Competency Problems:
On Finding and Removing Artifacts in Language Data

Anonymous EMNLP submission

Abstract

Much recent work in NLP has documented dataset artifacts, bias, and spurious correlations between input features and output labels. However, how to tell which features have "spurious" instead of legitimate correlations is typically left unspecified. In this work we argue that for complex language understanding tasks, *all* simple feature correlations are spurious, and we formalize this notion into a class of problems which we call *competency problems*. For example, the word "amazing" on its own should not give information about a sentiment label independent of the context in which it appears, which could include negation, metaphor, sarcasm, etc. We theoretically analyze the difficulty of creating data for competency problems when human bias is taken into account, showing that realistic datasets will increasingly deviate from competency problems as dataset size increases. This analysis gives us a simple statistical test for dataset artifacts, which we use to show more subtle biases than were described in prior work, including demonstrating that models are inappropriately affected by these less extreme biases. Our theoretical treatment of this problem also allows us to analyze proposed solutions, such as making local edits to dataset instances, and to give recommendations for future data collection and model design efforts that target competency problems.

1 Introduction

Attempts by the natural language processing community to get machines to understand language or read text are often stymied in part by issues in our datasets (Chen et al., 2016; Sugawara et al., 2018). Many recent papers have shown that popular datasets are prone to shortcuts, dataset artifacts, bias, and spurious correlations (Jia and Liang, 2017; Rudinger et al., 2018; Costa-jussà et al., 2019). While these empirical demonstrations of deficiencies in the data are useful, they often leave unanswered fundamental questions of what exactly makes a correlation "spurious", instead of a feature that is legitimately predictive of some target label.

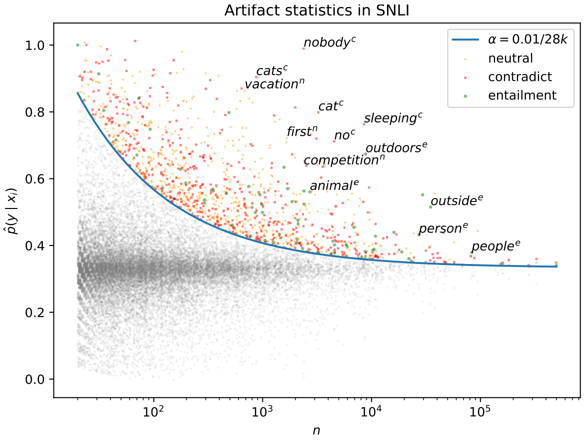

Figure 1: A statistical test for deviation from a competency problem, where no individual feature (here words) should give information about the class label, plotting the number of occurrences of each word against the conditional probability of the label given the presence of the word. The label associated with each point is marked by color and superscript. All features above the blue line have detectable correlation with class labels, using a very conservative Bonferroni-corrected statistical test.

In this work we attempt to address this question theoretically. We begin with the assumption that in a language understanding problem, no single feature on its own should contain information about the class label. That is, *all* simple correlations between input features and output labels are spurious: $p(y|x_i)$, for any feature x_i, should be uniform over the class label. We call the class of problems that meet this assumption *competency problems* (§2).[1]

This assumption places a very strong restriction on the problems being studied, but we argue that it

[1] Our use of the term "competency problems" is inspired by, but not identical to, the term "competence" in linguistics. We are referring to the notion that humans can understand essentially any well-formed utterance in their native language.

is a reasonable description of complex language understanding problems. Consider, for example, the problem of sentiment analysis on movie reviews. A single feature might be the presence of the word *"amazing"*, which could be legitimately correlated with positive sentiment in some randomly-sampled collection of actual movie reviews. However, that correlation tells us more about word frequency in movie reviews than it tells us about a machine's ability to understand the complexities of natural language. A competent speaker of a natural language would know that *"amazing"* can appear in many contexts that do not have positive sentiment and would not base their prediction on the presence of this feature alone. That is, the information about the sentiment of a review, and indeed the *meaning* of natural language, is contained in complex feature interactions, not in isolated features. To evaluate a machine's understanding of language, we must remove all simple feature correlations that would allow the machine to predict the correct label without considering how those features interact.

Collecting data that accurately reflects the assumptions of a competency problem is very challenging, especially when humans are involved in creating it. Humans suffer from many different kinds of bias and priming effects, which we collectively model in this work with rejection sampling during data collection. We theoretically analyze data collection under this biased sampling process, showing that any amount of bias will result in increasing probability of statistically-significant spurious feature correlations as dataset size increases (§3).

This theoretical treatment of bias in data collection gives us a new, simple measure of data artifacts (§3.2), which we use to explore artifacts in several existing datasets (§4). Figure 1 revisits prior analyses on the SNLI dataset (Bowman et al., 2015) with our statistical test. An analysis based on pointwise mutual information (e.g., Gururangan et al., 2018) would correspond to a horizontal line in that figure, missing many features that have less extreme but still significant correlations with class labels. These less extreme correlations still lead models to overweight simple features. The problem of bias in data collection is pervasive and not easily addressed with current learning techniques.

Our framework also allows us to examine the theoretical impact of proposed techniques to mitigate bias, including performing local edits after data collection (§5) and filtering collected data (§6). We derive properties of any local edit procedure that must hold for the procedure to effectively remove data artifacts. These proofs give dataset builders tools to monitor the data collection process to be sure that resultant datasets are as artifact-free as possible. Our analysis of local edits additionally suggests a strong relationship to *sensitivity* in boolean functions (O'Donnell, 2014), and we identify gaps in the theory of sensitivity that need to be filled to properly account for bias in sampled datasets.

We believe our theoretical analysis of these problems provides a good starting point for future analyses of methods to improve NLP data collection, as well as insights for inductive biases that could be introduced to better model competency problems.

2 Competency Problems

We define a *competency problem* to be one where the marginal distribution over labels given any single feature is uniform. For our analysis, we restrict ourselves to boolean functions: we assume an input vector $\mathbf{x}$ and an output value y, where $\mathbf{x} \in \{0,1\}^d$ and $y \in \{0,1\}$.[2] In this setting, competency means $p(y|x_i) = 0.5$ for all i. In other words, the information mapping $\mathbf{x}$ to y is found in complex feature interactions, not in individual features.

Our core claim is that language understanding requires composing together many pieces of meaning, each of which on its own is largely uninformative about the meaning of the whole. We do not believe this claim is controversial or new, but its implications for posing language understanding as a machine learning problem are underappreciated and somewhat counterintuitive. If a model picks up on individual feature correlations in a dataset, it has learned something *extra-linguistic*, such as information about human biases, not about how words come together to form meaning, which is the heart of natural language understanding. To push machines towards linguistic competence, we

[2] Boolean functions are quite general, and many machine learning problems can be framed this way. For NLP, consider that before the rise of embedding methods, language was often represented in machine learning models as bags of features in a very high-dimensional feature space, exactly as we are modeling the problem here. The first (embedding) layer of a modern transformer is still very similar to this, with the addition of a position encoding. The choice of what counts as a "simple feature" is admittedly somewhat arbitrary; we believe that considering word types as simple features, as we do in most of our analysis, is uncontroversial, but there are other more complex features which one still might want to control for in competency problems.

must control for all sources of extra-linguistic information, ensuring that no simple features contain information about class labels.

For some language understanding problems, such as natural language inference, this intuition is already widely held. We find it surprising and problematic when the presence of the word *"cat"*, *"sleeping"* or even *"not"* in either the premise or the hypothesis gives a strong signal about an entailment decision (Gururangan et al., 2018; Poliak et al., 2018). Competency problems are broader than this, however. Consider the case of sentiment analysis. It is true that a movie review containing the word *"amazing"* is more likely than not to express positive sentiment about the movie. This is because of distributional effects in how humans choose to use phrases in movie reviews. If our goal is to build a system that can accurately classify the sentiment of movie reviews, this is a useful feature. But if our goal is instead to build a machine that understands how sentiment is expressed in language, this feature is a red herring that must be controlled for to truly test linguistic competence.

3 Biased Sampling

To get machines to perform well on competency problems, we need data that accurately reflects the competency assumption, both to evaluate systems and (presumably) to train them. However, humans suffer from blind spots, social bias, priming, and other psychological effects that make collecting data for competency problems challenging. Examples of these effects include instructions in a crowdsourcing task that prime workers to use particular language,[3] or distributional effects in source material, such as the *"amazing"* examples above, or racial bias in face recognition (Buolamwini and Gebru, 2018) and abusive language detection datasets (Davidson et al., 2019; Sap et al., 2019).

In order to formally analyze the impact of human bias on collecting data for competency problems, we need a plausible model of this bias. We represent bias as rejection sampling from the target competency distribution based on single feature values. Specifically, we assume the following dataset collection procedure. First, a person samples an instance from an unbiased distribution $p_u(\mathbf{x}, y)$ where the competency assumption holds. The person examines this instance, and if feature $x_i = 1$ appears with label $y = 0$, the person rejects the instance and samples a new one, with probability r_i. If $y = 0$ corresponds to negative sentiment and x_i is the word *"amazing"*, a high value for r_i would lead to *"amazing"* appearing more often with positive sentiment, as is observed in typical sentiment analysis datasets.

We do not claim rejection sampling is a plausible psychological model of dataset construction. However, we *do* think it is a reasonable first-order approximation of the *outcome* of human bias on data creation, for a broad class of biases that have empirically been found in existing datasets, and it is relatively easy to analyze.

3.1 Emergence of Artifacts Under Rejection Sampling

Let $p_u(y|x_i)$ be the conditional probability of $y = 1$ given $x_i = 1$ under the unbiased distribution, $p_b(y|x_i)$ be the same probability under the biased distribution, and $\hat{p}(y|x_i)$ denote the empirical probability within a biased dataset of n samples. Additionally, let f_i be the marginal probability $p_u(x_i)$. Recall that $p_u(y|x_i)$ is 0.5 by assumption.

We will say that dimension i has an *artifact* if the empirical probability $\hat{p}(y|x_i)$ statistically differs from 0.5. In this section, we will show that an artifact emerges if there is a bias at dimension i in the sampling procedure, which is inevitable for some features in practice. We will formalize this bias in terms of a rejection sampling probability r_i.

For a single sample $\mathbf{x}, y$, we first derive the joint and marginal probabilities $p_b(y, x_i)$ and $p_b(x_i)$, from which we can obtain $p_b(y|x_i)$. These formulas use a recurrence relation obtained from the rejection sampling procedure.

$$p_b(y, x_i) = \frac{1}{2}f_i + \frac{1}{2}f_i r_i p_b(y, x_i)$$

$$\therefore p_b(y, x_i) = \frac{f_i}{2 - f_i r_i}$$

$$p_b(x_i) = \frac{1}{2}f_i + \frac{1}{2}f_i(1 - r_i) + \frac{1}{2}f_i r_i p_b(x_i)$$

$$\therefore p_b(x_i) = \frac{2f_i - f_i r_i}{2 - f_i r_i}$$

$$\therefore p_b(y \mid x_i) = \frac{p_b(y, x_i)}{p_b(x_i)} = \frac{1}{2 - r_i}$$

With no bias ($r_i = 0$), this probability is 0.5, as expected, and it rises to 1 as r_i increases to 1.

We define $\hat{p}(y|x_i)$ as the empirical expectation of $p_b(y|x_i)$ over n samples containing x_i, with dif-

[3] This is ubiquitous in crowdsourcing; see, e.g., common patterns in DROP (Dua et al., 2019) or ROPES (Lin et al., 2019) that ultimately derive from annotator instructions.

ferent samples indexed by superscript j. $\hat{p}(y|x_i) = \frac{1}{n}\sum_{j=1}^{n} y^j$. Note that $\hat{p}$ is a conditional binomial random variable. By the central limit theorem, $\hat{p}$ is approximately $\sim \mathcal{N}(\mu_{\hat{p}}, \sigma^2_{\hat{p}})$ for large n, where

$$\mu_{\hat{p}} = p_b(y \mid x_i) = \frac{1}{2 - r_i}$$

$$\sigma^2_{\hat{p}} = \left(\frac{1 - r_i}{(2 - r_i)^2}\right)^2 \cdot \frac{1}{n}.$$

This variance is inversely proportional to the number of samples n. Thus, $\hat{p}(y|x_i)$ can be well approximated by its expected value for a large number of samples. As the rejection probability r_i increases, the center of this distribution tends from 0.5 to 1. This formalizes the idea that bias in the sampling procedure will cause the empirical probability $\hat{p}(y|x_i)$ to deviate from 0.5, even if the "true" probability is 0.5 by assumption. Increasing the sample size n concentrates the distribution inversely proportional to $\sqrt{n}$, but the expected value is unchanged. Thus, artifacts created by rejection sampling will not be combated by simply sampling more data from the same biased procedure—the empirical probability will still be biased by r_i even if n increases arbitrarily. These persistent artifacts can be exploited at *i.i.d.* test time to achieve high performance, but will necessarily fail if the learner is evaluated under the competency setting.

3.2 Hypothesis Test

Here we set up a hypothesis test to evaluate if there is enough evidence to reject the hypothesis that r_i is 0, i.e., that the data is unbiased. In this case, we can use a one-sided binomial proportion hypothesis test, as our rejection sampling can only lead to binomial proportions for $p_b(y \mid x_i)$ that are greater than $\frac{1}{2}$. Our null hypothesis is that the binomial proportion $p_b(y \mid x_i) = 0.5 = p_0$, or equivalently, that $r_i = 0$. Our alternative hypothesis is that $p_b(y \mid x_i) \geq 0.5$. Let $\hat{p}$ be the observed probability. We can compute a z-statistic using the standard formula:

$$z^* = \frac{\hat{p} - p_0}{\sqrt{p_0(1 - p_0)/n}} \tag{1}$$

Thus, if our observed proportion $\hat{p}$ is far from $p_0 = 0.5$, we will have enough evidence to reject the null hypothesis that $r_i = 0$. This depends on n as well, and to explore this interaction, we solve for $\hat{p}$ for a given n and confidence level z^*: $\hat{p} = \frac{z^*}{2\sqrt{n}} + \frac{1}{2}$.

4 Empirical Analysis

With a hypothesis test in hand, we can examine existing datasets for evidence of statistically-significant feature bias, and then explore the extent to which this bias impacts models supervised with this data. Prior work has used pointwise mutual information (PMI) to find features that have high correlation with labels (e.g., Gururangan et al., 2018). This measure is useful for understanding why certain features might get used as deterministic decision rules by models (Ribeiro et al., 2018; Wallace et al., 2019). However, studies involving PMI have also intuitively understood that PMI by itself does not tell the whole story, as a strict ranking by PMI would return features that only appear once in the dataset. To account for this problem, they used arbitrary cutoffs and included information about feature occurrence in addition to their PMI ranking. A benefit of our approach to defining and detecting artifacts is that we have a single statistical test that takes into account both the number of times a feature appears and how correlated it is with a single label. We use this test to find features with the strongest statistical evidence for artifacts (§4.1) and then show empirically that models use these features inappropriately when making predictions (§4.2). This analysis goes beyond deterministic prediction rules, showing that the impact of sampling bias on model behavior is subtle and pervasive.

4.1 Data Analysis

We analyze two datasets with the hypothesis test from §3.2: SNLI (Bowman et al., 2015) and the Universal Dependencies English Web Treebank (Silveira et al., 2014).

SNLI Each feature x_i represents the presence of a word in a given example, counting each appearance in an instance as a separate occurrence[4] for the purposes of computing n and $\hat{p}$ in Equation 1. We compute a z-statistic for every token that appears in the SNLI data, where $p_0 = \frac{1}{3}$, as SNLI has three labels. We then plot the z-statistic for each token against the number of times the token appears in the data. We also plot a curve for the value of the z-statistic at which the null hypothesis (that $r_i = 0$) should be rejected, using a significance level of $\alpha = 0.01$ and a conservative Bonferroni correction (Bonferroni, 1936) for all 28,000 vocabulary items. This analysis is shown in Figure 1.

[4] We remove punctuation and tokenize on whitespace only.

We label in Figure 1 several words that were also found to be artifacts by Gururangan et al. (2018) and Wallace et al. (2019), among others.

We find a very large number of deviations from the competency assumption, many more than would be suggested by a PMI-based analysis. PMI equals $\log \frac{\hat{p}(y|x_i)}{\hat{p}(y)}$; because $\hat{p}(y)$ does not vary across features, and the data is balanced over labels, a PMI analysis ranks features by $\hat{p}(y|x_i)$, looking only at the y-axis in Figure 1. But the threshold for which a deviation in $\hat{p}(y|x_i)$ becomes a statistical artifact depends on the number of times the feature is seen, so our statistical test gives a simpler and more complete picture of data artifacts. Strong statistical deviations with less extreme PMI values still impact model behavior (§4.2 and §6).

UD English Web Treebank Next we turn to dependency parsing. In particular, we focus on the classic problem of prepositional phrase (PP) attachment (Collins and Brooks, 1995), which involves determining whether a PP attaches to a verb (e.g., *We ate spaghetti with forks*) or a noun (e.g., *We ate spaghetti with meatballs*). We heuristically extract (verb, noun, prepositional phrase) constructions with ambiguous attachment from the UD English Web Treebank (EWT) training data.[5] We treat (verb, preposition) tuples as features and attachment types (*noun* or *verb*) as labels, and we compute a z-statistic for each tuple.

Figure 2 shows the z-statistic for each tuple that appears 10 or more times in the data. We labeled tuples that also appear in the locally edited samples from the UD English contrast set created by Gardner et al. (2020). Many of these tuples fall either above or close to the significance curve, suggesting that the low contrast consistency reported by Gardner et al. (2020) could potentially be explained by models' reliance on these artifacts.[6]

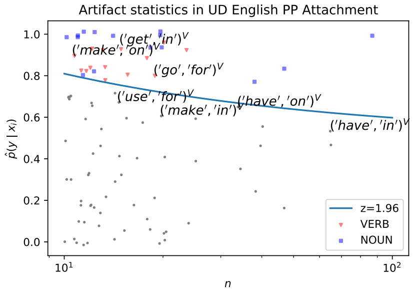

Figure 2: Artifact statistics of (verb, preposition) tuples in PP attachment in the UD English Web Treebank (EWT). Plotted are tuples that appear in the original EWT training data, and labeled are tuples that also appear in the UD English locally edited contrast set.

4.2 Model Analysis

The previous section reveals a large number of individual word dataset artifacts in the SNLI dataset. Here, we ask whether typical NLP models learn to bias their predictions based on these artifacts for both the SNLI and RTE (Dagan et al., 2005)[7] datasets. That is, we will show that these single words noticeably influence a model's confidence in particular predictions, even when the PMI value is not extreme enough to create a universal trigger (Wallace et al., 2019). Importantly, this analysis focuses on words with high z-statistics, which are often words that show up very frequently with slight deviations from $p_u(y|x_i)$. This includes words such as *"for"* and *"to"* (the two words with highest z-statistic for the neutral class), and *"there"* and *"near"* (the highest and fifth-highest z-statistic for the entailment class).

To measure the model bias learned from these words, we employ RoBERTa-base (Liu et al., 2019) fine-tuned on RTE, and ALBERT-base (Lan et al., 2020) fine-tuned on SNLI.[8] Given a single type such as $x_i =$ *"nobody"* and a target class such as $y =$ "contradiction", we estimate the model $\tilde{p}(y|x_i)$ as follows. We first create two synthetic input examples, one with the premise containing only the single token with an empty hypothesis, and one with an empty premise and hypothesis containing the single token. As each input contains only a single token without additional context, this tests whether the model will bias its output based on the token. We run a forward pass with each input and average the target class probabilities as an estimate of $\tilde{p}(y|x_i)$. All of the words in each dataset appearing at least 20 times are partitioned among the classes based on their largest class conditional z^*, and for each class we form two cohorts of 50 words each with the highest and lowest z^*. Let $\mathcal{X}_{y,z^*_<}$ denote the set of x_i with the lowest z^* for class y and similarly let $\mathcal{X}_{y,z^*_>}$ denote the set with

[5] See Appendix A for how we extract these constructions.

[6] Some tuples in the plot with high $p(y|x_i)$ are not artifacts, as the attachment decision is nearly deterministic. For instance, the top right blue dot corresponds to (*have, of*); *of* can only attach to *have* in archaic or idiosyncratic constructions.

[7] We use the RTE data from SuperGlue (Wang et al., 2019).

[8] Both models are from Morris et al. (2020), and are implemented in the Transformers library (Wolf et al., 2020).

Dataset	Class	$\Delta\hat{p}_y$
RTE	entailment	+2.2 %
SNLI	entailment	+14.7 %
SNLI	neutral	+7.9 %
SNLI	contradiction	+12.5 %

Table 1: Comparison of the average model predicted class probability for a single token across two cohorts of large z^* tokens and low z^* tokens using RoBERTa (RTE) and ALBERT (SNLI).

the largest z^*. Finally, we compute the average $\Delta\tilde{p}_y = \sum_{x_i \in \mathcal{X}_{y,z^*_>}} \tilde{p}(y|x_i) - \sum_{x_i \in \mathcal{X}_{y,z^*_<}} \tilde{p}(y|x_i)$.

The results are shown in Table 1. As can be seen, these models exhibit significant bias based on the single token inputs, with $\Delta\tilde{p}_y$ exceeding 10% for some classes. The bias is much more extreme for SNLI versus RTE, likely due to the fact that RTE has two orders of magnitude less data than SNLI.

5 Mitigating Artifacts with Local Edits

Many works have tried to remove data artifacts by making minimal changes to existing data (Shekhar et al., 2017; Sennrich, 2017; Zhao et al., 2018, *inter alia*). In this section we show that this kind of data augmentation can be effective with an appropriately sensitive edit model, where *sensitivity* refers to how often a change to inputs results in the label changing. However, because humans are involved in making these changes, achieving appropriate sensitivity is challenging, and bias in this process can lead to the introduction of new artifacts. This suggests that care must be taken when performing edit-based data augmentation, as large edited training datasets are not likely to be artifact-free (cf. Tafjord et al., 2019; Huang et al., 2020).

Imagine a new dataset $\mathcal{D}_e$ consisting of samples $\mathbf{x}', y'$ generated by making local edits according to the following repeated procedure:

1. Randomly sample an instance $\mathbf{x}$ from a dataset $\mathcal{D}_b$ of n instances created under p_b.
2. Make some changes to $\mathbf{x}$ to arrive at $\mathbf{x}'$.
3. Manually label y' and add $\langle \mathbf{x}', y' \rangle$ to $\mathcal{D}_e$.

We examine the expected probability $p_e(y'|x'_i)$ under this edit process. To derive this probability, we will need to know the probability that a change to $\mathbf{x}$ changes y. We define the *edit sensitivity* s to be this probability, i.e., $s = p_b(y' = \neg y)$. The other quantity of interest for an edit model is e_i, the probability that dimension i gets flipped when going from $\mathbf{x}$ to $\mathbf{x}'$. We now show theoretically that s and e_i control whether samples generated by local editing debias first-order artifacts.

Proposition 1 (Proof in §B)**.** *If* $r_i \neq 0$*, then,*

$$\frac{1+e_i}{s} = 2 \iff p_e(y' \mid x'_i) = \frac{1}{2}.$$

An analogous statement holds when $=$ *is replaced by* $\leq$ *on both sides.*

Thus, local editing will dilute or cancel out artifacts from rejection sampling if the edit sensitivity is high enough, but if it is too high, it can introduce artifacts in the opposite direction. In practice, if it is possible to pick a dimension to flip more or less uniformly, then $e_i \approx 0$ in a high-dimensional space, so engineering $s \approx 0.5$ should produce artifact-free additional samples with $p_e(y'|x'_i) \approx 0.5$.

Furthermore, edit sensitivity and edit dimension are empirically measurable when constructing a dataset. This empirical measurement can give theoretical guarantees for the degree to which the local editing will alleviate artifacts (i.e., this gives us a principled way to decide between edit models).

5.1 Local Edits in Pracitce

We empirically investigate the effectiveness of local edits for reducing single feature artifacts using locally edited samples generated from two datasets: (1) the Boolean Questions dataset (BoolQ; Clark et al., 2019), which consists of pairs of paragraphs (p) and questions (q), where each q has a binary answer that can be found by reasoning over p; and (2) IMDb (Maas et al., 2011), a sentiment classification dataset in the domain of movie reviews. We define each feature x_i as the occurrence of a particular word within q for BoolQ, and within the text of the review for IMDb. Gardner et al. (2020) generated additional data for BoolQ and IMDb by making local edits to the question or review text and recording the updated binary label.

Figure 3 visualizes the effect of these changes on single-feature artifacts by comparing the artifact statistics for the original texts to the statistics for the edited texts generated by Gardner et al. (2020). For BoolQ, many tokens in the original data exhibit artifacts in the positive (> 0.5) direction, while, within the edited data, almost all tokens fall within the confidence region. In contrast, there is no apparent distributional difference between artifact statistics for the original vs. edited texts on IMDb. We find that for BoolQ the values $(1 + e_i)/s$ tightly concentrate around a mean of 1.94, which, by Proposition 1, explains why most of the $\hat{p}(y|x_i)$ values for the

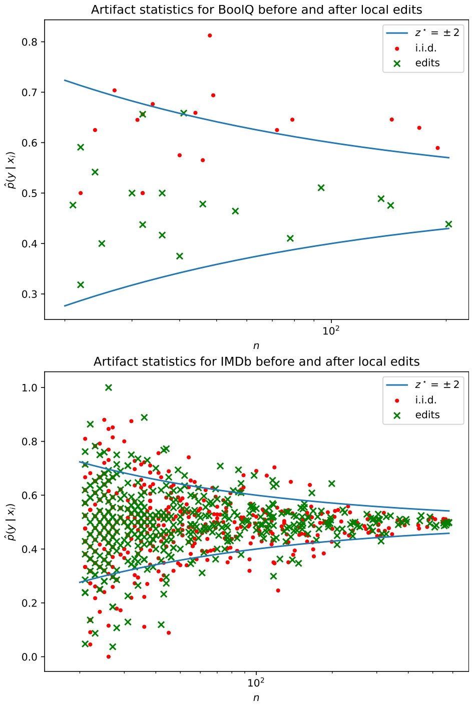

Figure 3: The artifact statistics of the original BoolQ (above) and IMDb (below) samples are plotted in red, compared to the artifact statistics over the edited instances, plotted in green.

edited samples are not significantly different from 0.5. For IMDb, $(1 + e_i)/s$ was close to 1. This case study illustrates the importance of leveraging our theory to engineer better edit models.

5.2 Local Edits and Boolean Sensitivity

In the above discussion we used the term *sensitivity* in an informal way to describe the probability that a local edit changes the label. This term also has a related formal definition in the study of boolean functions, where it is an implicit complexity measure (Wegener, 1987). Sensitivity in this sense has been shown to correlate with generalization in neural networks (Franco, 2001), and has been extended for use with practical NLP datasets (Hahn et al., 2021). In this section we discuss the intersection of our theory with sensitivity analysis, highlighting limitations in sensitivity analysis for sampled datasets that could be addressed in future work.

For a boolean vector x, let x^i be the *Hamming neighbor* of x at dimension i: i.e., the vector where x_i has been flipped and all other bits remain the same. Consider $f : \{0,1\}^d \to \{0,1\}$. The *sensitivity set* $S(f, x)$ is the set of Hamming neighbors of x with different labels:

$$S(f,x) = \left\{ i \in [0, \ldots, d] \mid f(x) \neq f(x^i) \right\}.$$

The *local sensitivity* $s(f, x)$ is the size of this set: $s(f, x) = |S(f, x)|$. Finally, the *global sensitivity* is defined $s(f) = \max_x s(f, x)$.

Significance of sensitivity In our case, the effect of local editing on a dataset can be understood in terms of sensitivity. Imagine a boolean function $f : \{0,1\}^d \to \{0,1\}$ from which we draw n samples $\langle \mathbf{x}, y \rangle$. If these samples are drawn uniformly over $\{0,1\}^d$, then the probability of observing any Hamming neighbors goes to 0 rapidly with d. Thus, it is possible to pick a low sensitivity function that can perfectly fit the data. In this sense, the true sensitivity of f is likely *underspecified* by the dataset.

Imagine we give this data to a learner with inductive bias resembling some variant of Occam's razor. If the learner's notion of complexity is correlated with sensitivity (which many complexity measures are), then the learner will favor low sensitivity decision boundaries. Thus, the fact that sensitivity is underspecified in the training data is a problem if the gold-standard function has high sensitivity, as the inductive bias of the learning algorithm may favor low-sensitivity alternatives.

Contrast this with a dataset where some local neighborhoods in the input space have been filled in with local edits. The set of observed neighbors around a point x provide a lower bound on $s(f, x)$, which is a lower bound on $s(f)$. In this sense, $s(f)$ is no longer underspecified by the dataset.

In this discussion we have used *underspecified* in an informal way; there is no precise measure of the sensitivity of a sampled dataset (as opposed to a fully-specified function), particularly when generalizing from finite boolean functions to natural language inputs. Attempts to generalize sensitivity to natural language have done so by leveraging large language models to generate neighbors from which sensitivity can be estimated (Hahn et al., 2021). Resampling data in this way can give reasonable estimates of the sensitivity of the underlying task, but it is fundamentally incompatible with measuring dataset artifacts of the kind we discuss in this paper, as the generative model can fill in parts of the data distribution that are missing due to sampling bias, giving a higher estimate of sensitivity than is warranted by the sampled dataset.

6 Data Filtering

A recent trend is to remove data from a training set that is biased in some way in order to get a model that generalizes better (Bras et al., 2020; Swayamdipta et al., 2020; Oren et al., 2020). While this method can be effective for very biased datasets, it is somewhat unsatisfying to remove entire instances because of bias in a single feature. In the extreme case where $r_i \approx 1$, such as with *"nobody"* in SNLI (Fig. 1), this process could effectively remove x_i from the observed feature space.

To understand the effect of these automated methods on dataset artifacts, we repeat the analysis from §4.1 on data that was classified as "ambiguous" according to Dataset Cartography (Swayamdipta et al., 2020). This data was shown to provide better generalization when used as training data compared to the original training set. The ambiguous instances did not have a balanced label distribution, so we downsampled the data to balance it, then downsampled the whole training data to get the same number of instances as the balanced ambiguous set.

The resulting artifact plots are shown in Figure 4. As can be seen, the "ambiguous" instances have many fewer deviations from the competency assumption, across the entire range of our hypothesis test. It is not just high PMI values that are getting corrected by finding ambiguous instances; all statistical deviations are impacted. This effect is striking, and it further corroborates our arguments about the importance of the competency assumption.

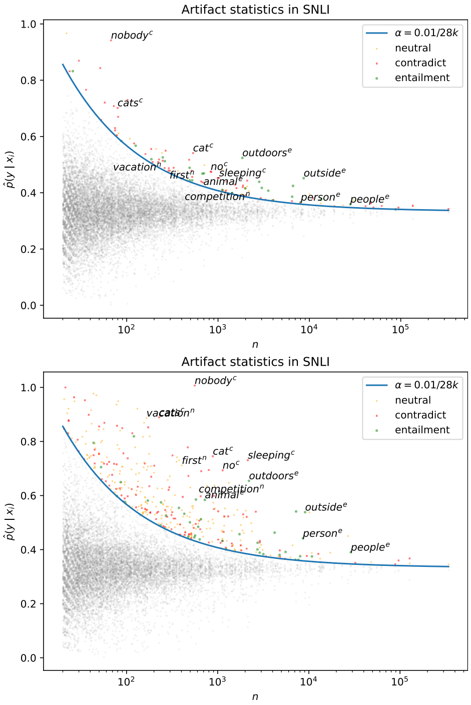

Figure 4: Statistical artifacts in ambiguous instances (Swayamdipta et al., 2020; above) versus a random (same-size) sample from the SNLI training set (below). The filtering done by ambiguous instance detection targets statistical artifacts across the whole range of the statistical test, not just high PMI values.

7 Other Related Work

Theoretical analysis of bias Several recent works explore sources and theoretical treatments of bias or spurious correlations in NLP (Shah et al., 2020a; Kaushik et al., 2020) or ML more broadly (Shah et al., 2020b). Our work differs by introducing a competency assumption and exploring its implications. The difference between our biased and unbiased distributions is an instance of covariate shift (Quionero-Candela et al., 2009).

Competent models An interesting question is whether we can inject a "competency inductive bias" into models, i.e., discourage relying on individual features. The closest works we are aware of are methods that ensemble weak models together with strong models during training (Clark et al., 2020; Dagaev et al., 2021), or ensembles of models with unaligned gradients (Teney et al., 2021).

8 Conclusion

The more NLP models advance, the better they are at learning statistical patterns in datasets. This is problematic for language understanding research if some statistical patterns allow a model to bypass linguistic competence. We have formalized this intuition with a class of problems called *competency problems*, arguing that, for any language understanding task, *all* correlations between simple features and labels are spurious. Collecting data meeting this assumption is challenging, but we have provided theoretical analysis that can inform future data collection efforts for such tasks.

We conclude with some final thoughts on general best practices for data collection, informed by the analysis in this paper. If annotators are generating text for some data collection task, find ways to decrease priming effects. This could involve using images as prompts instead of text (Novikova et al., 2017; Weller et al., 2020), or randomly sampling words to include in the generated text. If existing text is being collected and annotated, make local edits to the text while monitoring the sensitivity of those edits according to the guidelines in §5, perhaps using difference processes between train and test, to minimize correlations between train features and test labels.

References

C. E. Bonferroni. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

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.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *ICML*.

Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAT*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics.

Michael Collins and James Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Third Workshop on Very Large Corpora*.

Marta R. Costa-jussà, Christian Hardmeier, Will Radford, and Kellie Webster, editors. 2019. *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*. Association for Computational Linguistics, Florence, Italy.

Nikolay Dagaev, Brett D. Roads, Xiaoliang Luo, Daniel N. Barry, Kaustubh R. Patil, and Bradley C. Love. 2021. A too-good-to-be-true prior to reduce shortcut reliance.

Ido Dagan, Oren Glickman, and B. Magnini. 2005. The pascal recognising textual entailment challenge. In *MLCW*.

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.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Leonardo Franco. 2001. A measure for the complexity of boolean functions related to their implementation in neural networks. *arXiv preprint cond-mat/0111169*.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

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 Conference 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.

Michael Hahn, Dan Jurafsky, and Richard Futrell. 2021. Sensitivity as a complexity measure for sequence classification tasks. *Transactions of the Association for Computational Linguistics*.

William Huang, Haokun Liu, and Samuel R. Bowman. 2020. Counterfactually-augmented SNLI training data does not yield better generalization than unaugmented data. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 82–87, Online. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Divyansh Kaushik, Amrith Rajagopal Setlur, E. Hovy, and Zachary Chase Lipton. 2020. Explaining the efficacy of counterfactually-augmented data. *ArXiv*, abs/2010.02114.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Ryan O'Donnell. 2014. *Analysis of boolean functions*. Cambridge University Press.

Inbar Oren, Jonathan Herzig, Nitish Gupta, Matt Gardner, and Jonathan Berant. 2020. Improving compositional generalization in semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2482–2495, Online. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and N. Lawrence. 2009. *Dataset Shift in Machine Learning*. MIT Press.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI*.

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 *ACL*.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? Assessing MT quality with contrastive translation pairs.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020a. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online. Association for Computational Linguistics.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. 2020b. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9573–9585. Curran Associates, Inc.

Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. 2017. Foil it! Find One mismatch between image and language caption.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pages 9275–9293, Online. Association for Computational Linguistics.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. QuaRTz: An open-domain dataset of qualitative relationship questions. 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 5941–5946, Hong Kong, China. Association for Computational Linguistics.

Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel. 2021. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Ingo Wegener. 1987. *The complexity of Boolean functions*. Wiley.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. 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.

Head: ***Noun***	Head: ***Verb***
I think 2012 is going to be a great year for Fujairah as we have A LOT of projects to be done by 2012. V: *going* NP: *year* PP: *for Fujairah*	Went to the Willow Lounge this past weekend for dinner and drinks ... place is awesome. V: *Went* NP: *weekend* PP: *for dinner and drinks*

Table 2: Examples of constructions extracted from the UD EWT training data using the heuristics described in Appendix A. Shown are sentences with differing PP attachment types for tuple (*go, for*).

A Ambiguous PP Attachment Extraction

Here, we describe how we heuristically extract (verb, noun, prepositional phrase) constructions with ambiguous attachment from the UD English Web Treebank training data (Section 4.1). Examples of such constructions are shown in Table 2. We extract (V, N, PP) constructions from UD EWT inputs that meet the following criteria, which operate over the dependency relation annotations:

1. V, NP, and PP are contained in same sentence
2. *Either* PP depends on NP *or* PP depends on V
3. NP depends on V and is not subject of V
4. PP follows both V and NP in the sentence

B Proof of Proposition 1

Proposition 1. *If $r_i \neq 0$, then,*

$$\frac{1+e_i}{s} = 2 \iff p_e(y' \mid x_i') = \frac{1}{2}.$$

Proof. First consider the case where $y' = \neg y$. We marginalize the probability as follows:

$$\begin{aligned} p_e(y' \mid x_i') &= p_b(\neg y \mid x_i)(1-e_i) + p_b(\neg y \mid \neg x_i)e_i \\ &= \frac{2-2r_i+r_ie_i}{2(2-r_i)}. \end{aligned}$$

We now consider the other case where $y' = y$.

$$\begin{aligned} p_e(y' \mid x_i') &= p_b(y \mid x_i)(1-e_i) + p_b(y \mid \neg x_i)e_i \\ &= \frac{2-r_ie_i}{2(2-r_i)}. \end{aligned}$$

Finally we marginalize over whether $y' = \neg y$, an event which has probability s.

$$\begin{aligned} p_e(y' \mid x_i') &= \frac{(2-2r_i+r_ie_i)s + (2-r_ie_i)(1-s)}{2(2-r_i)} \\ &= \frac{2-(2s-e_i)r_i}{2(2-r_i)}. \end{aligned}$$

At this point, we solve for when $p_e(y'|x_i') = 0.5$. Here we use the assumption that $r_i \neq 0$.

$$\begin{aligned} \frac{1}{2} &= \frac{2-(2s-e_i)r_i}{2(2-r_i)} \\ 2-r_i &= 2-(2s-e_i)r_i \\ r_i &= (2s-e_i)r_i \\ \therefore s &= \frac{1+e_i}{2}. \end{aligned}$$

□

The proof generalizes to the case where $p_e(y' \mid x_i') \leq 0.5$, giving us the analogous result where $=$ is replaced by $\leq$ in the statement of Proposition 1.