Improving Classifier Robustness through Active Generation of Pairwise Counterfactuals

Ananth Balashankar[*] Google
Xuezhi Wang Google
Yao Qin Google
Ben Packer Google
Nithum Thain Google
Jilin Chen Google
Ed H. Chi Google
Alex Beutel[†] OpenAI

Abstract

Counterfactual Data Augmentation (CDA) is a commonly used technique for improving robustness in natural language classifiers. However, one fundamental challenge is how to discover meaningful counterfactuals and efficiently label them, with minimal human labeling cost. Most existing methods either completely rely on human-annotated labels, an expensive process which limits the scale of counterfactual data, or implicitly assume label invariance, which may mislead the model with incorrect labels. In this paper, we present a novel framework that utilizes counterfactual generative models to generate a large number of diverse counterfactuals by actively sampling from regions of uncertainty, and then automatically label them with a learned pairwise classifier. Our key insight is that we can more correctly label the generated counterfactuals by training a pairwise classifier that interpolates the relationship between the original example and the counterfactual. We demonstrate that with a small amount of human-annotated counterfactual data (10%), we can generate a counterfactual augmentation dataset with learned labels, that provides an 18-20% improvement in robustness and a 14-21% reduction in errors on 6 out-of-domain datasets, comparable to that of a fully human-annotated counterfactual dataset for both sentiment classification and question paraphrase tasks.

1 Introduction

Counterfactual data augmentation (CDA) has been used to make models robust to distribution shift and mitigate biases towards spuriously correlated attributes. Often, counterfactuals are generated as labeled examples through pre-specified templates [1, 2] or crowd-sourcing [3]. While natural text templates codify a specific number of assumptions of how counterfactual sentences and labels might vary, crowd-sourcing that can cover various types of counterfactuals, can be expensive. On the other hand, many existing methods [4, 5, 6, 7] simply rely on a label-invariance assumption: the label of the generated counterfactual example is the same as the corresponding original example. However, this simple label-invariance assumption does not always hold [8, 9, 10] and thus greatly increases the risk of using incorrect labels for counterfactuals during training. For example, for many NLP tasks a small perturbation can easily change the ground-truth label [3, 11], e.g., changing the input from *This movie is great* to *This movie is supposed to be great* for sentiment classification, or changing the hypothesis from *The lady has three children* to *The lady has many children* for natural language inference. Therefore, in this work, we mainly focus on addressing this challenging research problem:

"How can we automatically explore diverse counterfactual examples and learn their labels, given a counterfactual text generator?"

Beyond costly human annotation or simplifying assumptions of label invariance, researchers have explored to use a classifier f that has learnt to predict the label on the original dataset (X, Y). Such a classifier has been used to directly label generated examples (our "trust" baseline; [3]) or to weight generated examples based on the model uncertainty (our weighted-trust baseline; [12]). However, we see that using such simplistic labeling assumptions for counterfactual data augmentation have limited benefits for improving robustness (defined as the accuracy over a counterfactual test set of interest).

In this paper we propose an alternative approach to this problem: we leverage the sample efficiency of generative models and exploration capabilities of active learning [13] to 1) first generate a large number of diverse counterfactuals, and 2) then train an auxiliary classifier to automatically annotate the generated counterfactual data based on the difference between the original and counterfactual labels.

[*]Email: ananthbshankar@google.com
[†]work done while author was at Google

Findings of the Association for Computational Linguistics: EMNLP 2023, pages 127–139
December 6-10, 2023

Specifically, we propose to generate counterfactual examples that lie in the region of uncertainty of the classifier f, and learn a pairwise classifier h to predict the counterfactual label y'. The pipeline of our method is shown in Figure 1.

In particular, we utilize a very small set of human-annotated counterfactual examples to train the pairwise counterfactual classifier h, which takes in the pair of original and counterfactual sentences $(x, c_s(x))$ and the original label y as input. Then in the inference stage, the pairwise counterfactual classifier h is used to predict the labels to produce a large counterfactual augmentation dataset used to fine-tune f to improve robustness.

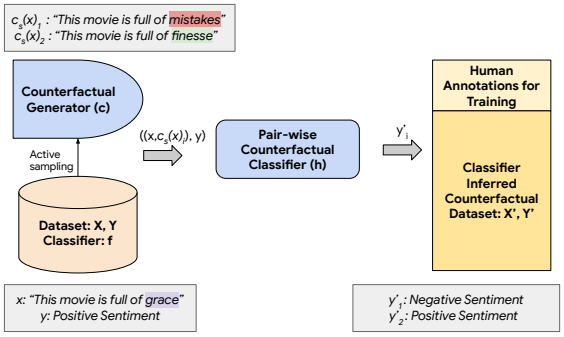

Figure 1: **Overview of proposed approach:** We propose to generate diverse counterfactuals through active sampling, and label them using a pairwise counterfactual classifier at scale. We use the labeled counterfactuals as data augmentation over the original classifier (not shown) to significantly improve robustness.

By using active sampling over counterfactual generators and auxiliary pairwise counterfactual classifiers, we show that we greatly reduce the number of counterfactual examples for which we need human annotation, while providing similar gains in robustness comparable to a fully human annotated counterfactual dataset. We attribute this to two core components of our method. First, the active-learning based sampling method helps diversify the types of generated counterfactuals and enables them to capture different robustness issues not previously captured by our classifier model. Second, the proposed auxiliary pair-wise classifier can automatically annotate the generated counterfactuals with more accurate labels and help efficiently scale up the size of the counterfactual augmentation dataset. Our core contributions in this work include:

- We propose an active-learning based sampling method to generate diverse counterfactuals and effectively improve robustness of classifiers for the sentiment classification task on Stanford Sentiment Treebank (SST-2) dataset, and the question paraphrase task on Quora Question Pair (QQP) dataset.

- We propose a novel pairwise counterfactual classifier to automatically label counterfactually generated examples based on a small set of annotated counterfactuals, improving sample efficiency of counterfactual data augmentation.

- The generated augmented dataset, which uses just 10% of human-annotated labels, produces an improvement in counterfactual robustness of 18-20%, comparable to a fully human annotated dataset, and a reduction in errors by 14-21% on out-of-domain datasets that were not used during training: IMDB, Amazon, SemEval (Twitter), and Yelp reviews.

2 Related Work

Our work is built on advances from various domains as outlined below:

Adversarial Text Generation Training against adversarial examples which perturb inputs in the vicinity of the existing training data by making geometric assumptions [10, 14] on a lower dimensionality of the data to improve robustness has been extensively studied recently. Natural examples which are syntactically and semantically similar to the original sentence, but produce different model predictions have been produced [7]. Similarly, defenses against adversarial attacks on self-attentive models have shown improvement in robustness to label invariant examples [15]. In FairGAN [4], they showed it is possible for a discriminator to achieve statistical parity on the real dataset, while performing the auxiliary task of detecting real and generated examples. Such controlled adversarial generative approaches [16] have demonstrated the effectiveness of automating data augmentation in text-based tasks. Generative models which optimize for fluency have passed human annotation checks where the model generated text is almost indistinguishable from human generated ones [17, 18]. We build on this body of work and utilize a generative model [19] that captures template-based counterfactuals to improve robustness. Generic adversar-

ial notions of robustness however applicable, fail to incorporate specific counterfactuals directly in their training and orthogonal to our scope of study. Through carefully disentangling specific attributes and the rest of the latent variables in text, we generate counterfactuals across all possibilities, and utilize human-annotated templates to label a small fraction of the generated examples to train a pairwise counterfactual classifier.

Semi-Supervised and Self-Supervised Learning Labeling functions which provide crude estimates of the label have been used in semi-supervised methods [20, 21, 22], and are further used to learn a generative model to generalize over them. Further, utilizing unlabeled data [23] to improve adversarial robustness leverages geometric smoothing-based techniques to bridge the sample complexity gap between accuracy and robustness [24]. Thus, semi-supervised learning approaches aim to generate examples where the discriminator is least confident about [12]. Language models with very large number of parameters have also shown to be few-shot learners with minimal supervision [25]. Similarly, reinforcement learning based approaches with minimal labels have been proposed to combine the objectives of accuracy and counterfactual robustness [26]. Generalization against counterfactual examples by making models not to rely on salient features (easy examples) have been extensively studied by modeling biases in corpora [27, 28, 29, 30, 31, 32]. While the goal in these works have been building ensembles or end-to-end bias mitigation models, our goal is to minimize the number of human labels required to achieve an equivalent improvement in robustness. In this spirit of efficiently capturing the patterns already prevalent in the original dataset, and learning only the new ones introduced in the counterfactual templates, we learn the pairwise counterfactual classifier on a small number of samples, and use it to capture the label variations in the remaining counterfactual dataset.

Counterfactual Applications The counterfactual datasets we use throughout this paper were intended to highlight the shortcomings of existing models at the time. Improving robustness through training on the augmented data has been extensively explored [33, 34]. Learning how counterfactuals differ have been explored by comparing against gradient supervision [35] and the generalizability between original and counterfactuals [3]. The generated counterfactuals have also been used for explanations [36], highlighting biases [1] and debiasing through statistical methods [37]. This rich set of contrast sets [11], checklists [8], paraphrases [38, 39], adversarial schemes [40] and lexical diagnostic datasets [41] form the foundation of our method, which re-purposes them to build a counterfactual generative model and improve counterfactual robustness.

Generative Learning Generative adversarial active learning has been proposed with pool-based and synthesizing-based sampling strategies [42]. While the pool-based strategy selects from an existing sample of generated examples, the synthesizing-based sampling re-samples from the generator based on information theoretic measures like mutual information [43, 44] or model uncertainty [45] or informativeness [46, 13] over the initial sample. Further, recent work has highlighted that using large language models for generation and annotation of that generated data can be very useful [47, 48]. We build on this work, and use the synthesizing-based Bayesian Generative Active Learning approach [44], where we condition on a counterfactual with low uncertainty in the model, to actively generate more counterfactual examples. We then iteratively sample the examples with the highest classifier uncertainty and annotate manually. The human annotations are then used to train the pairwise classifier h, which is then used to scale the annotation process for all other generated counterfactuals.

3 Methodology

Problem Framing

Let x, y be the input sentence and its associated label in the original dataset, respectively. We assume $y \in \{0, 1\}$ throughout the paper (i.e., we focus on binary classification tasks), but our framework can be extended to multi-class tasks as well.

Our core challenge is what is the true label y' for a generated counterfactual x'? Although we can further obtain human annotations, this can quickly become time consuming and budget intensive to do at scale. If we make the simplified assumption of label invariance throughout the counterfactual inputs x' generated, which is a common assumption in adversarial literature [49, 6, 7], we could end up with an incorrect counterfactual dataset which might hurt robustness and accuracy. Our goal is thus, to *generate a counterfactual augmentation dataset*

that produces a comparable improvement in accuracy and robustness as that of human-annotated counterfactuals with minimal supervision.

We frame this problem as how to learn when the labels flip, i.e., identifying when the label of the counterfactual is different from the label of the original sentence: $P(y \neq y') = \delta$, $(0 < \delta < 1)$, in the counterfactual distribution $x' \in X'$. Given a generation model c, we denote $c_s(x)$ as the generated counterfactual over x by changing an attribute s in x. We also assume that a classifier $f : X \rightarrow Y$ has been learnt on the original dataset (X, Y) by optimizing for accuracy A.

$$A = E_{(x,y)\in(X,Y)}\mathbb{I}(f(x) = y) \tag{1}$$

In our paper, the objective is to use the counterfactual data to train a model f' that improves robustness, i.e., to make sure the models we trained generalize to unseen scenarios. We measure this by the counterfactual accuracy $\tilde{A}$ of f on multiple held-out counterfactual datasets (X', Y') split based on domains (OOD), patterns (e.g. negation, insertion):

$$\tilde{A} = E_{(x',y')\in(X',Y')}\mathbb{I}(f'(x') = y') \tag{2}$$

In the remainder of this section, we first explain how the counterfactuals are generated using active learning. Then, we explain how we account for the possibility that the label of the generated counterfactual $c_s(x)$ might have flipped, using a pairwise classifier. Finally, we explain how when both these components are combined, we can further improve robustness.

Active Counterfactual Generation

To achieve the goal of improving counterfactual accuracy on held-out counterfactual datasets (Eqn 2), we use a controlled generative model to generate additional training counterfactual data $c_s(x) \in X'_t$ (here the subscript t denotes the training set) that modifies original input $x \in X$ based on the attribute s. In natural language tasks, the attribute s cannot be directly inferred from the sentence x and hence we rely on templates to define the types of counterfactual (e.g., negation, insertion, deletion) as commonly used in [8, 19] to infer the attribute s. Let $y \in Y, y' \in Y'_t$ be the label for the original and counterfactual sentences in our counterfactual training dataset. The training objective of robustness is to minimize the error $\tilde{\mathcal{E}}_t$ of the model f aggregated by attribute s on the training counterfactuals (X'_t, Y'_t), where CE refers to the cross-entropy loss, as follows:

$$\tilde{\mathcal{E}}_t(s) = E_{x\in X,(c_s(x),y')\in(X'_t,Y'_t)}CE(f(c_s(x)), y') \tag{3}$$

$$\tilde{\mathcal{E}}_t = E_{s\in S}\tilde{\mathcal{E}}_t(s) \tag{4}$$

The counterfactual generator that optimizes the above cross-entropy loss then generates several counterfactuals $c_s(x)$ by relying on instructions provided in controlled generation methods [19] such as "negation", "restructure". However, these counterfactuals are not necessarily diverse, and fails to incorporate the classifier's uncertainty to get the most informative set of generated counterfactuals. To improve generalization across a diverse set of counterfactual types, we fine-tune the generator to actively sample counterfactuals the most informative set from the unlabeled dataset $x* \in X'_t$ (BALD, [43]) that synthesizes examples by maximizing the acquisition function given by the Monte Carlo (MC) dropout approximation method [45] using the class-wise probability scores of the pairwise classifier h, where $H[y|x, .]$ represents the Shannon entropy of the corresponding conditional probability:

$$x^* = argmax_{x'\in X'_t}[H[y'|x', X, Y] - \mathbb{E}_{x\in X_t}H[y'|x', x, f(x)]] \tag{5}$$

Since y' is not readily available for counterfactual generated sentences $c_s(x)$ in our training dataset and gathering them for all examples can be expensive, our goal is to minimize the number of human-annotations of counterfactuals y' in the training dataset Y'_t, while achieving comparable improvement in robustness (Eqn 2). Hence, the training sentence and label set (X'_t, Y'_t) can be decomposed into two sets, one whose labels are human-annotated: (X'_a, Y'_a) and the other with model generated labels: (X'_g, Y'_g), such that $X'_t = X'_a \cup X'_g, Y'_t = Y'_a \cup Y'_g$. Our goal is to automatically discover informative counterfactuals X'_g and learn their labels with access to a limited human-annotated counterfactual data (X'_a, Y'_a), where $|Y'_a| \ll |Y'_g|$, while achieving counterfactual robustness $\tilde{A}$ (Eqn 2) comparable to the scenario when all the training labels are human-annotated.

Pairwise-Counterfactual (PC)

In order to generate labels for the counterfactuals, we construct a novel *auxiliary pairwise classifier* h,

which at inference time, takes in as input both the original dataset $(x, y) \in (X, Y)$, and a corresponding counterfactual $c_s(x) \in X'_g$, to output $y' \in Y'_g$. This classifier h is trained on *pairs* of input sentences $x, c_s(x)$ and the original label y to predict the human-annotated label $y' \in Y'_a$.

Specifically, the classifier h takes in the original input sentence x and its associated label y, as well as its corresponding counterfactual example $c_s(x)$. The output of the classifier $h(x, c_s(x), y)$ is the predicted label of the counterfactual example $c_s(x)$. In the training stage, the classifier h is optimized on the counterfactual examples with human-annotated labels $(c_s(x), y') \in (X'_a, Y'_a)$ via minimizing the loss function:

$$\ell_h = E_{\substack{(x,y)\in(X,Y) \\ (c_s(x),y')\in(X'_a,Y'_a)}} CE(h(x, c_s(x), y), y') \tag{6}$$

With the well-trained classifier h, we can generate the labels for any counterfactual example $c_s(x) \in X'_g$ (the counterfactual set without human annotation) according to:

$$y' = h(x, c_s(x), y) : (x, y) \in (X, Y), c_s(x) \in X'_g \tag{7}$$

Classifier-Aware Pairwise-Counterfactual (CAPC)

Additionally, since we know that f is already optimized to predict the label accurately on the original dataset, the auxiliary classifier h could potentially leverage f in its pairwise prediction through transfer learning. Specifically, if we decompose the counterfactual distribution (X', Y') as a mixture of samples from the original distribution (X, Y) and those that are independent of the original distribution, we would benefit by training h to identify samples from the latter distribution. In addition, assuming the correspondence between $f(x)$ and $f(c_s(x))$ is easier to learn (e.g., with a lower model complexity), we could also benefit from learning a classifier-aware function to better capture this correspondence. Thus, we propose to augment the predictions of the original classifier $f(x), f(c_s(x))$ as input to h as follows:

$$y' \in Y'_g = h(x, c_s(x), y, f(x), f(c_s(x))) : \quad (x, y) \in (X, Y), c_s(x) \in X'_g \tag{8}$$

Any uncertainty that f has on the counterfactual samples $P(f(c_s(x)) \neq y')$ can be mitigated by the auxiliary classifier h by identifying patterns in $c_s(x)$ when f predicts incorrectly. As a simple example, without any human annotation, the original model f might make incorrect assumptions on $c_s(x)$ that lead to incorrect predictions $f(c_s(x)) \neq y'$, e.g., a sentiment analysis model might give "positive" sentiment predictions due to the presence of qualifiers like "terrific", "amazing" (*this movie was amazing*) even when the counterfactual input $c_s(x)$ alters aspects of a sentence that changes the label (*this movie was supposed to be amazing*). But, this can be corrected using Eqn 8 after h has observed some data over the correct correlation between $x, c_s(x), y, f(x), f(c_s(x))$ and y', especially if there exists a lower-complexity function mapping between them - for instance, adding the phrase "supposed to be" may alter the label of a review.

4 Evaluation

We evaluate on two NLP tasks, sentiment classification and question paraphrase, using two datasets namely the Stanford Sentiment Treebank (SST-2) [50] and the Quora Question Pair (QQP) [51, 52]. When the CAPC classifier is used in conjunction with generated examples through active learning, we correspondingly prefix the model name as **p-CAPC** (pool-based sampling with no retraining as per Eqn 5) or **s-CAPC** (examples synthesized with re-generation of counterfactuals optimizing Eqn 5).

Counterfactual Generator: Polyjuice

We use a general purpose counterfactual text generator called Polyjuice [19], which extends CheckList [8], that has shown promise by improving diversity, fluency and grammatical correctness as evaluated by user studies. It covers a wide variety of commonly used counterfactual types including patterns of negation [3], adding or changing quantifiers [11], shuffle key phrases [38], word or phrase swaps which do not alter POS tags [40] or parse trees [39], along with insertions or deletion of constraints that do not alter the parse tree [41]. Specifically, we use 8 types of counterfactuals - negation, quantifier, lexical, resemantic, insert, delete, restructure, shuffle; in Polyjuice to generate the augmented dataset. Other text generative models like [5, 3, 6] that improve adversarial robustness or like [53, 54] that allow controlled generation could be used as well.

Experiment Setup

We test our methods on two popular text datasets. We briefly describe the two datasets below, and discuss the different evaluations of counterfactual robustness we perform over them.

Stanford Sentiment Treebank (SST-2): The sentiment analysis task in SST-2 [50] assigns a binary sentiment (negative/positive) to a sentence mined from RottenTomatoes movie reviews. The corresponding counterfactuals are generated using the Polyjuice generator [19]. The original dataset contained 4,000 samples, while the counterfactual dataset had 2,000 samples with human labels against which we evaluate. We show a sample of the dataset in the following:

Positive: A dog is embraced by the dog **Negative**: A dog is not embraced by the dog

Quora Question Pair: In the QQP dataset [51, 52], given a pair of questions, the task is to predict if they are semantically equivalent, hence marked as duplicate. Here, again the second question is modified by Polyjuice [19] as per the templates used for the SST-2 dataset including negation, insertion, deletion, rephrasing, etc, out of which 1,911 samples were human annotated for evaluation. The original dataset had 20,000 samples.

Duplicate: How can I help a friend experiencing serious depression?; How can I help a friend who is in depression? **Non-duplicate**: How can I help a friend experiencing serious depression?; How can I play with a friend who is in depression?

Evaluation: In both datasets, we have a small number of counterfactual human annotations available (SST-2: 2,000; QQP: 1,911) [19]. We divide these examples into two sets, one for training and annotating using h, and another held-out test dataset used to compute counterfactual robustness of f. The former dataset is used for fine-tuning f for counterfactual robustness, while the latter is used only as a held-out test set. In the SST-2 dataset, this means we split out 1,000 samples for training/annotation and 1,000 as the test set, while in the QQP dataset, we use 1,000 samples for training/annotation and the remaining 911 samples for testing counterfactual robustness. However, our aim is to use a minimal subset of the 1,000 samples available for training the base classifier directly. Instead, we use a smaller training dataset (100) to train our pairwise classifier which in-turn can then *artifically* annotate the remaining (say 900) samples. The combination of these (sum to 1000) will then be used to train the base classifier. Thus, in all our experiments, the number of counterfactual samples available to the base classifier to train on remains the same, although at different levels of human labeling costs.

The classifier f is first trained on the original classifier and then fine-tuned on the counterfactual dataset. We also perform 10 random initializations of the model f and h and a 10-fold cross-validation split on the training/annotation data, thus report the mean and standard error bounds $\sigma/\sqrt{n}$ over $n = 1000$ runs for each model-based annotation and training for counterfactual robustness. We used the standard hyperparameters provided[1] for training f on (X, Y) and the hyperparameters for fine-tuning f on (X'_t, Y'_t) include learning rate of $5e^{-5}$, batch size of 16 and a sequence length of 120 for 20 epochs. The pairwise counterfactual classifier's hyperparameters were chosen after a grid search to have a learning rate of $5e^{-4}$, batch size of 32 for 50 epochs, sequence length of 240 including the original label and classifier predictions with special marker characters. While the base classifier f is trained on contextual embeddings of the sentence(s), h is trained by further augmenting the original and counterfactual sentence embeddings as input to RoBERTa followed by the base classifier's predictions separated by special delimiters [DEL]. A similar 10-fold cross-validation split is used to finetune the parameters of the classifier h.

Out-of-Distribution (OOD): To test the methodology on out-of-domain datasets, we test on sentiment analysis tasks in 6 reviews datasets - IMDB movie (3 including contrast sets) reviews, Amazon, SemEval, and Yelp reviews [55]. The IMDB reviews (1,700) were collected by [3] through careful human elicitation to produce label varying counterfactuals of existing IMDB reviews. In the Yelp reviews [56], the task is to predict the ratings of 115,907 reviews on a scale of 1-5, and in the Amazon reviews [57], we evaluate on the 57,947 reviews in the clothing product category. Each of these datasets was not used for training either the base classifier or the pairwise classifier, and the training relies solely on the SST-2 dataset. So, we can measure the generalizability of the pairwise classifier based data augmentation methodology.

Baselines

We now briefly describe five different baselines used to generate the labels of counterfactual augmented data (Y'_g), given access to a small number of annotated labels Y'_a. **No-cda**: f without any counterfactual data used for robustness. **Label-invariant (invariant)** : the labels of the counterfactual examples are assumed to be the same as the original sentence: $y' = y$ (except for the counterfactuals generated for the negation type, where it is the opposite). **Trust**: we trust the classifier f to annotate the counterfactual labels $y' = f(c_s(x))$ - a form of semi-supervision based on the existing base classifier. **Weighted-trust** (**w-trust**): the label of the counterfactual example is computed via the maximum score weighted by the confidence score of the classifier f on the pair for a label l : $p_l(x)$ such that $y' = \arg\max_l p_l(x) \cdot p_l(c_s(x))$. **Random**: In order to understand the importance of the counterfactual sentences used in the pairwise classifier, we also evaluate against a classifier which takes two randomly paired sentences from the original dataset as input and predicts the second label given the label of one sentence. **Training**: we only use those counterfactual examples with human-annotated labels (X'_a, Y'_a) and drop all other counterfactual examples.

For all these baselines as well as our proposed methods, we use the RoBERTa [58] fine-tuned model as the choice of classifier f, and a corresponding pairwise fine-tuning task using RoBERTa [1] for the auxiliary pairwise counterfactual classifier h.

5 Results

Improving Counterfactual Robustness

To demonstrate the effectiveness of our proposed method: actively synthesized classifier-aware pairwise-counterfactual (**s-CAPC**), we perform counterfactual data augmentation using 10% counterfactual examples with human-annotated labels as well as 90% counterfactual examples (a total of 1,000 samples), whose labels are predicted using each method. The error rate on the hold-out counterfactual examples (referred as robustness) as well as on the original test set are shown in Figure 2.

We can clearly see that (1) the error rate of our proposed method: **s-CAPC** significantly outperforms other baselines on models' robustness.

[1] huggingface.co/roberta-large-mnli, textattack/roberta-base-SST-2, ji-xin/roberta_base-QQP-two_stage

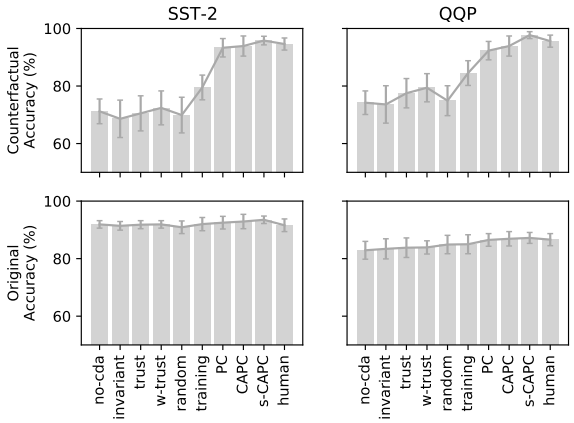

Figure 2: **(a) Robustness:** (first row) Training on 10% of human-annotated counterfactuals, and annotating the rest using the auxiliary classifier, we achieve a comparable improvement in robustness (lower error rate) for both Stanford Sentiment and Quora Question Pair datasets; **(b) Accuracy:** This improvement in robustness does not sacrifice the accuracy on the original held-out dataset.

(2) Comparing PC and CAPC, we can see that CAPC performs slightly better than PC. This indicates that the prediction of the original classifier $f(x), f(c_s(x))$ does provide additional information to help with labels prediction. (3) In addition, we also compare our methods with the extreme case that all the counterfactual examples (100%) are provided human-annotated labels, denoted as (**human-labels**). Surprisingly, our methods, which only use 10% human-annotated labels and predict the labels for the other 90% counterfactual data, achieve comparable performance in improving models' robustness. This sufficiently supports that our proposed methods can effectively predict the labels for counterfactual examples. (4) Looking at the error rate on the hold-out original test set, all the methods share a similar performance on SST-2 and our methods are better than other baselines and comparable to human-labels on QQP.

How much human-annotated data do we need?

To understand the impact of the training data provided to the auxiliary classifier h, we increased the % of data Y'_a provided to the classifier. While this increases costs of annotation, it is important to understand the headroom improvement in counterfactual robustness one would get had they opted for complete human-annotation. Figure 3 shows that across both datasets, the improvement in accuracy and robustness in providing more human

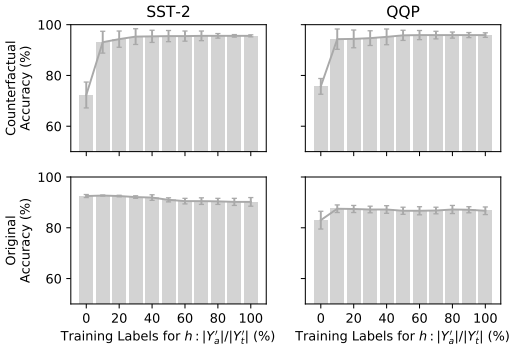

Figure 3: **Impact of training size:** As the number of samples $|Y'_a|$ increases more than 10% in the s-CAPC model , there is not much headroom in counterfactual accuracy, and does not significantly impact the accuracy on the held-out original test dataset on both SST-2 and QQP datasets (overlapping error bounds).

annotations to train h : CAPC and subsequently training the model f : RoBERTa-{SST-2, QQP} is not significant and hence further demonstrates that, with just 10% of the augmentation dataset, we can already achieve an improvement comparable to a fully human annotated dataset. This further confirms our method can achieve high *sample efficiency* in improving models' robustness.

Generalization across Counterfactual Types

We evaluate the generalization of our pairwise counterfactual classifier h by ablating one counterfactual type (e.g negation, quantifier, etc) at a time during training h, but still annotate them to generate the augmented training data for f. The results are shown in Table 1 (rows 2-9). We see that for the SST-2 task, our approach outperforms existing baselines on counterfactual robustness. This further indicates the importance of learning a counterfactual classifier which captures patterns of label invariance that generalizes across counterfactual templates. Finally, we evaluate if our generated augmentation dataset can be used to improve *unseen* counterfactual types - ablated while training both h and f. While this is not the goal of our paper, it is useful to understand what types of counterfactuals are captured by our generator and if any overlap between the types of counterfactuals is leveraged. Table 1 (row 10) shows that our approach is comparable with baselines (rows 2-5 in Table 1) when a specific counterfactual type is ablated completely from the data augmentation pipeline. This is consistent with existing work [59, 60] and further highlights the need to incorporate diverse types of counterfactuals to perform data augmentation.

Checklist Evaluation

To further validate that the generated labels by our auxiliary model can be used for other tasks, we evaluate it against the labels in CheckList [8] which capture other types of counterfactuals. We measure the *Absolute Failure Gap*: $|\epsilon - \epsilon_a|$ computed as the difference between the true error rate ϵ and the error rate as reported by using our augmented dataset ϵ_a while evaluating the models and tasks in the CheckList dataset. In Figure 4, we see that even when the training data provided to the auxiliary classifier is synthetically made explicitly label-invariant (90%), evaluating against counterfactuals with minimal label-invariance (10%), our model generalizes with a lower failure gap than other augmentation approaches. However, on the original Checklist dataset there is no significant improvement in failure gap compared to reporting the failure gap just on the training data alone.

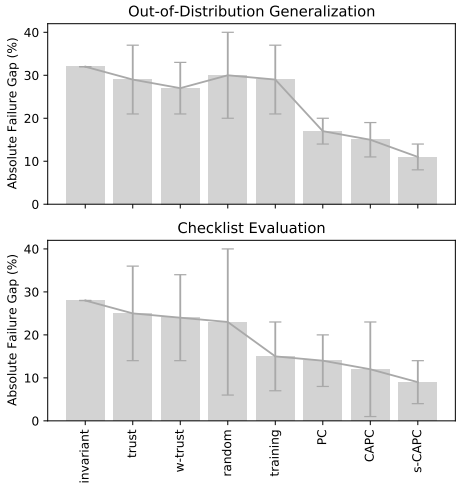

Figure 4: **Checklist Evaluation - (a) Out of distribution data**: Our methods perform well over different label-invariant distributions with 90% counterfactual label flips ($y \neq y'$) in the Checklist dataset even when the training distribution has only 10% counterfactual label flips; **(b) Model Comparison**: However, on the original Checklist dataset [8], we achieve a comparable failure gap with the golden error rate to other model-based annotations

Out-of-Domain Reviews

To validate that the counterfactuals we augment through our pairwise classifier's annotations have generalizability to 6 out-of-domain datasets, we evaluate the reduction in error rates of the base RoBERTa model when they are trained on the pairwise classifier's data augmentation in Table 2. In

Sliced Error by Counterfactual Type %								
Model	**negation**	**quantifier**	**lexical**	**resemantic**	**insert**	**delete**	**restructure**	**shuffle**
s-CAPC-no-ablation	2.20	1.81	1.94	1.40	2.01	1.75	2.01	2.12
no-cda	19.12	18.10	21.40	20.65	17.54	20.99	18.32	17.42
invariant	14.62	4.82	4.32	3.10	7.72	7.83	6.48	9.24
trust	12.96	4.15	4.73	3.00	4.95	12.49	3.74	9.02
w-trust	5.09	3.55	8.91	10.60	7.72	5.57	10.51	10.60
random	4.74	4.04	6.92	2.22	7.42	5.55	5.72	4.96
training	4.53	3.53	6.32	2.62	7.24	5.32	5.83	4.83
Slice error when counterfactual type is ablated from training h								
PC	4.50	5.35	2.73	3.20	2.12	2.13	5.30	5.10
CAPC	4.04	2.20	4.76	2.10	4.56	4.67	3.56	4.50
p-CAPC	3.12	2.01	2.21	1.78	2.66	2.65	2.08	2.48
s-CAPC	**2.54**	**1.84**	**2.19**	**1.46**	**2.07**	**1.86**	**2.02**	**2.14**
Sliced error when counterfactual type is ablated from training h and f								
s-CAPC	11.17	13.02	7.55	13.33	4.98	5.76	10.77	9.01

Table 1: **Generalization of Counterfactual Types:** Comparison of error rates (%) sliced by different counterfactual sentence types shows that our approach s-CAPC continues to perform well even when those types are held out during training h. However, when we ablate the counterfactual type both while training f and h, our approaches performs comparably to the baselines sliced error rates. This shows that h does not just memorize the templates, but training on diverse counterfactual types continues to be important for robustness.

Test error rate %						
Model	**IMDB**	**Yelp**	**Amazon**	**SemEval**	**IMDB-cont**	**IMDB-CAD**
no-CDA	9.2	15.7	20.0	15.2	7.8	13.5
invariant	11.3	15.9	21.5	15.4	8.0	13.8
trust	9.3	15.8	20.5	15.5	8.1	13.8
w-trust	9.2	15.5	20.2	15.5	8.0	13.7
random	10.4	16.3	23.8	17.2	9.5	14.3
PC	8.0	14.3	18.1	14.2	7.4	12.9
CAPC	**7.2**	13.1	17.2	13.6	6.0	10.3
p-CAPC	9.2	13.7	15.9	13.6	5.9	10.1
s-CAPC	7.2	**10.4**	**12.9**	**11.9**	**5.5**	**9.9**
domain-trained	6.7	10.0	11.7	10.8	5.4	9.5

Table 2: Out-of-domain reviews: Using data augmentation with SST-2 counterfactuals from the Polyjuice generator and classified using s-CAPC performs comparable to a model trained on within-domain data.

the IMDB reviews dataset [61], we see an improvement in error rates from 9.2% without data augmentation to 7.2% through CAPC and s-CAPC. This out-of-domain error rate is comparable to the error rate obtained by the model trained by [3] after incorporating samples from the counterfactuals drawn from the same distribution as part of the training (6.7%). In the Yelp reviews too [2], we see a reduction from 15.7% to 10.4% whereas other baseline approaches lead to an increase in error rates. In the Amazon reviews, the s-CAPC approach (12.9%) outperforms the baselines and is comparable to the augmentation from the training split from the Amazon reviews (11.7%). Similar improvements can be seen on the SemEval [62] and IMDB contrast sets (IMDB-cont, IMDB-CAD) [11, 3]. Each of these improvements has to be viewed with the context that it was achieved in a more sample efficient manner (1,000 counterfactuals generated from the original SST-2 dataset by Polyjuice) as compared to the in-distribution training approach, where the training data has 3,400 samples from their own respective datasets. This further confirms that training on augmented counterfactuals using a generator and pairwise classifier approach is comparable to human-annotated samples from other domains, while providing us the ability to scale both in terms of domain generalization as well as labeling efficiency.

6 Conclusion

Counterfactual Data Augmentation approaches have been extensively used to train for counterfactual robustness. As the types of counterfactuals - both label-invariant and label-modifying, over which to evaluate natural language models increase, there is a need to adopt a methodology that can scale with increasing types of counterfactuals. We overcome a significant challenge in doing so, by learning an auxiliary pairwise counterfactual classifier that leverages the patterns of counterfactuals produced by vairous generative models. Using only a small amount of human annotated counterfactual samples, we demonstrate that our method can produce a dataset that improves counterfactual robustness comparable to a fully human-annotated dataset.

[2] https://www.yelp.com/dataset/

7 Limitations

In this work, we have demonstrated new methods to safely use more diverse counterfactuals and their value, but in taking on this broader goal, we discover a number of further steps that could take the work further forward. One of the limitations of our paper is that the set of counterfactuals we improved robustness over is limited and restricted to perturbations in the English language. Our analysis indicates the value of using more diverse counterfactual types that require a case-by-case contextual understanding. We show that adding more counterfactual types can be done in a sample efficient manner by using a generator trained to produce counterfactuals. However, this still suffers from the limitation that to extend to more counterfactuals and languages, a classifier which labels them by training on a small set of human annotations is required. Further, we do not investigate the quality of the counterfactuals annotated, and we do not study the performance using more nuanced counterfactuals with low levels of inter-rater agreement. Since we use an auxiliary classifier to label the generated counterfactuals, the risk of label drift remains a clear challenge and we do not control for this label drift based on the certainty of these labels from the auxiliary classifier. Further, a natural drift in concepts based on active exploration might render invalid sentences that are not grammatically or semantically correct, and new methods would be needed to filter based on these text patterns.

As in other generative models, the risk of perpetuating or amplifying biases in the generated text data continues to be important and while we believe counterfactual generation and augmentation can help address such biases, there is also uncertainty in using more flexible, generated counterfactuals. For example, it is quite possible that one of the generated counterfactuals relies on an identify term in the generated sentence, and attributes a negative sentiment spuriously based on prevalent stereotypes in the text corpus. For this reason, we refer the reader to incorporating bias mitigation strategies like [63] in addition to improving counterfactual robustness.

While we show generalization across label variance in templates, we cannot guarantee that by learning solely on label invariant counterfactuals, a classifier can generalize over label modifying counterfactuals or obtain the same levels of sample efficiency on harder classification tasks. While generators like Polyjuice [19] have been evaluated for fluency, diversity, etc., there is a need to evaluate them within the context of a task and its labels. However, the gains in robustness shown in Figure 3 and Table 2 further illustrate the need for dataset generation in an efficient manner. As future work, one can also look towards an efficient crowdsourcing strategy that minimizes the gain provided by the pairwise classifiers as each sample in the annotated dataset provides a unique and diverse counterfactual.

References

[1] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. 2018.

[2] Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. 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 5266–5274, Hong Kong, China, November 2019. Association for Computational Linguistics.

[3] Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020.

[4] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575, 2018.

[5] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *ICLR*, 2018.

[6] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. 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 4129–4142, Hong Kong, China, November 2019. Association for Computational Linguistics.

[7] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[8] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics.

[9] Florian Tramer, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Joern-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9561–9571. PMLR, 13–18 Jul 2020.

[10] Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. SSMBA: self-supervised manifold based data augmentation for improving out-of-domain robustness. *CoRR*, abs/2009.10195, 2020.

[11] Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna 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. Evaluating NLP models via contrast sets. *CoRR*, abs/2004.02709, 2020.

[12] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

[13] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[14] Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. Certified robustness to text adversarial attacks by randomized [mask], 2021.

[15] Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1520–1529, Florence, Italy, July 2019. Association for Computational Linguistics.

[16] Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. CATgen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5141–5146, Online, November 2020. Association for Computational Linguistics.

[17] Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. Generate your counterfactuals: Towards controlled counterfactual generation for text. *CoRR*, abs/2012.04698, 2020.

[18] Alexis Ross, Ana Marasovic, and Matthew E. Peters. Explaining NLP models via minimal contrastive editing (mice). *CoRR*, abs/2012.13985, 2020.

[19] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S. Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2021.

[20] Henry Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[21] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[22] Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. *CoRR*, abs/1711.10160, 2017.

[23] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness, 2019.

[24] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. Adversarial robustness through local lipschitzness. *CoRR*, abs/2003.02460, 2020.

[25] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[26] Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics, 2020.

[27] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 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, November 2019. Association for Computational Linguistics.

[28] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 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, November 2020. Association for Computational Linguistics.

[29] Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 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, July 2020. Association for Computational Linguistics.

[30] Nafise Sadat Moosavi, Marcel de Boer, Prasetya Ajie Utama, and Iryna Gurevych. Improving robustness by augmenting training sentences with predicate-argument structures. *CoRR*, abs/2010.12510, 2020.

[31] Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online, November 2020. Association for Computational Linguistics.

[32] Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, Online, August 2021. Association for Computational Linguistics.

[33] Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H. Chi, and Alex Beutel. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 219–226, New York, NY, USA, 2019. Association for Computing Machinery.

[34] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional bert contextual augmentation, 2018.

[35] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision, 2020.

[36] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review, 2020.

[37] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender bias in neural natural language processing, 2019.

[38] Yuan Zhang, Jason Baldridge, and Luheng He. Paws: Paraphrase adversaries from word scrambling, 2019.

[39] John Wieting and Kevin Gimpel. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[40] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019.

[41] Tom McCoy, Ellie Pavlick, and Tal Linzen. 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, July 2019. Association for Computational Linguistics.

[42] Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017.

[43] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[44] Toan Tran, Thanh-Toan Do, Ian Reid, and Gustavo Carneiro. Bayesian generative active deep learning. In *International Conference on Machine Learning*, pages 6295–6304. PMLR, 2019.

[45] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017.

[46] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[47] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd-workers for text-annotation tasks, 2023.

[48] Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[49] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[50] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[51] Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. First quora dataset release: Question pairs, 2017.

[52] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 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, November 2018. Association for Computational Linguistics.

[53] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858, 2019.

[54] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *CoRR*, abs/1912.02164, 2019.

[55] Divyansh Kaushik, Amrith Setlur, Eduard Hovy, and Zachary C Lipton. Explaining the efficacy of counterfactually augmented data. *International Conference on Learning Representations (ICLR)*, 2021.

[56] Nabiha Asghar. Yelp dataset challenge: Review rating prediction. *CoRR*, abs/1605.05362, 2016.

[57] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China, November 2019. Association for Computational Linguistics.

[58] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[59] Rohan Jha, Charles Lovering, and Ellie Pavlick. When does data augmentation help generalization in nlp? *CoRR*, abs/2004.15012, 2020.

[60] William Huang, Haokun Liu, and Samuel R. Bowman. 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, November 2020. Association for Computational Linguistics.

[61] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 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, June 2011. Association for Computational Linguistics.

[62] Sara Rosenthal, Noura Farra, and Preslav Nakov. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[63] Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H. Chi, and Alex Beutel. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 219–226, New York, NY, USA, 2019. Association for Computing Machinery.