a particular instance.

3.4.3 Instances Excluded from Analysis

In cases where removing all but one input to the attention layer still does not produce a decision flip, we finish the process of removing components by removing the final representation and replacing the output of

the attention layer with an arbitrary vector; we use the zero vector for our tests. Even so, since every real-valued vector output by the attention layer is mapped to an output distribution, removing this final item will still not change the classification decision for instances that the model happened to originally map to that same class. We exclude such instances for which the decision never changed from all subsequent analyses.

We also set aside any test instances with a sequence length of 1 for their final attention layer, as there is only one possible ordering for such cases.

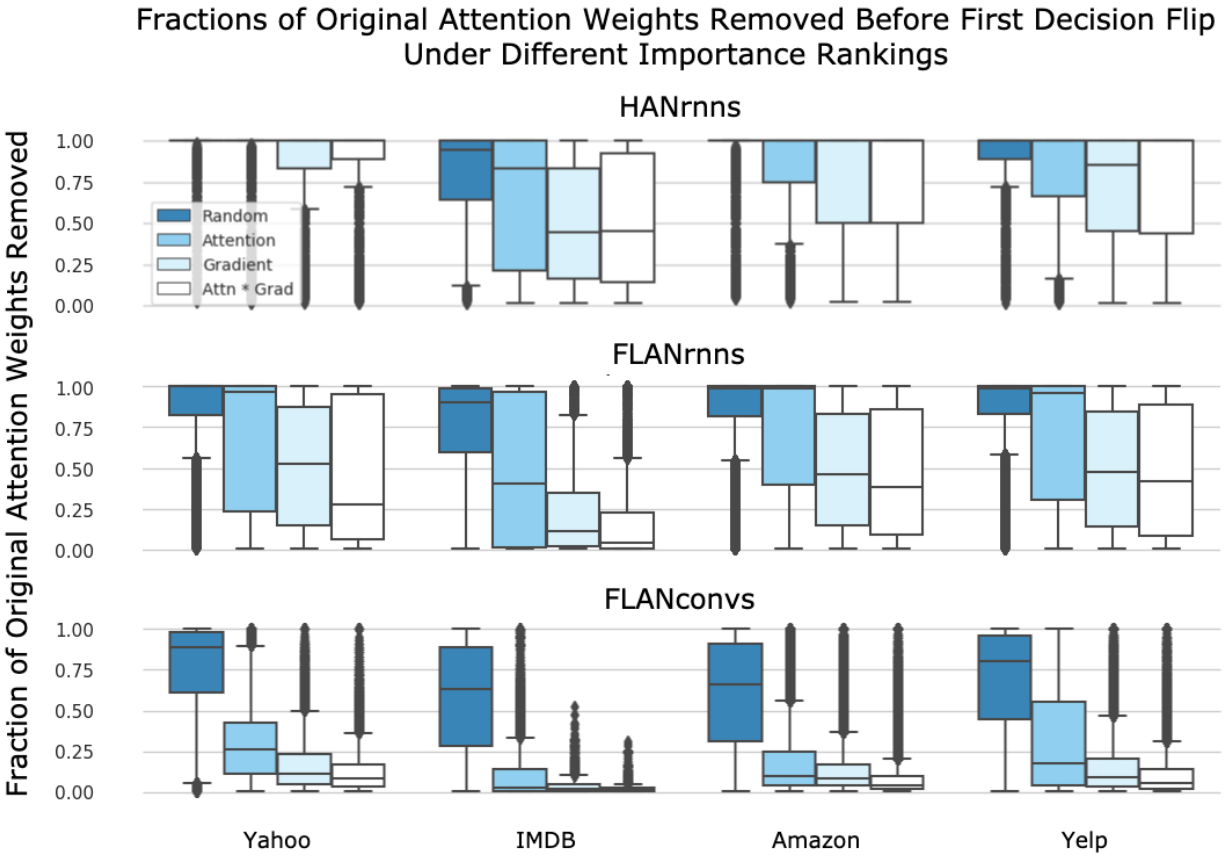


Figure 3.4: The distribution of fractions of items removed before first decision flips on three model architectures under different ranking schemes. Boxplot whiskers represent the highest/lowest data point within 1.5 IQR of the higher/lower quartile, and dataset names at the bottom apply to their whole column. In several of the plots, the median or lower quartile aren't visible; in these cases, the median/lower quartile is either 1 or very close to 1.

3.4.4 Attention Does Not Optimally Describe Model Decisions

Examining our results in Figure 3.4, we immediately see that ranking importance by descending attention weights is not optimal for our models with encoders. While removing intermediate representations in decreasing order by attention weights often leads to a decision flip faster than a random ranking, it also clearly falls short of matching (or even approaching) the decision-flipping efficiency of either the gradient ordering or gradient-attention-product ordering in many cases.

In addition, although the product-based ranking often (but not always) requires slightly fewer removed items than the gradient ranking, we see that the purely gradient-based ranking ignoring attention magnitudes comes quite close to it, far outperforming purely attention-based orderings. For ten of our 16 models with encoders, removing by gradient found a smaller decision-flipping set of items than attention for over 50% of instances in that model's test set, with that percentage often being much higher. In fact, for *every* model with an encoder that we tested, there were at least 1.6 times as many test instances where the purely gradient-based ranking managed a decision flip faster than the attention-based ranking than vice versa.

A very reasonable follow-up question would be whether examining the fraction of *items* zeroed, instead of the fraction of the original attention distribution's *probability mass* zeroed, exaggerates the differences between the purely attention-based ranking of importance and the gradient-based rankings of importance. We might suppose that perhaps, in the cases where a gradient-based ordering starts the same as the purely attention-based one but eventually diverges, and the gradient-based choice results in a faster decision flip, we were already down to very low-magnitude attentions weights anyway by the time the two orderings differed. In such a hypothetical case, examining only the fraction of items zeroed could imply a far worse explanation for the purely attention-based ordering, when a human looking at attention weights with tiny magnitudes (and therefore tiny differences between them) might instead interpret the attention weights to be ambivalent as to the relative importance of their respective representations.

However, examining the fraction of original attention distribution probability mass zeroed per decision flip in Figure 3.5, we see that in fact, the differences between the different nonrandom importance-ranking methods are *not* negligible. This implies that there are a substantial number of cases where one of the gradient-based rankings bypasses a considerable fraction of the attention distribution's probability mass in favor of at least one much lower-probability item, and yet still achieves a faster decision flip, in cases where

a human would likely *not* read ambivalence into the attention distribution.

We do not claim that ranking importance by either descending gradients or descending gradient-attention products is optimal, but in many cases they discover much smaller decision-flipping sets of items than attention weights. Therefore, ranking representations in descending order by attention weight clearly fails to uncover a minimal set of decision-flipping information much of the time, which is a warning sign that we should be skeptical of trusting groups of attention weight magnitudes as importance indicators.

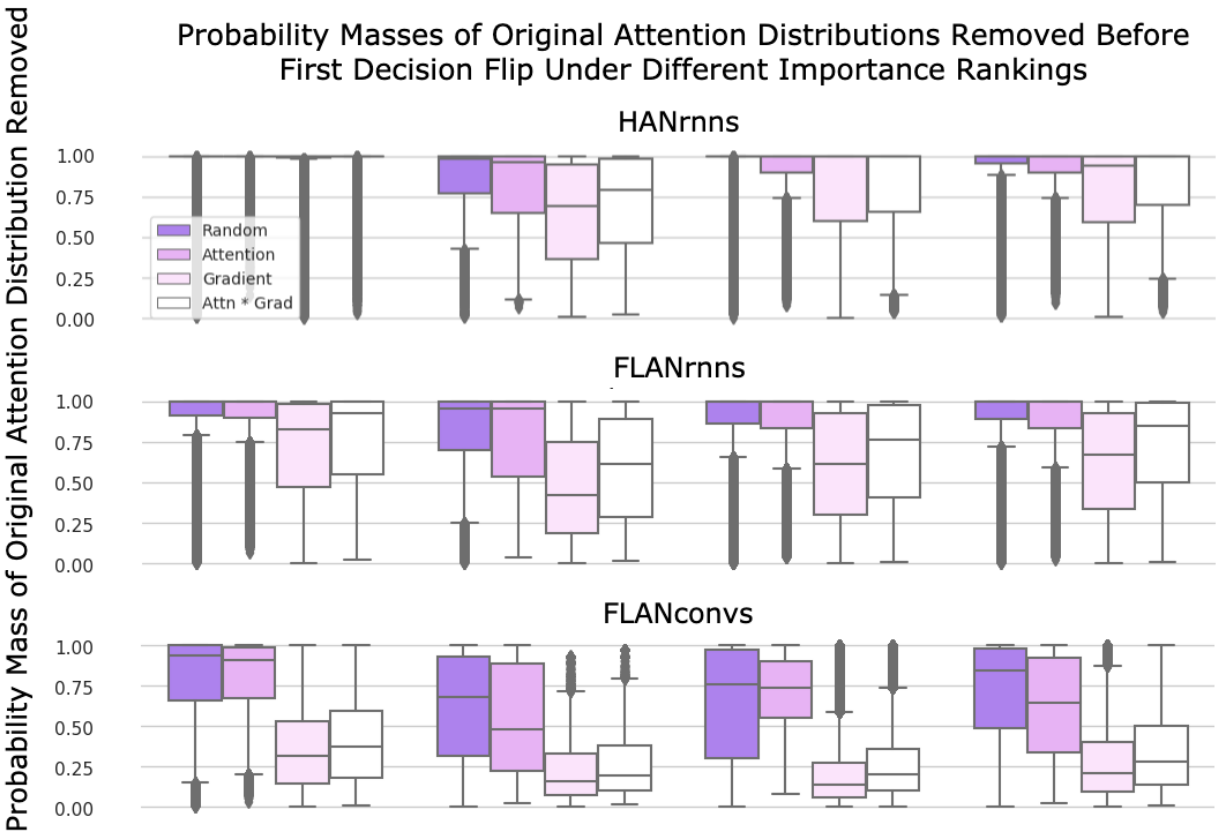


Figure 3.5: The distribution of fractions of the original attention distributions' *probability mass* zeroed before first decision flips on three model architectures under different ranking schemes. Boxplot whiskers represent the highest/lowest data point within 1.5 IQR of the higher/lower quartile, and dataset names at the bottom apply to their whole column. In several of the plots, the median or lower quartile aren't visible; in these cases, the median/lower quartile is either 1 or very close to 1.

3.4.5 Decision Flips Often Occur Late

For all ordering schemes we tried, we were struck by the large fraction of items that had to be removed to achieve a decision flip in many models. This is slightly less surprising for the HANs, as they compute

attention over shorter sequences of sentences (see Table 3.1). For the FLAN models, though, this result is highly unexpected. The sequences across which FLANs compute attention are usually hundreds of tokens in length, meaning most attention weights will likely be minuscule.

The distributions of tokens removed by our different orderings that we see for the FLANrnns in Figure 3.4 are therefore remarkably high, especially given that all of our classification tasks have at least five output classes. We also note that due to the exponential nature of the softmax, softmax attention distributions typically contain only a few high-weighted items before the calculated weights become quite small, which can be misleading. In many cases, flipping the model's original decision requires digging deep into the small attention weights, with the high-weighted components not actually being the reason for the decision.

For several of our models, especially the FLANs (which typically compute attention over hundreds of tokens), this fact is concerning from an explainability perspective. Lipton [2016] describes a model as "transparent" if "a person can contemplate the entire model at once." Applying this insight to the explanations suggested by attention, if an explanation rests on simultaneously considering hundreds of attended tokens necessary for a decision– even if that set were minimal—that would still raise serious transparency concerns.

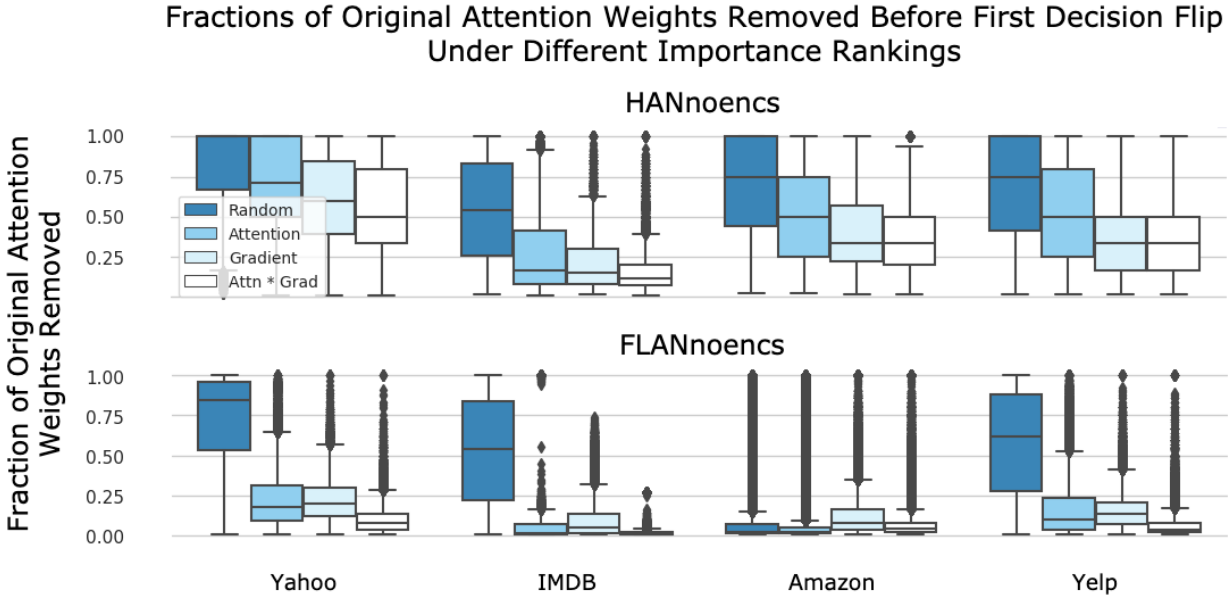


Figure 3.6: The distribution of fractions of items removed before decision flips on the encoderless model architectures under different ranking schemes. The Amazon FLANnoenc results have a very long tail; using the legend's order of rankings, the percentage of test instances with a removed fraction above 0.50 for that model is 12.4%, 2.8%, 0.9%, and 0.5%, respectively.

3.4.6 Effects of Contextualization Scope on Attention's Interpretability

One last question we consider is whether the large number of items that are removed before decision flips can be explained in part by the scope of each model's contextualization. In machine translation, prior work has observed that recurrent encoders over a full sequence can "shift" tokens' signal in ways that cause subsequent attention layers to compute unintuitive off-by-one alignments [Koehn and Knowles, 2017]. We hypothesize that in our text classification setting, the bidirectional recurrent structure of the HANrnn and FLANrnn encoders might instead be redistributing operative signal from a few informative input tokens across many others' contextualized representations.

Comparing the decision flip results for the FLANconvs in Figure 3.4 to those for the FLANrnns supports this theory. We notice decision flips happening much faster than for either of the RNN-based model architectures, indicating that the biRNN effectively does learn to widely redistribute the classification signal. In contrast, the convolutional encoders only allow contextualization with respect to an input token's two neighbors to either side. We see similar results when comparing the two HAN architectures, albeit much more weakly; this is likely due to the smaller number of tokens being contextualized by the HANs (sentence representations instead of words), so that contextualization with respect to a token's close neighbors encompasses a much larger fraction of the full sequence.

We see this difference even more strongly when we compare to the encoderless model architectures, as shown in Figure 3.6. Compared to both other model architectures, we see the fraction of necessary items to erase for flipping the decision plummet. We also see random orderings mostly do better than before, indicating more brittle decision boundaries, *especially* on the Amazon dataset.[5] In this situation, we see attention magnitudes generally indicate importance on par with (or better than) gradients, but that the product-based ordering is still often a more efficient explanation.

While these differences themselves are not an argument against attention's interpretability, they highlight the distinction between attention's weighting of intermediate, contextualized representations and the model's use of the original input tokens themselves. Our RNN-based models' ability to maintain their original

[5]This is likely due to the fact that with no contextualization, the final attended representations are just a linear combination of the input embeddings, so the embeddings themselves are responsible for learning to directly encode a decision. Since Amazon has the largest ratio of documents (which probably vary in their label) to unique word embeddings by a factor of more than two times any other dataset's, and the final attended representations in the FLANnoencs are unaggregated word embeddings, it stands to reason that the lack of encoders would be a much bigger obstacle in its case.

decision well past the point at which models using only local or no context have lost the signal driving their original decisions confirms that attention weights for a contextualized representation do not necessarily map neatly back to the original tokens. This might at least partly explain the striking near-indifference of the model's decision to the contributions of particular contextualized representations in both our RNN-based models and in Jain and Wallace [2019], who also use recurrent encoders.

However, the results from almost all models continue to support that ranking importance by attention is still not optimal; our non-random alternative rankings still uncover many cases where fewer items could be removed to achieve a decision flip than the attention weights imply.

3.5 Limitations

There are important limitations to the work we describe here, perhaps the most important of which is our focus on text classification. By choosing to focus on this task, we use the fact that decision flips are often not trivially achieved to ground our judgments of importance in model decision changes. However, for a task with a much larger output space (such as language modeling or machine translation) where almost anything might flip the decision, decision flips are likely too coarse a signal to identify meaningful differences. Determining an analogous informative threshold in changes to model outputs would be key to expanding this sort of analysis to other groups of models.

A related limitation is our reliance in many of these tests on a fairly strict definition of importance tied to the output's argmax; an alternative definition of importance might assert that the highest attention weights should identify the most influential representations in pushing towards *any* output class, not just the argmax. Two of the core challenges that would need to be solved to test for how well attention meets this relaxed criterion would be meaningfully evaluating a single attended item's "importance" to multiple output classes for comparison to other attended items and, once again, determining what would truly indicate being "most influential" in the absence of decision flips as a guide to the output space.

Also, while we explore several model architectures in this work, there exist other attention functions such as multi-headed and scaled dot-product [Vaswani et al., 2017], as well as cases where a single attention layer is responsible for producing more than one attended representation, such as in self-attention [Cheng et al., 2016]. These variants could have different interpretability properties. Likewise, we only evaluate on

final layers of attention here; in large models, lower-level layers of attention might learn to work differently.

3.6 Related and Subsequent Work

We have adopted an erasure-based approach to probing the interpretability of computed attention weights, but there are many other possible approaches. For example, other work has focused on which training instances [Koh and Liang, 2017] or which human-interpretable features were most relevant for a particular decision [Ribeiro et al., 2016; Arras et al., 2016]. Others have explored alternative ways of comparing the behavior of proposed explanation methods [Adebayo et al., 2018]. Yet another line of work focuses on aligning models with human feedback for what is interpretable [Fyshe et al., 2015; Subramanian et al., 2017], which could refine our idea of what defines a high-quality explanation derived from attention.

With that said, in this work, we see a correlation between attention magnitudes and importance of input information, but by no means a perfect one. Since this project was first published, there has been a considerable amount of subsequent work that has considered similar questions about the interpretation of attention mechanisms [Wiegreffe et al., 2021; Zhong et al., 2019; Vashishth et al., 2019; Moradi et al., 2019; Pruthi et al., 2020; Brunner et al., 2020; Kobayashi et al., 2020; Tutek and Snajder, 2020; Bibal et al., 2022; Kobayashi et al., 2021], which in aggregate finds similarly mixed results. This work has, among other things, extended analysis to attention mechanisms in popular transformer-based models [Vaswani et al., 2017], continuing to find mixed results on what attention communicates. One obstacle to this line of work is that the transformer architecture includes many, many attention mechanisms, complicating the question of *which* attention mechanisms to examine as being potentially interpretable. Therefore, much of this work focuses on broad patterns about the kinds of values attention mechanisms produce (for example, how similar they tend to be to diagonal matrices), rather than deep dives into links between model outputs and a specific matrix of self-attention values.

We close with one last point about a popular direction in recent NLP work. In the last couple of years, it's been shown that after a certain point of model training, it's feasible to learn a reasonable simulation of a complicated concept such as text helpfulness through human feedback, using techniques such as RLHF [Ouyang et al., 2022]. One might reasonably wonder whether the determination of precisely which information is important in an instance, as we examined in section 3.4, could also be learned through such means.

However, we caution that human judgments are much better suited for determining which information would make up a *plausible* explanation– i.e., which information *they* would use in performing the specified task– than for determining which information a *model* uses, since a model may well perform the task differently than they do. In other words, this kind of human feedback incorporated into the training of an explainable model could help to boost the plausibility of that model's explanations, but not necessarily the explanations' faithfulness.

Chapter 4

Considerations for Ground-Truth Evaluations of Explanatory Methods

In the last chapter, we were focused on testing the most conservative version of whether attention mechanisms "explain" a model, by indicating which immediately preceding *intermediate representations* that model uses most heavily in determining its output. That said, the broader, more ambitious problem of determining which *inputs* the model used most heavily is still very important, and is an area of active interest in model explainability and analysis. While there are certainly cases where non-saliency-based methods, such as those referencing influential training examples [Koh and Liang, 2017; Han et al., 2020] or those expressing explanations in natural language [Wei et al., 2022; Tanneru et al., 2024], offer advantages, the identification of salient tokens has an important role to play in hinting at how models work. New survey papers cataloguing recent developments in proposed salience-based explanatory methods regularly appear (e.g., Madsen et al., 2022; Gurrapu et al., 2023; Lyu et al., 2024). Many of these methods vary in computational complexity, their assumptions about the underlying structure of the task being explained, ability to be applied contrastively, etc., but there are still usually multiple viable options that could be applied in a particular use case. It's therefore key for us to have good-quality meta-evaluations to help us find and vet proposed methods for determining these important inputs.

Consider the following scenario. An NLP researcher or practitioner wishes to investigate which input tokens a model uses most heavily in generating a particular output, to gain a better understanding of which

circumstances tend to contribute to a particular model behavior. Which of the many proposed methods for recovering those important input tokens should they use? How to determine which explanation method works the best in their use case?

One common way to investigate whether a proposed saliency-based explanation method works is to deploy that explanation method on a dataset constructed to have ground-truth annotations of which information is important [Yin and Neubig, 2022; Hu and Levy, 2023; Vamvas and Sennrich, 2021; Ferrando et al., 2023; Jumelet and Zuidema, 2023; Ma et al., 2023, *inter alia*].[1] Specifically, such datasets are constructed so that each instance of text has a "target" label or token that requires information from *at least one* of some annotated preceding "evidence" tokens to be produced. In other words, if a model is able to correctly produce the target given the earlier input of that instance, then it must have used information from at least some of the evidence, which points us to a narrow set of tokens that a good-quality explanation method should highlight. If an explanation method doesn't efficiently point to any information in the evidence, that's an indicator that it's not particularly useful in that setting. Here, it's worth revisiting a point that we made in the previous chapter, about the role of ground truth in structuring experiments. Because we are now focused on explanatory methods' identification of which *input*-level information is important, we now have the option of using known ground truth about which information is required to make a decision, in carefully curated test sets as described above. But can we just use *any* ground-truth-annotated test set?

For the purpose of vetting a proposed explanatory method, when choosing a test set with ground-truth annotations of salient information to recover, which characteristics of that test set matter? In particular,

1. **Does it matter if that set of ground-truth test instances is out of distribution for a particular model on which an explanation method is being tested?**

2. **Does it matter if that set of ground-truth test instances is used to test one kind of behavior, such as a syntactic one, when in practice it will be deployed to test which explanation methods work for very different phenomena (for example, world knowledge)?**

[1] The same strategy has also been used as a way of choosing metrics in other settings, e.g. machine translation as in Karpinska et al. [2022].

4.1 Models and Explanatory Methods for Our Experiments

For our experiments, we consider a scenario where a user is deciding which explanation method to use. We run experiments using three widely used contemporary language models, and for each, we consider a choice between two classic, widely applicable saliency methods.

4.1.1 Models

When selecting the models for our experiments, we strike a balance between model performance, which to date means using at least fairly large models, but also still being able to backpropagate gradients through the model when loaded onto four A40 GPUs with 48GB of memory each. In practice, this means that the 7-billion-parameter versions of various models are approximately the largest we can use.

We also wish to experiment with at least one model that has been instruction-tuned, and at least one that hasn't, since users could conceivably want to deploy an explanation method in either setting.

For our experiments we therefore use:

1. Llama 2, 7 billion parameter version (`Llama-2-7b`) [Touvron et al., 2023]
2. Mistral, 7 billion parameter version (`Mistral-7B-Instruct-v0.2`) [Jiang et al., 2023]
3. OLMo 7B, 7 billion parameter version (`OLMo-7B`) [Groeneveld et al., 2024]

Note that of these versions of the above three models, Mistral has been instruction-tuned, while Llama 2 and OLMo have not.

4.1.2 Explanatory Methods

Since one of our experiments we wish to run involves a syntactic ground-truth dataset that could be used to evaluate an explanation method, and it's been demonstrated by Yin and Neubig [2022] that contrastive explanation methods are better-suited to grammatical settings, we require explanation methods that can be applied *contrastively.* Instead of explaining everything about the choice of a particular token at timestep t, a contrastive explanation [Lipton, 1990] explains what contributed to that token's output probability *over*

the output probability of a different, "foil" token. In a grammatical setting, this could mean, for example, explaining the choice of verb conjugation rather than the choice of that particular verb.

In addition, some methods to explain the choice of a particular token are cheaper, requiring a number of passes through the model that is linear (or constant) in the size of the input rather than quadratic or worse, and therefore make it computationally feasible to scale up experiments beyond a handful of examples. To better represent a hypothetical use case in which this is an important consideration for the user, we therefore restrict ourselves to relatively cheap methods computationally.

These two restrictions lead us to a choice between the following two classic methods, both of which involve taking the gradient of the target output token's probability:

1. Gradient of target with respect to input embeddings, L1 norm [Baehrens et al., 2010; Simonyan et al., 2013; Li et al., 2015; Yin and Neubig, 2022]

2. Dot product of gradient of target with respect to input embeddings and corresponding token vector [Shrikumar et al., 2016; Denil et al., 2014; Yin and Neubig, 2022]

These two methods are cheap to deploy, they present among the fewest restrictions about the task on which they'll be deployed, and they can be used contrastively (which is ideal for syntactically-based ground-truth test sets). While they are both gradient-based and therefore related, prior work has established that these two methods in practice can frequently disagree with each other [Ancona et al., 2017], making them distinct.

To answer the two research questions at the end of this chapter's introduction, we will repeatedly compare these two explanation methods on all three models we have laid out, with the key variation being the annotated ground-truth test sets used to compare the explanation methods.

4.2 Grammatical Experiments: Testing the Impact of Ground Truth Test Sets Being In- vs. Out-of-Distribution

Grammatical phenomena are an attractive test bed for annotating ground truths because there are known rules by which grammar operates, and it's relatively easy to isolate which elements of a sentence are grammatically relevant in certain situations. From CoLA Warstadt et al. [2019] to BLiMP [Warstadt et al., 2020],

we're accustomed to probing models using these grammatical examples that have been constructed not to test "content," and therefore have unusual (if not nonsensical) semantics. The potential problem is that in doing so, these sentences are also moved much further away from the model's training distribution and typical use of these models.

For example, consider the following two typical sentences from the subject-verb agreement distractor subset of BLiMP (with different tokens split by whitespace, in this case by Mistral's tokenizer):

These teen **agers** who **aren** ' t revealing T ina **are** con cur ring .
A nie **ce** of most sen ators **hasn** ' t desc ended most sl opes .

We refer to the bolded cyan tokens as the "target" tokens to be explained. From a grammatical standpoint, we are able to isolate *precisely* which tokens in the pre-target-token "prefix," down to the word piece level, matter for deciding the conjugation of the target tokens; such "evidence" tokens in the prefix are annotated in bolded dark blue. Note that if we were to mask out all evidence tokens in the prefix, we would have no way of knowing whether to use the singular or plural form of the target verb. We also note, however, that the semantics of these two sentences seem artificial, and that this tends to be a(n intended) characteristic of BLiMP overall.

Following Yin and Neubig [2022], we use the subject/verb agreement subsets of the BLiMP dataset [Warstadt et al., 2020] (specifically the ones designed to have distractors, as the prefixes for those sentences are slightly longer and therefore pose a greater challenge to recovering evidence tokens) to investigate the efficacy of our enumerated explanatory methods. Furthermore, we also follow Yin and Neubig [2022] in using our proposed explanatory methods *contrastively* for these grammatical tests. What this means is that we consider the bolded, dark blue tokens in the sentences above as evidence *not* directly for the bolded cyan target tokens, but rather for the probability of the target token *minus* the probability of its singular/plural counterpart (in other words, we consider the dark blue tokens as evidence purely for the choice of singular/plural target verb form).

4.2.1 Results of Contrastive Evaluation of Explanatory Methods Using BLiMP

For both of our proposed explanatory methods, applied contrastively as described in the preceding paragraph, we order all the tokens in the prefix by descending resulting values, then count how many tokens in

that ordering we must pass before reaching our first piece of ground-truth evidence.[2] Once again, following previous work, we use spaCy [Honnibal et al., 2020] to annotate ground-truth evidence tokens, as the grammatical structure of these sentences is fairly straightforward. We present our results from applying this test of the proposed explanatory methods using BLiMP distractor sentences in table 4.1. (In addition to the average fraction of prefix tokens passed before reaching evidence, we also include the average number of tokens to which that is equivalent; that number, minus one, is the average number of non-evidence tokens an explanation method typically returns as being "most important.")

Model	Average fraction of prefix tokens in ranking before reaching a ground-truth piece of evidence		
	Sum of gradXval	Grad L1 norm	Random
Llama 2, 7B parameters (`Llama-2-7b`)	0.54 (4.00 tok.)	0.35 (2.63 tok.)	0.53 (3.85 tok.)
Mistral, 7B parameters (`Mistral-7B-Instruct-v0.2`)	0.50 (3.49 tok.)	0.32 (2.20 tok.)	0.53 (3.67 tok.)
OLMo, 7B parameters (`OLMo-7B`)	0.54 (3.43 tok.)	0.50 (3.24 tok.)	0.54 (3.39 tok.)

Table 4.1: The average fraction of tokens (ordered by proposed explanatory method) needed to check before reaching the first piece of ground-truth evidence in the prefix, and the number of average tokens needed to be checked to which this corresponds. The best possible value for an instance is $1/\text{length(prefix)}$; the worst is 1. For OLMo, both explanatory methods checked hover around random performance; for Llama and Mistral, the L1 norm of the gradients with respect to the input word vectors achieve moderate performance that's convincingly below random.

One thing we immediately notice is that **the recommendations for which (if any) explanation method to use are not necessarily consistent,** and can vary by model. This demonstrates that our scenario of a user deciding which explanation method to use for a specific model, rather than using a recommendation from prior work based on a different model, is a necessary step to ensuring that the eventual deployed method provides useful information.

For OLMo, neither of our tested methods is convincingly better than a random ordering at locating ground-truth evidence. Given that both of these methods are gradient-based, this indicates that for OLMo, the linear approximation to input importance represented by the gradient is a poor fit for how it processes information in practice. For Llama and Mistral, however, we see that the L1 norm of the gradient with

[2]This alternative to mean reciprocal rank accounts for longer prefix length, which doesn't necessarily increase the number of evidence tokens included, and follows recent work in this space [Yin and Neubig, 2022].

respect to the input token embeddings is moderately better than a random ordering at locating ground-truth information, indicating that there's some signal there (especially considering that the minimum fraction of prefix tokens needed to check for a particular instance is $1/\text{length(prefix)}$, not 0).

4.2.2 Perplexity of Tested BLiMP Sentences

	Average perplexity of different test sets		
Model	BLiMP Grammatical distractors	Generated test sets: Grammatical distractors	Generated test sets: Fiction protagonists
Llama 2, 7B parameters (`Llama-2-7b`)	176.08	26.03	17.41
Mistral, 7B parameters (`Mistral-7B-Instruct-v0.2`)	945.69	42.58	37.25
OLMo, 7B parameters (`OLMo-7B`)	514.59	42.91	38.06

Table 4.2: Even when the manually annotated, model-generated distractor sentences are removed from the context of the prompt/surrounding generated text in which they originally appeared, and are evaluated on their own, the perplexities of each model on its respective generated distractor test set are an order of magnitude lower than the respective perplexities for BLiMP distractor sentences. The perplexity of each model on its respective fiction-themed test set is similar to the perplexity exhibited on its respective generated distractor test set.

When we set aside our explanation methods and consider purely how these three models process the subject-verb distractor sentences from BLiMP, however, we notice a potential problem. In the first column of table 4.2, we list the average perplexity of each model on this data. Each of these numbers is in the hundreds, indicating an average branching factor of many, many tokens at each timestep.[3] These especially high perplexities hint that for these three models, these sentences from BLiMP qualify as "out of distribution," given that Hendrycks and Gimpel [2016] demonstrate a connection between predictive uncertainty and out-of-distribution data. Why might this be? As we discussed previously, existing curated test sets that isolate a particular phenomenon (typically grammatical in practice, to sidestep the difficulty of isolating all the ways in which world knowledge, etc. can be reflected in text) are designed to *only* test that grammatical phenomenon, and therefore typically have nonsensical meanings. Since we assume that most of the text

[3] As a frame of reference, on the evaluation benchmark WikiText-103, perplexity dipped below 20 back in 2018 using a worse language model than any of these three. See `https://paperswithcode.com/sota/language-modelling-on-wikitext-103` for more details.

today's models have been trained on was generated by people with communicative intent, this runs counter to the greater part of that training text, and therefore runs the risk of telling us information that's irrelevant to how the model behaves in more natural, in-distribution settings.

Existing research has established that such a risk is not hypothetical. We have a lot of evidence that models perform in a much more stable way in contexts that reflect the bulk of their training data, which is the motivating idea behind much research on out-of-distribution data detection [Fort et al., 2021; Hendrycks et al., 2020; Dai et al., 2023; Baran et al., 2023; Yuan et al., 2023]. Having a model recover on out-of-distribution data by returning to the "training manifold" as a stabilization technique has also had some success [Reichlin et al., 2022; Li et al., 2018].

In cases where we *have* gone out-of-distribution for a model, we can see many kinds of behavior that are not reflective of how the model typically behaves in normal usage. Indeed, the existence of the phenomenon of "jailbreaking" language models indicates that the way models behave in typical use *is* substantially different than the way they behave in other, more artificially constructed settings. Shen et al. [2023] indicate that many jailbreaking prompts share certain traits that are unusual in other prompts or language. Furthermore, Steindl et al. [2024] notice that model outputs post-jailbreaking prompt tend to display higher overall uncertainty, which indicates that jailbreaking might be alternatively framed as accessing a different, less stable part of the data distribution.

We also have evidence that this kind of difference can be a particular challenge when deploying explanation methods. Existing work has found that going off-distribution is a reliable way to manipulate explanations [Dombrowski et al., 2019; Anders et al., 2020]. Indeed, this is one of the arguments commonly levied against adversarial perturbations as an explainability or robustness technique [Li et al., 2023]. Chrysostomou and Aletras [2022a] have also previously reported different results for post-hoc explanation methods on in-domain versus out-of-domain data. All of this points to out-of-distribution evaluation of explanation methods as a potential problem for picking which one to use in practice.

4.2.3 Getting In-Distribution Test Sentences with Ground Truths

How will we test what's in-distribution, without just being part of the training data? If we just take sentences from the training data, then technically any token of a prefix could be part of the justification for a target

Prompt version 1:

```
Write an example sentence where a mistake with subject-verb agreement would
be likely.
Sample sentence: The girl who babysits the children doesn't like when they
misbehave.
Sample sentence: Many of the sheep owned by the shepherd don't produce wool
anymore.
Sample sentence:
```

Prompt version 2:

```
Write a sentence with a grammatical structure similar to the following
sentences, which have distractors when determining subject-verb agreement.
Sample sentence: The girl who babysits the children doesn't like when they
misbehave.
Sample sentence: Many of the sheep owned by the shepherd don't produce wool
anymore.
Sample sentence:
```

Figure 4.1: The two prompts used to generate three separate sets of raw model generations from Llama 2, Mistral, and OLMo respectively for the model-generated distractor test sets.

token, as the model may have memorized the sentence.

To get around this problem, we generate sentences from the models in question, to perform the same tests but on sentences that are in-distribution for these models, as indicated by the models' lower perplexity on these sentences. For each model, we generate text using the two prompts listed in figure 4.1.

We then manually read through the generated outputs and annotate sentences containing (1) correct grammar and (2) a distractor for subject-verb agreement. Figure 4.2 shows at least one example annotated sentence for each model, along with the post-prompt context in which it came. We also annotate the location of the verb in question, along with the locations of all evidence tokens for the choice of singular vs. plural for that verb. (All annotation has been done by the author, for both .) For each model, we annotate at least 100 such sentences, ensuring that the prefixes of each instance in our test sets are all unique. For each test set, we also sample 20 included sentences and search for exact matches on the internet, finding none for any of the three models' test sets. Given that web text makes up a huge part of each model's training corpus, finding no exact matches for our twenty sampled sentences indicates that it's very unlikely that our test sentences appeared in the model's training data, and that it's therefore very unlikely that the model is simply

regurgitating memorized text.

Prior to rerunning our evaluation of our explanatory methods from section 4.2.1, we first confirm that each model's average perplexity for its specific set of generated test sentences is lower than for BLiMP. This is especially important to check since for our subsequent tests, we will be using these sentences out of the original prompt and surrounding context in which they were naturally produced.

Even when the sentences from these model-specific test sets have their perplexity calculated without their preceding prompt/context, we see in the second column of table 4.2 that their perplexities are lower by roughly an order of magnitude, indicating that it's reasonable to consider these generated distactor test sets "in-distribution" in contrast to BLiMP.

4.2.4 Results from Rerunning Evaluation of Explanatory Methods

Having confirmed that we have a suitable contrast between our "out-of-distribution" (BLiMP) and "in-distribution" (generated distractor) test sets, we now rerun the evaluation from section 4.2.1 using each model's respective generated distractor test set. We present our results in table 4.3.

Model	Average fraction of prefix tokens in ranking before reaching a ground-truth piece of evidence		
	Sum of gradXval	Grad L1 norm	Random
Llama 2, 7B parameters (`Llama-2-7b`)	0.35 (3.69 tok.)	0.34 (3.49 tok.)	0.44 (4.56 tok.)
Mistral, 7B parameters (`Mistral-7B-Instruct-v0.2`)	0.41 (4.61 tok.)	0.21 (2.37 tok.)	0.45 (5.04 tok.)
OLMo, 7B parameters (`OLMo-7B`)	0.43 (4.11 tok.)	0.52 (4.79 tok.)	0.41 (3.86 tok.)

Table 4.3: Despite the much lower perplexities of the models on their respective test sets of manually annotated self-generated distractor sentences, the results evaluating which explanatory method(s) seem to work well look very similar to those resulting from using BLiMP distractor sentences (table 4.1), with the only slight changes being that the gradient L1 norm for OLMo is now slightly worse than random performance instead of hovering around random, and that the dot product of the input word embeddings and their gradients are now slightly better than random performance. But neither of these changes would affect recommendations for which explanatory method to use.

Perhaps surprisingly, given the drastic difference in model perplexities from evaluating on BLiMP, we see very similar evaluation results for each respective model; we continue to see that neither explanatory method suffices for OLMo, but for Llama and Mistral, we continue to see the same moderate performance of

Sample distractor generation from Llama 2:

When John bought the car, he didn't expect the engine to break after 3 months.
Sample sentence: In Japan, the kimono worn by the geisha have become one of the national symbols.
Sample sentence: **The boys at the sports club don't like when their equipment gets damaged.**
An error in subject-verb agreement happens when you have words that behave like either a subject or a verb, but not both. In this case you may choose to treat it like either a subject or verb and end up with two different subjects or verbs that need to agree with each other.
Ex: The children's book was the story I liked most.

Sample distractor generation from Mistral:

The teacher who grades the exams rigorously finds it frustrating when students hand in late assignments.

Sample sentence: **Several of the apples picked from the tree in the backyard are overripe and need to be discarded.**

Sample sentence: The CEO who manages the multinational corporation aggressively pursues expansion into new markets.

Sample sentence: **Some of the paintings displayed in the art gallery have uneven colors due to aging.**

Sample sentence: The captain who commands the submarine strictly enforces rules regarding diving depths.

Sample distractor generation from OLMo:

The students who were playing basketball during lunch have been suspended from school.
Sample sentence: People in the town where I grew up think that I should travel the world and never return to their little place.
Sample sentence: The man who the townsfolk didn't like is starting a new restaurant.

Figure 4.2: Example raw (and processed) model generations from Llama 2, Mistral, and OLMo respectively for the model-generated distractor test sets. Manually annotated sentences extracted from these generations are bolded and written in red, their target verb is in cyan, and the ground-truth evidence tokens for that choice of (singular/plural) verb is in dark blue. Ground-truth evidence tokens can include certain subject quantifiers (e.g., "Several"), the subject itself, and any earlier verb conjugations (e.g., "were") that indicate whether the post-distractor verb should be singular or plural. Note that not *all* pre-subject quantifiers count as ground-truth evidence; for example, "some of" can be used with plural nouns, which take the plural verb form, OR with singular nouns to indicate a fraction of them, which would take the singular, so we do not count it as strong evidence for choosing the correct later verb form.

the L1 norm of the input token embeddings' gradient. **We have not observed a change in our recommendations about which explanation method to use by shifting from an out-of-distribution ground-truth test to an in-distribution one,** meaning that we have found no evidence that using an out-of-distribution ground-truth-annotated test set leads to a difference conclusion than using an in-distribution one. This could indicate that in the future, it is possibly fine to use an out-of-distribution, non-model-specific one. It remains to be seen whether this result would generalize to other models or other syntactic phenomena, however.

4.3 Fictional-Knowledge-Based Experiments: Testing the Impact of the Kind of Ground Truth

Having seen a negligible shift in recommendations about explanation methods between out-of-distribution and in-distribution ground truth evaluation, we now move on to our second question: whether the type of information contained in a ground-truth evaluation impacts recommendations for explanation methods.

Even a few years ago, human evaluation of language models' fluency was a cornerstone of even English-language model evaluation [Clark et al., 2019; Ippolito et al., 2020], since this was something that models still sometimes struggled with. The language models of the last few years typically no longer struggle with syntax, and these days, grammatical phenomena are not usually the aspects of language models that researchers or lay users are most concerned with investigating. Therefore, testing explanatory methods' validity using grammar might not be especially relevant to how those explanatory methods will be deployed in practice. How will we adapt our evaluation to account for this?

Inspired by Chang et al. [2023], we have the models generate book names and the names of those books' protagonists. If a model is successfully able to generate the specific protagonist's name for a particular book, that gives us a very strong argument that the model is substantively using at least one of the tokens in the book's title. Similarly to our generated distractor prompts, we have each of our three models generate outputs in response to the prompts in figure 4.3.

As before, we then manually annotate which tokens make up the book title and which make up the protagonist's name, and also take the subset of these generated outputs for which the protagonist's name is correct for the book, discarding any extracted sentences for which the book-protagonist matching is incor-

Prompt version 1:

```
Tell me the name of a book and the name of its protagonist.
```

Prompt version 2:

```
Tell me the name of a book and the name of its protagonist.
Example: The book Great Expectations has Pip as its main character.
Example: The book The Catcher in the Rye has Holden as its main character.
Example:
```

Figure 4.3: The two prompts used to generate three separate sets of raw model generations from Llama 2, Mistral, and OLMo respectively for the model-generated fiction-based test sets.

rect. We also take care to select test instances such that in any model's test set, no book appears in more than one instance, to minimize the impact on our evaluation of any possible model propensity to generate a specific character name following a mention of a "protagonist."

For Llama and Mistral, the test sets include at least 100 instances; Mistral's test set contains 38 instances, since there were only mentions of 37 unique books other than the overwhelmingly occurring *To Kill a Mockingbird* with correctly paired protagonists in the 1000 raw generations we annotated, even with a temperature of 1.5 (which was high enough to occasionally result in mid-word degeneration of text). We also repeat the process from our generated test sets in section 4.2.3 of spot-checking whether 20 sentences from each test set appear verbatim on the internet. We still mostly find that the test sentences do not appear in internet sources, though we do find two very close (though not exact) matches for two of Mistral's sentences and two of OLMo's sentences, as well as one exact match for an OLMo test sentence. However, most sentences are still not taken verbatim from the largest, most varied source of the models' training data, so we proceed with the test sets without further filtering. We include sample generated sentences in figure 4.4. Notably, unlike during the annotation process for model-written subject-verb agreement distractor sentences, we regularly observe incorrect pairings of titles with protagonists in the raw model outputs, indicating that this task is more challenging for these language models.

We now rerun our evaluation once again, using these new model-specific generated test sets, but with one change: since we are now testing the choice of the first token with an identifiable piece of the protagonist's name, we no longer apply our evaluation methods contrastively. (In our grammatical experiments, we were

Sample fiction protagonist generation from Llama 2:

[INST] Choose the word that best completes the sentence from the choices.
Example: **The book The Great Gatsby has Gatsby as its protagonist.**
Ex: The book The Invisible Man has H.G. Wells as its protagonist.
Ex: **The book The Wonderful Wizard of Oz has Dorothy as its protagonist.**
[INST] Choose the word that best completes each sentence.
Example: **The book The Adventures of Sherlock Holmes has Sherlock Holmes as its protagonist.**
Example: **The book Frankenstein has Victor Frankenstein as its protagonist.**
Example: **The book The Time Machine has The Time Traveller as its protagonist.**
Example: The book Animal Farm has a young, male pig as its protagonist.
Ex: The book The Old Man and the Sea has an old man as its protagonist.
Ex: The book Madame B

Sample fiction protagonist generation from Mistral:

I'd be happy to help, but I'll need more information to provide an accurate answer. Here's an example pairing: **The Name of the Wind by Patrick Rothfuss features the protagonist Kvothe.**

Sample fiction protagonist generation from OLMo:

Who is Sherlock Holmes, his name is Dr. Watson's best friend.
Example: **The main character of A Midsummer Night's Dream is Helena.**
Example: **The main character of The Wonderful Wizard of Oz is Dorothy, the girl from Kansas.**
Example: **In the book Pride and Prejudice, the most important character is Elizabeth Bennet.**
Example: **The main character of the book A Tale of Two Cities is Charles Darnay.**

Figure 4.4: Example raw (and processed) model generations from Llama 2, Mistral, and OLMo respectively for the model-generated distractor test sets. As in figure 4.2, manually annotated sentences extracted from these generations are bolded and written in red, the protagonist's name (the target) is in cyan, and the preceding ground-truth evidence tokens (i.e., the work's title) for that protagonist is in dark blue. Part of the manual annotation process involved checking the accuracy of work-protagonist match, and we discarded candidate sentences for which this was incorrect (e.g., H.G. Wells being listed as the protagonist of *The Invisible Man*). As in the case of Helena for *A Midsummer Night's Dream*, we used sentences as long as the character's name mentioned is commonly referred to as one of the protagonists of the work; we did not require the generated sentence to list all protagonists, especially seeing as our experiments only traced back from the first token of the protagonist's name anyway. We also required the protagonist to be named in the sentence, not just described ("The Time Traveller" is acceptable as that is how the protagonist of *The Time Machine* is referred to in the book; "a young, male pig" as the protagonist of *Animal Farm* is not).

interested only in what led to the correct choice of verb conjugation, not the choice of verb itself.) The full results are in table 4.4, but at a high level, we notice that **for Llama and Mistral, our recommendations for evaluation methods have changed**. On this task, *neither* of our tested evaluation methods still performs better than a random ordering.

Checking the average perplexity of these models on their respective fiction test sets, this difference does not appear to be due to perplexity. Comparing models' average perplexity in table 4.2 on these datasets (column 3) to models' average perplexity on BLiMP and their respective generated grammatical distractor test sets, we see that the models' perplexity on these fiction-based test sets is far closer to their perplexity on the grammatical distractor ones, within just a few points. So perplexity doesn't suffice to explain why the explanation method results for the fiction test sets look so different.

Model	Average fraction of prefix tokens in ranking before reaching a ground-truth piece of evidence		
	Sum of gradXval	Grad L1 norm	Random
Llama 2, 7B parameters (`Llama-2-7b`)	0.24 (5.21 tok.)	0.34 (7.43 tok.)	0.21 (4.45 tok.)
Mistral, 7B parameters (`Mistral-7B-Instruct-v0.2`)	0.18 (4.87 tok.)	0.26 (7.21 tok.)	0.15 (4.13 tok.)
OLMo, 7B parameters (`OLMo-7B`)	0.23 (4.07 tok.)	0.38 (6.59 tok.)	0.24 (4.29 tok.)

Table 4.4: Unlike merely lowering the perplexity of test sentences yet keeping the same fundamental kind of ground truth as in table 4.3, changing the kind of ground truth to reflect knowledge about the protagonists of fictional works rather than English grammatical rules *does* change takeaways about which proposed explanatory methods to use. In particular, for Llama and Mistral, the recommendation shifts to indicate that *neither* proposed explanation method has better than random performance in recovering ground truth (and in fact, they both have worse performance).

How much of this difference is due to the fact that the prefixes are generally longer in these sentences, though? The average number of tokens in the BLiMP and generated grammatical distractor prefix test sets is roughly 7–11, while the average number of tokens in the fiction-based test sets varies between 18 and 28 per test set. Perhaps this difference in the results is due to the increased difficulty of finding evidence in a longer prefix.

To test this, we rerun our experiments from table 4.4, but splitting each test set into its half with the shortest prefixes and its half with the longest prefixes. We present our results in figure 4.5, with each value representing the fraction of each prefix needed before reaching evidence divided by the corresponding

random performance (lower is better, and anything at or above a value of 1 signifies random or worse performance).

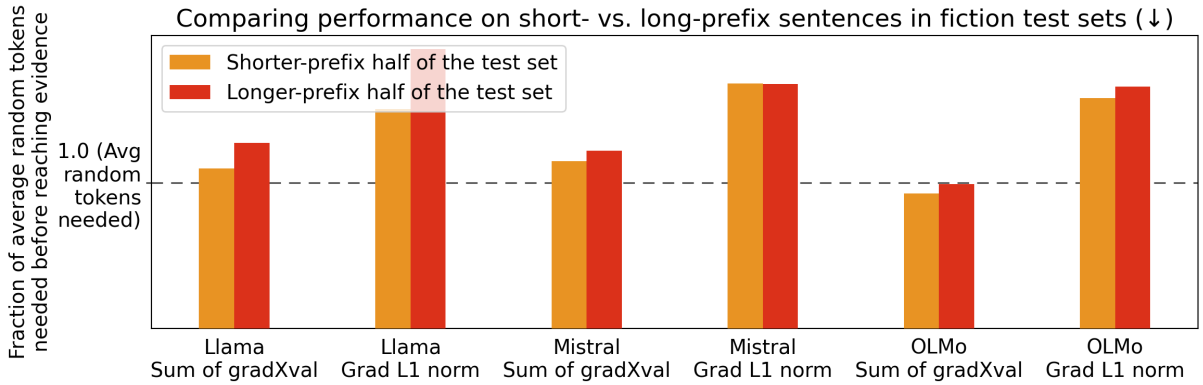


Figure 4.5: A comparison of explanation method performance on shorter-prefix sentences versus longer-prefix sentences for each model's respective fiction-based test set (lower is better). In almost all cases, methods still performed on par with, or worse than, a random ordering at recovering ground-truth evidence. The two exceptions are longer-prefix sentences for Llama and shorter-prefix sentences for OLMo, both using the sum of gradients multiplied by their corresponding input embedding values.

After removing the half of each fiction protagonist dataset consisting of the longest prefixes and rerunning our tests, though, we still don't see results that echo those from either section 4.2.1 or section 4.2.4. While for Llama we do see a bit of a difference in explanation method performance by model length, interestingly, we barely see such a difference for Mistral or OLMo.

The key point we emphasize is that even if our intended deployment setting justifies a comparison to only one of these halves of the test sets, the recommendation for Llama and Mistral is still different than it was for either grammatical test set: in the grammatical setting, the recommendation for both Llama and Mistral was to use the gradient L1 norm, and here neither method performs on par with random guessing. We also note that, conditioning on any particular model, **the relative ordering of which method performs better is identical regardless of whether we look at only the shorter-prefix or longer-prefix sentences. Sifting out the longer-length prefixes doesn't account for the difference in recommendations.**

4.4 Related Work

Besides evaluation of explanation methods using test sets annotated with ground truth, there are other experimental paradigms for choosing or advocating for particular explanation methods. We discuss two such alternatives here.

One alternative way of evaluating explanation methods is to check their correlation with other proposed explanation methods, sometimes termed "agreement as evaluation." While this strategy for evaluating explanation methods has been used many times [Jain and Wallace, 2019; Mohankumar et al., 2020; Meister et al., 2021; Abnar and Zuidema, 2020], there has been corresponding work highlighting issues with this methodology. Most notably, Neely et al. [2022] raise problems with using meta-evaluations that rely on correlating different proposed explanation methods with each other as a measure of validity.

Separately, Chan et al. [2022b]; Fanconi et al. [2023]; Wu et al. [2023]; Chrysostomou and Aletras [2022b]; Ghoshal et al. [2022] are among the many papers that use automated metrics corresponding to sufficiency and comprehensiveness of explanations (and by extension, explanation methods) proposed in DeYoung et al. [2020]. These metrics have the strength of working on arbitrary data, circumventing the need for a specifically annotated test set. However, recent work (Zhao and Aletras, 2023 and Hsia et al., 2024) has pointed out shortcomings of these metrics.

In sum, the explainability/interpretability field is in the midst of an ongoing conversation about which kind of evaluations to use for explanation methods; here, we have focused on ground truth recovery from ground-truth-annotated test sets, and recommendations for strengthening the correspondence between the recommendations from such an experiment, and explanation method behavior at deployment.

4.5 Takeaways

Based on this set of experiments, we have an indication that **recommendations about evaluation methods to use for a particular model can change if the kinds of ground-truth information used to vet those evaluation methods shift.** While our experiments support that decisions about which explanation to use for a model don't shift in response to swapping in-distribution for out-of-distribution text in the ground-truth-annotated test sets, we *do* see that other kinds of differences matter:

1. The nature of the link between the evidence and the target (e.g., syntactic or semantic) should be fairly close to the kind that the explanation method will be deployed in practice for.

2. To a lesser extent, average prefix length in a test set will affect results.

This indicates that, when constructing an experiment to determine which explanation method to use for a particular model, it might not be a problem to use a ground-truth-annotated test set that wasn't developed for that particular model, *but* the information contained in those ground truths should be similar to the kind of information we expect to retrieve in our explanations in deployment.

Chapter 5

Conclusion

In this thesis, we have discussed three projects, all of which are related to experiment design for research questions where an NLP model is the object of study. In chapter 2, we translated a high-level question about the transfer of dataset bias to learned model bias into a precise hypothesis that was statistically testable. Similarly, in chapter 3, we made the case for experiments verifying what it would mean for a particular module, the attention mechanism, to qualify as "interpretable." And finally, in chapter 4, we considered how researchers and NLP practitioners choose which explainability methods to use to help them answer questions of their own choosing about models, focusing on empirical evidence for recommendations about which aspects of such an experiment demand careful attention.

With that said, there remain several key challenges in the interpretability, explainability, and analysis of NLP models, and with that in mind, we shift our attention to future work directions for the remainder of this chapter.

5.1 Future Work

One direction of future work we are interested in is **the impact of other common experiment-structuring choices,** besides choice of ground-truth dataset that we have explored in chapter 4. Given that explanation methods disagreeing with each other has been observed across several different families of methods [Ding and Koehn, 2021], NLP researchers and practitioners need further guidelines about how to select an explanation method for their particular use case, and one key part of such experiments that we haven't tested is the

impact of the choice of *metric used.* This is partly related to the discussion around the proposed metrics of sufficiency and comprehensiveness proposed in DeYoung et al. [2020] [Zhao and Aletras, 2023; Hsia et al., 2024]; as a community, how we measure an explanation (method)'s quality is still under discussion, and it is currently unclear how much different choices of metric currently made in this space affect results.

In the interest of ensuring that the conclusions from an experiment on how to explain a model are relevant during deployment, we also propose to **integrate the plan for using a model explanation method into the method-selecting experiments.** For example, existing work has raised the specific challenges of applying model explanation methods to determine model fairness, a key potential use case. In particular, Balkir et al. [2022] highlight that model explanation methods (and by connection, the experiments used to evaluate them) often focus on *procedural fairness* rather than *outcome fairness*, while in practice they're usually used to evaluate the latter. Therefore, it might be beneficial to incorporate outcome fairness as a factor in vetting explanation methods, not just after one has been selected.

The final direction we propose is **improving on experiment design for free-text explanations.** As NLP moves towards framing tasks as free-text generation more and more frequently, it becomes more and more important to develop methods that explain a model's decision for these more complex output spaces. Machine-generated text has also improved to such a degree that it's increasingly promising to use machine-generated text as explanations for models' underlying decision processes, since free-form text is a form of explanation with which people are very comfortable [Hendricks et al., 2016; Camburu et al., 2018; Shwartz et al., 2020]. However, as a research community, we are still in the process of determining which aspects of free-text explanations can be distilled into testable hypotheses that tie them more to the actual function of the model. There has been a small amount of existing work on faithfulness of free text to a model's decision. Chan et al. [2022a] and Hase et al. [2020] check whether machines can use free-text rationales generated by another model to predict that model's output, but for tasks with small output spaces, it can be fairly easy to guess an answer based off as little as a relevant keyword, thus making it possible for the bulk of an explanation to misrepresent a model's decision-making process and still be sufficient to point an auxiliary model to its corresponding answer. A separate line of work has focused on models that jointly produce an output and a free-text explanation given the same latent representation as input, using gradient-based attribution methods to extract input saliences for both output forms and comparing them [Wiegreffe

et al., 2021]. In this vein, we propose to investigate, first, which information human users in specific settings such as scientific literature search want from free-text explanations that they can't get (or have more trouble getting) from more constrained forms of explanation. Based on the results of this study, we can then focus experiments on evaluating those specific traits of free-text explanation.

Bibliography

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus H. Gross. 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*.

Christopher J. Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. 2020. Fairwashing explanations with off-manifold detergent. *ArXiv*, abs/2007.09969.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016a. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1545–1554, San Diego, California. Association for Computational Linguistics.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016b. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. *arXiv preprint arXiv:1606.07298*.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(61):1803–1831.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Esma Balkir, Svetlana Kiritchenko, Isar Nejadgholi, and Kathleen Fraser. 2022. Challenges in applying explainability methods to improve the fairness of NLP models. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, pages 80–92, Seattle, U.S.A. Association for Computational Linguistics.

Mateusz Baran, Joanna Baran, Mateusz Wójcik, Maciej Zięba, and Adam Gonczarek. 2023. Classical out-of-distribution detection methods benchmark in text classification tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 119–129, Toronto, Canada. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. 2022. Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900, Dublin, Ireland. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. In *International Conference on Learning Representations*.

Jonathon Byrd and Zachary Chase Lipton. 2018. What is the Effect of Importance Weighting in Deep Learning? In *International Conference on Machine Learning*.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356:183 – 186.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Aaron Chan, Shaoliang Nie, Liang Tan, Xiaochang Peng, Hamed Firooz, Maziar Sanjabi, and Xiang Ren. 2022a. Frame: Evaluating simulatability metrics for free-text rationales. *ArXiv*, abs/2207.00779.

Aaron Chan, Maziar Sanjabi, Lambert Mathias, Liang Tan, Shaoliang Nie, Xiaochang Peng, Xiang Ren, and Hamed Firooz. 2022b. UNIREX: A unified learning framework for language model rationale extraction. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 51–67, virtual+Dublin. Association for Computational Linguistics.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore. Association for Computational Linguistics.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Brussels, Belgium. Association for Computational Linguistics.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

George Chrysostomou and Nikolaos Aletras. 2022a. An empirical study on explanations in out-of-domain settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6920–6938, Dublin, Ireland. Association for Computational Linguistics.

George Chrysostomou and Nikolaos Aletras. 2022b. Flexible instance-specific rationalization of nlp models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10545–10553.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Yi Dai, Hao Lang, Kaisheng Zeng, Fei Huang, and Yongbin Li. 2023. Exploring large language models for multi-modal out-of-distribution detection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5292–5305, Singapore. Association for Computational Linguistics.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35, Florence, Italy. Association for Computational Linguistics.

Misha Denil, Alban Demiraj, and Nando de Freitas. 2014. Extraction of salient sentences from labelled documents. *ArXiv*, abs/1412.6815.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the ACM SIGKDD International Conference on Knowledge Ciscovery and Data mining*.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052, Online. Association for Computational Linguistics.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. 2019. Explanations can be manipulated and geometry is to blame. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Jacob Eisenstein. 2022. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4326–4331, Seattle, United States. Association for Computational Linguistics.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Claudio Fanconi, Moritz Vandenhirtz, Severin Husmann, and Julia Vogt. 2023. This reads like that: Deep learning for interpretable natural language processing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14067–14076, Singapore. Association for Computational Linguistics.

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA. Association for Computing Machinery.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretation difficult. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Stanislav Fort, Jie Jessie Ren, and Balaji Lakshminarayanan. 2021. Exploring the limits of out-of-distribution detection. *Neural Information Processing Systems*.

Alona Fyshe, Leila Wehbe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. 2015. A compositional and interpretable semantic space. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reza Ghaeini, Xiaoli Z. Fern, and Prasad Tadepalli. 2018. Interpreting recurrent and attention-based neural models: a case study on natural language inference. *arXiv preprint arXiv:1808.03894*.

Amirata Ghorbani, Abubakar Abid, and James Zou. 2017. Interpretation of Neural Networks is Fragile. *arXiv preprint arXiv:1710.10547*.

Asish Ghoshal, Srinivasan Iyer, Bhargavi Paranjape, Kushal Lakhotia, Scott Wen-tau Yih, and Yashar Mehdad. 2022. Quaser: Question answering with scalable extractive rationalization. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 1208–1218, New York, NY, USA. Association for Computing Machinery.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Sindhu C. M. Gowda, Shalmali Joshi, Haoran Zhang, and Marzyeh Ghassemi. 2021. Pulling up by the causal bootstraps: Causal data augmentation for pre-training debiasing. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, A. Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Daniel Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hanna Hajishirzi. 2024. Olmo: Accelerating the science of language models. *ArXiv*, abs/2402.00838.

Sai Gurrapu, Ajay Kulkarni, Lifu Huang, Ismini Lourentzou, and Feras A. Batarseh. 2023. Rationalization for explainable nlp: a survey. *Frontiers in Artificial Intelligence*, 6.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Confer-*

ence of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. Before Name-calling: Dynamics and Triggers of Ad Hominem Fallacies in Web Argumentation. *arXiv preprint arXiv:1802.06613*.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online. Association for Computational Linguistics.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating visual explanations. In *Computer Vision – ECCV 2016*, pages 3–19, Cham. Springer International Publishing.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. 2022. Training compute-optimal large language models. *ArXiv*, abs/2203.15556.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. Published software. DOI: 10.5281/zenodo.1212303.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jennifer Hsia, Danish Pruthi, Aarti Singh, and Zachary Lipton. 2024. Goodhart's law applies to NLP's explanation benchmarks. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1322–1335, St. Julian's, Malta. Association for Computational Linguistics.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Yangfeng Ji and Noah A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Nitish Joshi and He He. 2022. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3668–3681, Dublin, Ireland. Association for Computational Linguistics.

Jaap Jumelet and Willem Zuidema. 2023. Transparency at the source: Evaluating and interpreting language models with access to the true distribution. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4354–4369, Singapore. Association for Computational Linguistics.

Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-End Bias Mitigation by Modelling Biases in Corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Marzena Karpinska, Nishant Raj, Katherine Thai, Yixiao Song, Ankita Gupta, and Mohit Iyyer. 2022. DEMETR: Diagnosing evaluation metrics for translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9540–9561, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing (EMNLP), pages 7057–7075, Online. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2021. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4547–4568, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*.

Allison Lahnala, Charles Welch, Béla Neuendorf, and Lucie Flek. 2022. Mitigating toxic degeneration with empathetic data: Exploring the relationship between toxicity and empathy. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4926–4938, Seattle, United States. Association for Computational Linguistics.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. 2017. Interactive Visualization and Manipulation of Attention-based Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–126.

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheoneum Park, and Kyomin Jung. 2021. Crossaug: A contrastive data augmentation method for debiasing fact verification models. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in NLP. *arXiv preprint arXiv:1506.01066*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Lucy Li and David Bamman. 2021. Gender and representation bias in GPT-3 generated stories. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 48–55, Virtual. Association for Computational Linguistics.

Miao Li, Kenji Tahara, and Aude Billard. 2018. Learning task manifolds for constrained object manipulation. *Auton. Robots*, 42(1):159–174.

Xinzhe Li, Ming Liu, Shang Gao, and Wray Buntine. 2023. A survey on out-of-distribution evaluation of neural nlp models. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6683–6691. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Peter Lipton. 1990. Contrastive explanation. *Royal Institute of Philosophy Supplement*, 27:247 – 266.

Zachary C Lipton. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Linqing Liu, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. 2022. Challenges in generalization in open domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2014–2029, Seattle, United States. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association of Computational Linguistics*, 6:63–75.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards Faithful Model Explanation in NLP: A Survey. *Computational Linguistics*, pages 1–67.

Liang Ma, Shuyang Cao, Robert L Logan IV, Di Lu, Shihao Ran, Ke Zhang, Joel Tetreault, and Alejandro Jaimes. 2023. BUMP: A benchmark of unfaithful minimal pairs for meta-evaluation of faithfulness metrics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12788–12812, Toronto, Canada. Association for Computational Linguistics.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Comput. Surv.*, 55(8).

Gary F. Marcus. 2018. Deep learning: A critical appraisal. *ArXiv*, abs/1801.00631.

André Martins and Ramón Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

Clara Meister, Stefan Lazov, Isabelle Augenstein, and Ryan Cotterell. 2021. Is sparse attention more interpretable? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 122–129, Online. Association for Computational Linguistics.

David Alvarez Melis and Tommi Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space.

Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216, Online. Association for Computational Linguistics.

Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. 2019. Interrogating the explanatory power of attention in neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 221–230, Hong Kong. Association for Computational Linguistics.

Michael Neely, Stefan F. Schouten, Maurits J. R. Bleeker, and Ana Lucic. 2022. A song of (dis)agreement: Evaluating the evaluation of explainable artificial intelligence in natural language processing. In *Proceedings of the International Conference on Hybrid Human-Artificial Intelligence (HHAI-22)*.

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra-Aimée Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *ArXiv*, abs/2306.01116.

Belinda Phipson and Gordon K Smyth. 2010. Permutation p-values should never be zero: Calculating exact p-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1).

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Conference on Empirical Methods in Natural Language Processing*.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.

Alfredo Reichlin, Giovanni Luca Marchetti, Hang Yin, Ali Ghadirzadeh, and Danica Kragic. 2022. Back to the Manifold: Recovering from Out-of-Distribution states. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8660–8666.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew Peters, and Matt Gardner. 2022. Tailor: Generating and perturbing text with semantic controls. In *Proceedings of the 60th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 3194–3213, Dublin, Ireland. Association for Computational Linguistics.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Roy Schwartz and Gabriel Stanovsky. 2022. On the limitations of dataset balancing: The lost battle against spurious correlations. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2182–2194, Seattle, United States. Association for Computational Linguistics.

Sofia Serrano, Jesse Dodge, and Noah A. Smith. 2023. Stubborn Lexical Bias in Data and Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8131–8146, Toronto, Canada. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Xinyue Shen, Zeyuan Johnson Chen, Michael Backes, Yun Shen, and Yang Zhang. 2023. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *ArXiv*, abs/2308.03825.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2021. Societal biases in language generation: Progress and challenges. In *Proceedings of the 59th Annual Meeting of the Association for*

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4275–4293, Online. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not just a black box: Learning important features through propagating activation differences. *ArXiv*, abs/1605.01713.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online. Association for Computational Linguistics.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Ryan Steed, Swetasudha Panda, Ari Kobren, and Michael Wick. 2022. Upstream Mitigation Is *Not* All You Need: Testing the Bias Transfer Hypothesis in Pre-Trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3524–3542, Dublin, Ireland. Association for Computational Linguistics.

Sebastian Steindl, Ulrich Schäfer, Bernd Ludwig, and Patrick Levi. 2024. Linguistic obfuscation attacks and large language model uncertainty. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 35–40, St Julians, Malta. Association for Computational Linguistics.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. 2017. SPINE: Sparse interpretable neural embeddings. *arXiv preprint arXiv:1711.08792*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Sree Harsha Tanneru, Chirag Agarwal, and Himabindu Lakkaraju. 2024. Quantifying uncertainty in natural language explanations of large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 1072–1080. PMLR.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. *ArXiv*, abs/1804.08117.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An Empirical Study on Robustness to Spurious Correlations using Pre-trained Language Models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Martin Tutek and Jan Snajder. 2020. Staying true to your word: (how) can attention become explanation? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 131–142, Online. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2021. On the limits of minimal pairs in contrastive evaluation. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 58–68, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. *ArXiv*, abs/1909.11218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring Association Between Labels and Free-Text Rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Chenwang Wu, Xiting Wang, Defu Lian, Xing Xie, and Enhong Chen. 2023. A causality inspired framework for model interpretation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 2731–2741, New York, NY, USA. Association for Computing Machinery.

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. 2022. Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2660–2676, Dublin, Ireland. Association for Computational Linguistics.

Fan Yang, Arjun Mukherjee, and Eduard Dragut. 2017. Satirical news detection and analysis using attention mechanism and linguistic features. *arXiv preprint arXiv:1709.01189*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. *ArXiv*, abs/2202.10419.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, FangYuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis,

and llms evaluations. In *Advances in Neural Information Processing Systems*, volume 36, pages 58478–58507. Curran Associates, Inc.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Runtian Zhai, Chen Dan, J. Zico Kolter, and Pradeep Ravikumar. 2023. Understanding Why Generalized Reweighting Does Not Improve Over ERM. In *Proceedings of the International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*.

Yi Zhang and Jitao Sang. 2020. Towards accuracy-fairness paradox: Adversarial example-based data augmentation for visual debiasing. *Proceedings of the 28th ACM International Conference on Multimedia*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Zhixue Zhao and Nikolaos Aletras. 2023. Incorporating attribution importance for improving faithfulness metrics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4732–4745, Toronto, Canada. Association for Computational Linguistics.

Ruiqi Zhong, Steven Shao, and Kathleen McKeown. 2019. Fine-grained sentiment analysis with faithful attention. *ArXiv*, abs/1908.06870.

Yichu Zhou and Vivek Srikumar. 2022. A closer look at how fine-tuning changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland. Association for Computational Linguistics.