Improving Robustness through Pairwise Generative Counterfactual Data Augmentation

Abstract

Counterfactual Data Augmentation (CDA) is a commonly used technique for improving robustness in natural language classifiers. However, one fundamental challenge is how to efficiently label such synthetic data, particularly when they are in regions where the model is already not confident. Most methods either rely on human-annotated templates, 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 utilize counterfactual generative models to generate a large number of diverse counterfactuals that include multiple label changing and invariant assumptions, and learn a classifier to automatically annotate more counterfactuals. Our key insight is that we can more effectively and efficiently annotate generated counterfactuals by training a *pairwise* classifier that uses the original example's ground-truth label and compares the original example to the counterfactual. We demonstrate that with a small amount of human-annotated counterfactual data (e.g., 10%), we generate a counterfactual augmentation dataset which provides an 18-20% improvement in robustness and a 14-21% reduction in errors on 3 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 (Dixon et al. 2018; Hall Maudslay et al. 2019) or crowd-sourcing (Kaushik, Hovy, and Lipton 2020). While natural text templates codify a specific number of assumptions of how counterfactual sentences and labels might vary, crowd-sourcing which can cover various types of counterfactuals, can be expensive. On the other hand, many existing methods (Xu et al. 2018; Zhao, Dua, and Singh 2018; Jia et al. 2019; Alzantot et al. 2018) simply rely on a *label-invariance* assumption: the label of the generated counterfactual example and the corresponding clean data are the same. However, this simple label-invariance assumption does not always hold true (Tramer et al. 2020; Ng, Cho, and Ghassemi 2020) and thus greatly increases the risk of using incorrect labels for counterfactual examples during training. For example, for sentiment classification, given an input (e.g., *This movie is great*), a counterfactual generator can create a small perturbation to generate a counterfactual that maintains the original label (e.g., *This movie is exquisite*), but in some cases a small perturbation can also change the ground-truth label (e.g., *This movie was supposed to be great*) (Kaushik, Hovy, and Lipton 2020). Therefore, *"how can we automatically learn the labels for counterfactual examples, given a diverse counterfactual text generator?"* remains a challenging research problem.

Beyond costly human annotation or simplifying assumptions of label invariance, researchers have explored how to make use of 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; (Kaushik, Hovy, and Lipton 2020)) or to weight generated examples based on the model uncertainty (our weighted-trust baseline; (Ovadia et al. 2019)). However, as the value of counterfactual data augmentation is to improve training in regions of lower accuracy, we will see that such approach has more limited benefits.

In this paper we propose an alternative approach to this problem: we leverage the sample efficiency of generative models to generate a large number of *diverse* counterfactuals, and train an *auxiliary classifier* which learn the *difference* between the original and counterfactual labels to annotate the generated counterfactual data. Specifically, we propose to learn the patterns of how counterfactual labels vary by using the *pair* of original and counterfactual sentences $(x, c_s(x))$ and the original label y as input to our pairwise classifier h and learn to predict the counterfactual label y'. The pipeline of our method is shown in Figure 1. We should note that only a very small set of human-annotated counterfactual examples are used to train the pairwise counterfactual classifier. Then in the inference stage, the pairwise counterfactual classifier is used to predict the labels for a large set of counterfactual examples. By using counterfactual generators and auxiliary pairwise counterfactual classifiers, we can greatly reduce the number of counterfactual examples for which we need human annotation, while providing similar gains in robustness ccomparable to a fully human annotated counterfactual dataset.

Our proposed approach addresses some of the challenges

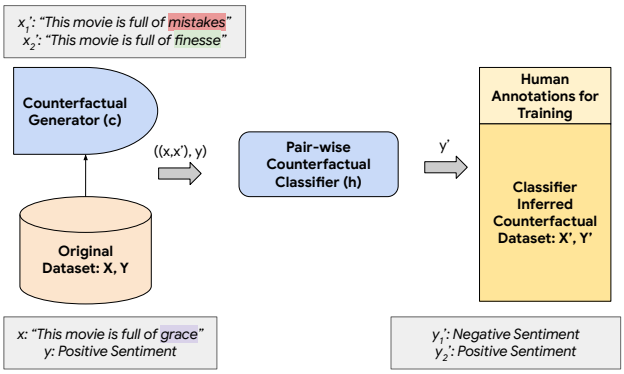

Figure 1: **Overview of proposed approach:** We propose a Pairwise Counterfactual Classifier to label generated counterfactuals (could be either label-invariant or label-modifying) at scale. We use the labeled counterfactuals as data augmentation and show it significantly improves robustness.

outlined in recent work like Checklists (Ribeiro et al. 2020) in scaling the different types of counterfactual robustness desired in models beyond accuracy. We also show that each one of the counterfactual templates that the counterfactual generator produces contribute to a different type of robustness not previously captured by our model, and hence further emphasizes the need to diversify the type of counterfactuals and generalize our performance against them for natural language classifiers. Our core contributions in this work include:

- We propose a novel pairwise counterfactual classifier that labels counterfactually generated examples at scale based on a small set of annotated counterfactuals, improving *sample efficiency* of counterfactual data augmentation.
- We model both label-invariant and label-modifying counterfactuals for the sentiment classification task on Stanford Sentiment Treebank (SST-2) dataset, and the question paraphrase task on Quora Question Pair (QQP) dataset, and show robustness improvements using just 10% of human-annotated labels.
- The generated augmented dataset when used for fine-tuning 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 IMDB, Amazon and Yelp reviews out-of-domain datasets.

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 (Ng, Cho, and Ghassemi 2020; Zeng et al. 2021) 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 (Alzantot et al. 2018). Similarly, defenses against adversarial attacks on self-attentive models have shown improvement in robustness to label invariant examples (Hsieh et al. 2019). In FairGAN (Xu et al. 2018), 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 (Wang et al. 2020) 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 (Madaan et al. 2020; Ross, Marasovic, and Peters 2020). We build on this body of work and utilize a generative model (Wu et al. 2021) that captures template-based counterfactuals to improve robustness. 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 Learning Labeling functions which provide crude estimates of the label have been used in semi-supervised methods (Ratner et al. 2017), and are further used to learn a generative model to generalize over them. Further, utilizing unlabeled data (Carmon et al. 2019) to improve adversarial robustness leverages geometric smoothing-based techniques to bridge the sample complexity gap between accuracy and robustness (Yang et al. 2020). Thus, semi-supervised learning approaches aim to generate examples where the discriminator is least confident about (Ovadia et al. 2019). Language models with very large number of parameters have also shown to be few-shot learners with minimal supervision (Brown et al. 2020). Similarly, reinforcement learning based approaches with minimal labels have been proposed to combine the objectives of accuracy and counterfactual robustness (Pitis, Creager, and Garg 2020). 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 (Garg et al. 2019; Wu et al. 2018). Learning how counterfactuals differ have been explored by comparing against gradient supervision (Teney, Abbasnedjad, and van den Hengel 2020) and the generalizability between original and counterfactuals (Kaushik, Hovy, and Lipton 2020). The generated counterfactuals have also been used for explanations (Verma, Dickerson, and Hines 2020), highlighting biases (Dixon et al. 2018) and debiasing through statistical methods (Lu et al. 2019). This rich set of contrast sets (Gardner et al. 2020), checklists (Ribeiro et al. 2020), paraphrases (Zhang, Baldridge, and He 2019; Wieting and Gimpel 2018),

adversarial schemes (Sakaguchi et al. 2019) and lexical diagnostic datasets (McCoy, Pavlick, and Linzen 2019) form the foundation of our method, which re-purposes them to build a counterfactual generative model and improve counterfactual robustness.

3 Methodology

Our 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'? While we can 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 (Goodfellow, Shlens, and Szegedy 2015; Jia et al. 2019; Alzantot et al. 2018), 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. Since the counterfactual $c_s(x)$ can either contribute to a label flip or not, it is important for us to understand the patterns in the counterfactuals that vary the labels. We further 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 a held-out counterfactual dataset (X', Y'):

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

To achieve this goal, we generate our training counterfactual inputs $c_s(x) \in X'_t$ (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 (Ribeiro et al. 2020; Wu et al. 2021) 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 $\mathcal{E}_t$ 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:

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

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

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 learn the labels for counterfactual examples X'_g with an access to a limited human-annotated counterfactual data (X'_a, Y'_a), where $|Y'_a| \ll |Y'_g|$, while achievable 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 takes in as input both the original dataset $(x, y) \in (X, Y)$, and a corresponding counterfactual $c_s(x) \in X'_t$ and the human-annotated labels $y' \in Y'_a$. The classifier h is trained on *pairs* of input sentences $x, c_s(x)$ and the original label y to predict $y' \in Y'_a$.

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:

$$\ell_h = E_{(x,y)\in(X,Y),(c_s(x),y')\in(X'_a,Y'_a)}CE(h(x, c_s(x), y), y') \tag{5}$$

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:

$$y' = h(x, c_s(x), y) : (x, y) \in (X, Y), c_s(x) \in X'_g \tag{6}$$

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:

$$y' \in Y'_g = h(x, c_s(x), y, f(x), f(c_s(x))) : \quad (7)$$
$$(x, y) \in (X, Y), c_s(x) \in X'_g$$

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 7 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.

This is similar to boosting (Freund and Schapire 1997) related methods where the original classifier f's errors on the counterfactuals is being learnt by the auxiliary classifier h. This helps us understand why the pairwise counterfactual classification task might be easier and perform better than simply annotating the counterfactual example $c_s(x)$ using the original classifier f. We can draw parallels to boosting (Freund and Schapire 1997) and draw insights as to why the number of samples required might be less. We now proceed to how our methodology compares to baselines (including using f for annotation) on held-out counterfactual robustness and the impact it has on the original accuracy.

4 Evaluation

We evaluate on two NLP tasks, sentiment classification and question paraphrase, using two datasets namely the Stanford Sentiment Treebank (SST-2) (Socher et al. 2013) and the Quora Question Pair (QQP) (Iyer, Dandekar, and Csernai 2017; Wang et al. 2018). Below, we briefly explain the problem set up in both datasets, how the counterfactuals are generated in each and the corresponding counterfactual datasets across which we evaluate counterfactual robustness.

Counterfactual Generator: Polyjuice

We use a general purpose counterfactual text generator called Polyjuice (Wu et al. 2021), which extends CheckList (Ribeiro et al. 2020), 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 (Kaushik, Hovy, and Lipton 2020), adding or changing quantifiers (Gardner et al. 2020), shuffle key phrases (Zhang, Baldridge, and He 2019), word or phrase swaps which do not alter POS tags (Sakaguchi et al. 2019) or parse trees (Wieting and Gimpel 2018), along with insertions or deletion of constraints that do not alter the parse tree (McCoy, Pavlick, and Linzen 2019). Other text generative models like (Zhao, Dua, and Singh 2018; Kaushik, Hovy, and Lipton 2020; Jia et al. 2019) that improve adversarial robustness or like (Keskar et al. 2019; Dathathri et al. 2019) that allow controlled generation could be used as well.

Tasks and Datasets

Stanford Sentiment Treebank: We use the sentiment analysis dataset SST-2 (Socher et al. 2013) which assigns a binary sentiment (negative/positive) to a sentence mined from RottenTomatoes movie reviews. The corresponding counterfactuals are generated using the Polyjuice generator (Wu et al. 2021). 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 (Iyer, Dandekar, and Csernai 2017; Wang et al. 2018), 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 (Wu et al. 2021) 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?

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 corresponding original sentence: $y' = y$.
- **trust**: we trust the classifier f to annotate the counterfactual labels $y' = f(c_s(x))$.
- **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 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 (Liu et al. 2019) 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.

Experiment Setup

In both datasets, we have a small number of counterfactual human annotations available (SST-2: 2,000; QQP: 1,911) (Wu et al. 2021). 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.

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.

To test the methodology on out-of-domain datasets, we test on sentiment analysis tasks in 3 class-balanced reviews datasets - IMDB movie reviews, Amazon reviews, and Yelp reviews (Kaushik et al. 2021). The IMDB reviews (1,700) were collected by (Kaushik, Hovy, and Lipton 2020) through careful human elicitation to produce label varying counterfactuals of existing IMDB reviews. In the Yelp reviews (Asghar 2016), the task is to predict the ratings of 115,907 reviews on a scale of 1-5, and in the Amazon reviews (Ni, Li, and McAuley 2019), we evaluate on the 57,947 reviews in the clothing product category. Each one of these datasets were 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.

5 Results

Improving Counterfactual Robustness

To demonstrate the effectiveness of our proposed methods: pairwise-counterfactual (**PC**) and classifier-aware pairwise-counterfactual (**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.

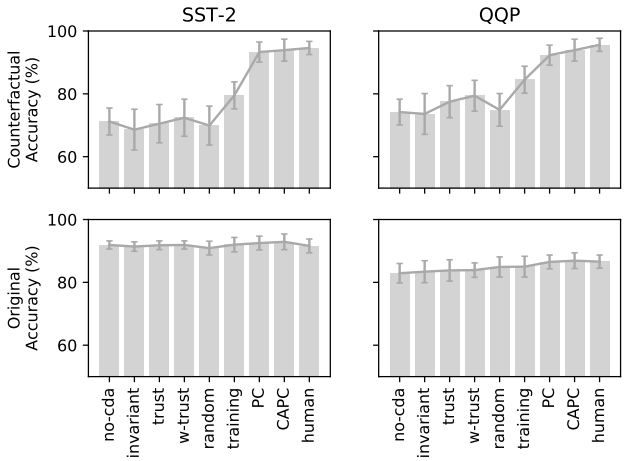

Figure 2: **(a) Robustness:** (first row) Training on 10% of human-annotated counterfactuals, and annotating the rest using the auxiliary classifier, we achieve a comparable improvement in robustness (lower error rate) for both Stanford Sentiment and Quora Question Pair datasets; **(b) Accuracy:** This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

We can clearly see that (1) the error rate of our proposed methods: **PC** and **CAPC** both significantly outperform other five baselines on models' robustness. (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

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

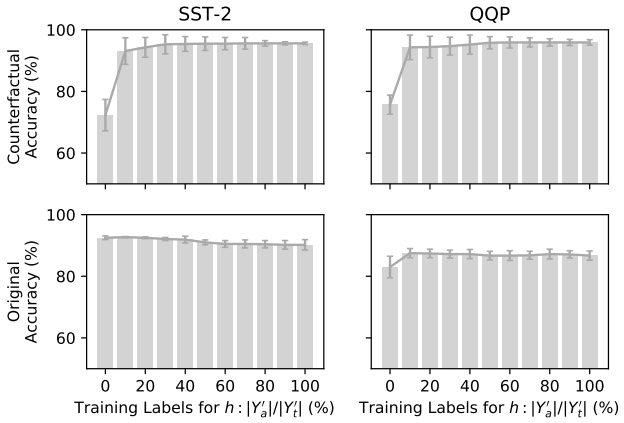

Figure 3: **Impact of training size:** As the number of samples $|Y'_a|$ increases more than 10%, there is not much headroom in counterfactual accuracy, and does not significantly impact the accuracy on the held-out original test dataset on both SST-2 and QQP datasets (overlapping error bounds).

datasets, the improvement in accuracy and robustness in providing more human 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, but including those examples at annotation time. The results are shown in Figure 4. We see that 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. 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 shows that our approach is comparable with baselines across all counterfactual types. This is consistent with existing work (Jha, Lovering, and Pavlick 2020) 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 (Ribeiro et al. 2020) 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.

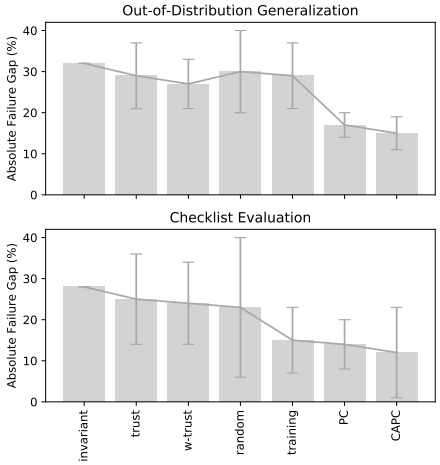

Figure 4: **(a) Out-of-distribution data generalization**: Our methods generalize well over different label-invariant distributions with 90% counterfactual label flips ($y \neq y'$) in the Checklist dataset even when the training distribution has only 10% counterfactual label flips; **(b) Model Comparison**: However, on the original Checklist dataset (Ribeiro et al. 2020), 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 has generalizability to 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 the IMDB reviews dataset, we see an improvement in error rates from 9.2% without data augmentation to 7.2% through CAPC. This out-of-domain error rate is comparable to the error rate obtained by the model trained by (Kaushik, Hovy, and Lipton 2020) after incorporating samples from the counterfactuals drawn from the same distribution as part of the training (6.7%). In the Yelp reviews too, we see a reduction from 15.7% to 13.1% whereas other baseline approaches lead to an increase in error rates. Finally, in the Amazon reviews, the CAPC approach (17.2%) outperforms the baselines and is comparable to the augmentation from the training split from the Amazon reviews (16.7%). Each of these improvements have to be viewed with the context that it was achieved in a more sample efficient manner (1,000 counterfactuals

Sliced Error when Counterfactual Type is Ablated %

Model	negation	quantifier	lexical	resemantic	insert	delete	restructure	shuffle
CAPC-no-ablation	3.20	2.01	1.94	2.00	2.10	2.45	3.32	4.03
Generalization when counterfactual type is ablated from training h								
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
CAPC	**4.04**	**2.20**	4.76	**2.10**	4.56	4.67	**3.56**	**4.50**
PC	4.50	5.35	**2.73**	3.20	**2.12**	**2.13**	5.30	5.10
Generalization when counterfactual type is ablated from training h and f								
CAPC	11.17	13.02	7.55	13.33	4.98	5.76	10.77	9.01
PC	7.02	7.40	4.63	5.35	2.42	2.54	6.85	9.34

Table 1: **Generalization of Counterfactual Types:** Increase in error rates (%) of different counterfactual sentence types shows that our approaches CAPC and PC generalize better when those types are held out during training h. However, when we ablate the counterfactual type both while training f and h, our approaches perform comparably to the baselines. This shows that h does not just memorize the templates, but training on diverse counterfactual types is important for robustness

Test error rate %

Model	IMDB	Yelp	Amazon
no-CDA	9.2	15.7	20.0
invariant	11.3	15.9	21.5
trust	9.3	15.8	20.5
w-trust	9.2	15.5	20.2
random	10.4	16.3	23.8
CAPC	**7.2**	**13.1**	**17.2**
PC	8.0	14.3	18.1
domain-trained	6.7	13.0	16.7

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

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.

Discussion

The need to ensure that natural language models predict reliably when sentences are perturbed in specific syntactic and semantically meaningful ways, beyond the observed training dataset is well established. Even though a checklist based framework introduces many constraints at once, it is important to ensure that enforcing one does not counter another counterfactual behavior. We now discuss how future work can build on top of our framework to overcome these limitations.

Importance of diverse templates While we show generalization across label variance in templates, we cannot guarantee that by learning solely on label invariant counterfactuals, our classifier can generalize over label modifying counterfactuals. Here, it is important to analyze counterfactual generators as to what type of sentences they generate and how it might be relevant to downstream tasks. While generators like Polyjuice (Wu et al. 2021) have been evaluated for fluency, diversity, etc., there is a need to evaluate them within the context of a task and its labels.

We improve what we measure We acknowledge that the set of counterfactuals we improved robustness over is limited. We are not claiming to have automated improving robustness of natural language classifier. Instead, our analysis further indicates the need for 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 and a classifier which labels them by training on a small set of human annotations.

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 that of a fully human-annotated dataset.

References

Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.-J.; Srivastava, M.; and Chang, K.-W. 2018. Generating Natural Language Adversarial Examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2890–2896. Brussels, Belgium: Association for Computational Linguistics.

Asghar, N. 2016. Yelp Dataset Challenge: Review Rating Prediction. *CoRR*, abs/1605.05362.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165.

Carmon, Y.; Raghunathan, A.; Schmidt, L.; Liang, P.; and Duchi, J. C. 2019. Unlabeled Data Improves Adversarial Robustness. arXiv:1905.13736.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2019. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. *CoRR*, abs/1912.02164.

Dixon, L.; Li, J.; Sorensen, J.; Thain, N.; and Vasserman, L. 2018. Measuring and Mitigating Unintended Bias in Text Classification.

Freund, Y.; and Schapire, R. E. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1): 119–139.

Gardner, M.; Artzi, Y.; Basmova, V.; Berant, J.; Bogin, B.; Chen, S.; Dasigi, P.; Dua, D.; Elazar, Y.; Gottumukkala, A.; Gupta, N.; Hajishirzi, H.; Ilharco, G.; Khashabi, D.; Lin, K.; Liu, J.; Liu, N. F.; Mulcaire, P.; Ning, Q.; Singh, S.; Smith, N. A.; Subramanian, S.; Tsarfaty, R.; Wallace, E.; Zhang, A.; and Zhou, B. 2020. Evaluating NLP Models via Contrast Sets. *CoRR*, abs/2004.02709.

Garg, S.; Perot, V.; Limtiaco, N.; Taly, A.; Chi, E. H.; and Beutel, A. 2019. Counterfactual Fairness in Text Classification through Robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, 219–226. New York, NY, USA: Association for Computing Machinery. ISBN 9781450363242.

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

Hall Maudslay, R.; Gonen, H.; Cotterell, R.; and Teufel, S. 2019. 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)*, 5266–5274. Hong Kong, China: Association for Computational Linguistics.

Hsieh, Y.-L.; Cheng, M.; Juan, D.-C.; Wei, W.; Hsu, W.-L.; and Hsieh, C.-J. 2019. On the Robustness of Self-Attentive Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1520–1529. Florence, Italy: Association for Computational Linguistics.

Iyer, S.; Dandekar, N.; and Csernai, K. 2017. First Quora Dataset Release: Question Pairs.

Jha, R.; Lovering, C.; and Pavlick, E. 2020. When does data augmentation help generalization in NLP? *CoRR*, abs/2004.15012.

Jia, R.; Raghunathan, A.; Göksel, K.; and Liang, P. 2019. 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)*, 4129–4142. Hong Kong, China: Association for Computational Linguistics.

Kaushik, D.; Hovy, E.; and Lipton, Z. 2020. Learning The Difference That Makes A Difference With Counterfactually-Augmented Data. In *International Conference on Learning Representations*.

Kaushik, D.; Setlur, A.; Hovy, E.; and Lipton, Z. C. 2021. Explaining the Efficacy of Counterfactually Augmented Data. *International Conference on Learning Representations (ICLR)*.

Keskar, N. S.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *CoRR*, abs/1909.05858.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Lu, K.; Mardziel, P.; Wu, F.; Amancharla, P.; and Datta, A. 2019. Gender Bias in Neural Natural Language Processing. arXiv:1807.11714.

Madaan, N.; Padhi, I.; Panwar, N.; and Saha, D. 2020. Generate Your Counterfactuals: Towards Controlled Counterfactual Generation for Text. *CoRR*, abs/2012.04698.

McCoy, T.; Pavlick, E.; and Linzen, T. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3428–3448. Florence, Italy: Association for Computational Linguistics.

Ng, N.; Cho, K.; and Ghassemi, M. 2020. SSMBA: Self-Supervised Manifold Based Data Augmentation for Improving Out-of-Domain Robustness. *CoRR*, abs/2009.10195.

Ni, J.; Li, J.; and McAuley, J. 2019. 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)*, 188–197. Hong Kong, China: Association for Computational Linguistics.

Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J. V.; Lakshminarayanan, B.; and Snoek, J. 2019. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. arXiv:1906.02530.

Pitis, S.; Creager, E.; and Garg, A. 2020. Counterfactual Data Augmentation using Locally Factored Dynamics. arXiv:2007.02863.

Ratner, A.; Bach, S. H.; Ehrenberg, H. R.; Fries, J. A.; Wu, S.; and Ré, C. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. *CoRR*, abs/1711.10160.

Ribeiro, M. T.; Wu, T.; Guestrin, C.; and Singh, S. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4902–4912. Online: Association for Computational Linguistics.

Ross, A.; Marasovic, A.; and Peters, M. E. 2020. Explaining NLP Models via Minimal Contrastive Editing (MiCE). *CoRR*, abs/2012.13985.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2019. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. arXiv:1907.10641.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. Seattle, Washington, USA: Association for Computational Linguistics.

Teney, D.; Abbasnedjad, E.; and van den Hengel, A. 2020. Learning What Makes a Difference from Counterfactual Examples and Gradient Supervision. arXiv:2004.09034.

Tramer, F.; Behrmann, J.; Carlini, N.; Papernot, N.; and Jacobsen, J.-H. 2020. Fundamental Tradeoffs between Invariance and Sensitivity to Adversarial Perturbations. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 9561–9571. PMLR.

Verma, S.; Dickerson, J.; and Hines, K. 2020. Counterfactual Explanations for Machine Learning: A Review. arXiv:2010.10596.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355. Brussels, Belgium: Association for Computational Linguistics.

Wang, T.; Wang, X.; Qin, Y.; Packer, B.; Li, K.; Chen, J.; Beutel, A.; and Chi, E. 2020. CAT-Gen: Improving Robustness in NLP Models via Controlled Adversarial Text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5141–5146. Online: Association for Computational Linguistics.

Wieting, J.; and Gimpel, K. 2018. 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)*, 451–462. Melbourne, Australia: Association for Computational Linguistics.

Wu, T.; Ribeiro, M. T.; Heer, J.; and Weld, D. S. 2021. 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.

Wu, X.; Lv, S.; Zang, L.; Han, J.; and Hu, S. 2018. Conditional BERT Contextual Augmentation. arXiv:1812.06705.

Xu, D.; Yuan, S.; Zhang, L.; and Wu, X. 2018. FairGAN: Fairness-aware Generative Adversarial Networks. In *2018 IEEE International Conference on Big Data (Big Data)*, 570–575.

Yang, Y.-Y.; Rashtchian, C.; Zhang, H.; Salakhutdinov, R.; and Chaudhuri, K. 2020. Adversarial Robustness Through Local Lipschitzness. *CoRR*, abs/2003.02460.

Zeng, J.; Zheng, X.; Xu, J.; Li, L.; Yuan, L.; and Huang, X. 2021. Certified Robustness to Text Adversarial Attacks by Randomized [MASK]. arXiv:2105.03743.

Zhang, Y.; Baldridge, J.; and He, L. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. arXiv:1904.01130.

Zhao, Z.; Dua, D.; and Singh, S. 2018. Generating Natural Adversarial Examples. In *ICLR*.