Don't Cheat! Catching Spurious Correlation in NLI Tasks

Guanwen Qiu* and **Hainiu Xu*** and **Zhirui Yao***
School of Engineering and Applied Sciences
University of Pennsylvania
{guanwenq,seacow,zhiruiy}@seas.upenn.edu

Abstract

Recent advancements in language models in Natural Language Processing (NLP) have brought another trend of discussion and excitement toward Artificial Intelligence. Yet, Large Language Models (LLMs) often consist of more than 100 billion parameters and demand tremendous resources to train. On the other hand, models belonging to the Pretrain-and-Finetune paradigm contain much fewer parameters and can achieve promising performance in various NLP. However, the seemingly promising performance of such models could be largely due to model learning spurious correlations in the training dataset. In this work, we catch such behavior of language models on various Natural Langauge Inference (NLI) datasets. Further, we attempted several remedies to prevent the model from learning spurious cues. Albeit the effort to eliminate spurious correlation from both a data-centric and modeling perspective, only data fusion provides improvements.

1 Introduction

Natural Langauge Processing (NLP) is a sub-field of Artificial Intelligence that dedicate to granting machines intelligence by teaching them to understand human languages. The research of NLP with deep learning has gone through numerous stages, of which the most impactful is the Pretrain-and-Finetune paradigm and the in-context learning paradigm.

The Pretrain-and-Finetune paradigm contains milestone works such as the Transformer architecture, BERT, GPT-1, and GPT-2 (Vaswani et al., 2017; Devlin et al., 2018; Radford et al., 2019), both of which had profound impacts on numerous areas including Computer Vision, Robotics, and Multi-modal learning (Li et al., 2019; Dosovitskiy et al., 2020; Ahn et al., 2022; Radford et al., 2021).

* Equal contribution.

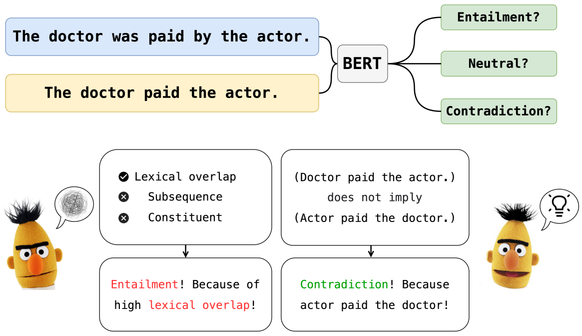

Figure 1: Demonstration of the NLI tasks and spurious correlation. The top half of the figure shows a typical NLI task where a `premise` (blue) and a `hypothesis` (yellow) are fed to a language model, which predicts the `relationship` (green). The bottom half shows a model that learns spurious correlation (left) and a model that learns the proposed task (right).

The in-context learning paradigm includes trendy models such as the GPT-3, the T5-series, the T0-series, LaMDA, and PaLM(Brown et al., 2020; Raffel et al., 2020; Sanh et al., 2021; Thoppilan et al., 2022; Chowdhery et al., 2022). The in-context learning models are gigantic. This is mainly because that in-context learning and few-shot learning is an emergent abilities that only models with more than 100 billion parameters can acquire. Such a class of large models is often referred to as Large Language Models (LLMs).

Training LLMs is devastating for the environment. For instance, training the 175-billion-parameter GPT3 would require *"several thousand petaflop/s-days of computing during pre-training"* (Brown et al., 2020). Albeit the strong few-shot learning capabilities, the performance of LLMs on many NLP tasks is oftentimes not on par with models from the Pretrain-and-Finetune paradigm with drastically fewer parameters.

Natural Language Inference (NLI) is one such task. The largest GPT3 model over 175 billion parameters is only able to achieve around 70% accuracy on the Adversarial NLI (ANLI) dataset whereas models from the BERT-family as well as the ELECTRA can achieve over 90% accuracy through fine-tuning (Brown et al., 2020; Devlin et al., 2018; Liu et al., 2019; Clark et al., 2020; He et al., 2020). Such a performance discrepancy between the two paradigms make people wonder what makes the small models of the Pretrain-and-Finetune paradigm "data efficient".

Recent works have shown that finetuned language models have poor generalization capabilities (Tu et al., 2020a). The lack of generalization capability is largely due to the model picking up spurious cues in the dataset. In other words, finetuned language models oftentimes learn to take a shortcut which is much easier to learn than the actual proposed task. Such behavior is undesirable and has become a crux in machine learning.

To this end, we investigate how the model of the Pretrain-and-Finetune paradigm utilizes spurious curs in NLI datasets and demonstrate the disastrous consequence of learning spurious correlation. Specifically, we use the BERT-base model (Devlin et al., 2018) and NLI datasets including Standford NLI (SNLI), Multi-Genre NLI (MNLI), Adversarial NLI (ANLI), and Heuristic Analysis for NLI Systems (HANS) (Bowman et al., 2015; Williams et al., 2018; Nie et al., 2020; McCoy et al., 2019). Further, we attempt numerous remedies from both a data-centric and modeling perspective, demonstrating the challenge of mitigating spurious correlation even on the NLI task alone.

2 Related Work

Language Model Language models refer to the class of neural networks that are designed to model the meaning of words via dense and real-valued vectors. Based on the distributional semantics (Harris, 1954), neural networks represent the meaning of a word using its surrounding words.

Earlier works utilize feed-forward networks (Mikolov et al., 2013; Iyyer et al., 2015), some take advantage of the sequential nature of languages by using Recurrent Neural Networks (Rumelhart et al., 1985; Hochreiter and Schmidhuber, 1997; Jordan, 1997), and some conduct 1-dimensional convolution on word embeddings (Chen, 2015; Conneau et al., 2017).

Ever since the invention of the Transformer architecture (Vaswani et al., 2017), study on language model has split into three different genres, namely *autoencoding*, which utilize the Transformer encoder, *autoregressive*, which utilize the Transformer decoder, and *seq2seq*, which incorporate the complete Transformer architecture. The autoencoding models such as the BERT model family and the ELECTRA are widely used in the Pretrain-and-Finetune paradigm for their bi-directionality (Devlin et al., 2018; Liu et al., 2019; He et al., 2020; Clark et al., 2020)

The autoregressive models such as the GPT family, LaMDA, and PaLM are widely used for coherent text generation as the unidirectional training enforces such models to easily encapsulate grammatical and coherence knowledge in its learned embeddings (Radford et al., 2019; Brown et al., 2020; Thoppilan et al., 2022; Chowdhery et al., 2022).

The seq2seq language models such as T5 and T0 take full advantage of both the encoder and decoder of the Transformer (Raffel et al., 2020; Sanh et al., 2021). The encoder allows the model to learn a better representation of the context, which in turn improves the generation quality of the decoder.

In this study, we focus on the autoencoding language models, BERT in particular. NLI demands the model to comprehend the meaning of two sentences. Therefore, we wish to leverage the bi-directionality of BERT and finetune the pre-trained BERT with various NLI datasets.

The Pretrain-and-Finetune paradigm Popularized by BERT, the Pretrain-and-Finetune paradigm is a significant improvement on the train-from-scratch approaches (Devlin et al., 2018; Liu et al., 2019; He et al., 2020; Clark et al., 2020). Specifically, language model architectures are designed so that it is able to conduct tasks ranging from Natural Language Understanding to Natural Language Generation (Qiu et al., 2020). By pretraining on large amounts of texts, including multiple NLP task datasets, language models can learn prominent contextualized word embeddings. During fine-tuning, depending on the amount of available training data, researchers can choose from (1) unfreeze all model weights, (2) unfreeze only the top layers, (3) completely freeze the model weights and only finetune the final feedforward networks (Devlin et al., 2018). In our case, we adopt strategy (1) and we use a pretrained BERT provided by Huggingface (Wolf et al., 2020).

Our code and results are available at: `https://github.com/SeacowX/ESE546-Project`

Spurious correlation For generalization to real-world target domains, a learned model's output should be representative of nature's true distribution under the target domain. Mathematically, if two features A and B are independent under nature's distribution

$$\{A \perp\!\!\!\perp B | A, B \sim U\} \quad (1)$$

then we expect the output of the model reflect the same relationship

$$\{A \perp\!\!\!\perp B | A, B \sim S\} \quad (2)$$

Where S is the probability space induced by the output of the model. However, this is often not the case. The problem is two-fold. On one hand, our training dataset may be unrepresentative of the true distribution in the first place (McMilin, 2022). On the other hand, the model may choose to learn unwanted correlations across features (D'Amour et al., 2020). In this paper, we restrict our scope to investigate the observed spurious correlation between the hypothesis and labels of an NLI task. It is a fact that NLI dataset such as SNLI contains severe spurious correlations between hypothesis and labels (Tu et al., 2020b). For example, McCoy create HANS, a synthetic, adversarial NLI dataset based on three fallible syntactic heuristics to test whether an NLI model is able to learn non-shallow text knowledge. It is quite surprising that the model achieving over 80 accuracies on SNLI dataset gives chance-level accuracy on HANS(McCoy et al., 2019). Spurious correlation is also studied in machine learning fairness literature where researchers try to remove demographic attributes from text data (Elazar and Goldberg, 2018).

Data augmentation in Text Counterfactual data augmentation (CDA) is a strategy that uses casual interventions to break the association between the biased term and the label (Datta). Some of the CDA methods proposed require extensive human intervention, such as specifying pre-defined substitution terms, generating templates, or labeling examples (Garg et al., 2019; Ribeiro et al., 2020). Others have mainly focused on incorporating the learning of spurious causal correlation and generating adversarial data as part of the training task. Wang and Culotta (2021) have proposed to automatically generate counterfactuals of texts by first using statistical matching to identify the spurious correlation between features and labels and replacing casual features with antonyms while reversing their labels to generate counterfactual samples. Jin et al. (2020) propose a method to generate high-profile adversarial example through word importance ranking and transformers to replace casual keywords through synonym expression, POS checking, and semantics similarity checking. They find out that both the after-attack accuracy and perturbed words ratio get higher, indicating the huge difficulty in generating adversarial data.

Adversarial Filtering Adversarial Filtering (AF) adepts ideology from the Boosting paradigm (Kearns and Valiant, 1994; Kearns, 1988). In short, AF works by selecting data that are difficult to learn based on some trained models. The term is first proposed in Zellers et al. (2018) where AF is used with a bias classification model to de-bias the dataset. Recent works adopt AF to LLMs and reported significantly improved performance on several NLP tasks including NLI (Liu et al., 2022). Therefore, we adopt a similar strategy to our study and conducted AF using a finetuned BERT-base model.

3 Task Formulation

The task of NLI is concerned with determining the logical relation between two sentences. Specifically, given A typical NLI dataset $\mathcal{D}$ of n samples consists of a set of premises $\{\mathcal{P}\}_{i=1}^{n}$, a corresponding set of hypothesis $\{\mathcal{H}\}_{i=1}^{n}$, and a set of label $\{y\}_{i=1}^{n}$ where $y \in \{\texttt{entailment}, \texttt{neutral}, \texttt{contradiction}\}$. Given a premise, $p_i \in \mathcal{P}$, and a hypothesis $h_i \in \mathcal{H}$, a model parameterized by weights $\mathcal{W}$ takes the premise and the hypothesis as input and output the predicted label

$$\hat{y} = f_{\mathcal{W}}\Big(p_i \oplus h_i\Big)$$

where $\oplus$ represents the concatenation operation of sentences. In BERT, the concatenation is done as follows

```
[CLS][Sentence-1][SEP][Sentence-2][SEP]
```

The embedding given by the `[CLS]` token will be used as the representation of the pair of sentences. The prediction is obtained by projecting the `CLS` embedding to the target space $\hat{y} \in \mathbb{R}^3$ using a feed-forward neural network.

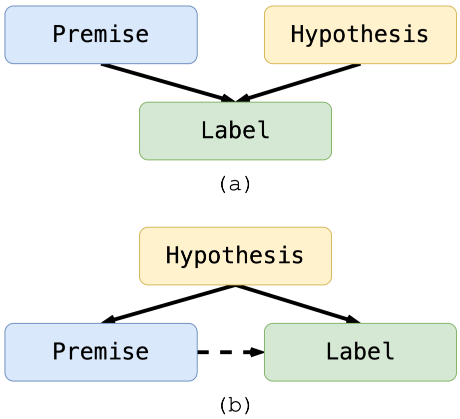

Figure 2: A simple causal graph for the NLI task. Figure (a) shows the ideal setup where we wish to draw a conclusion on the relationship between (`premise`, `hypothesis`) pair and the `label`. Figure (b) shows the consequence of the Hypothesis-only correlation where `hypothesis` is now a confounding covariate that prevents us from drawing a causal relationship between the `premise` and the `label` (represented with dashed arrow).

4 Cathcing Spurious Correlations

In our study, we consider two types of spurious correlation in NLI, namely the *hypothesis-only* spurious correlation and the *premise-hypothesis* spurious correlation.

4.1 Hypothesis-only Correlation

In an NLI task, we want to learn the possibility of a label given the corresponding hypothesis and premise $P(L|H,P)$. It is obvious that given only the hypothesis or premise, the performance of the model shall not be better than chance. However, because NLI datasets such as SNLI contain severe spurious correlations between their hypothesis and labels, this condition is substantially violated.

If we think of NLI as a causal inference task where we wish to study the causal relation between a (`premise, hypothesis`) pair and a `label`, the spurious correlation between the `hypothesis` and the `label` becomes the confounder and prevents the model from building meaningful causal relationships (Figure 2).

4.2 Syntactic Correlation

Another spurious correlation that exists in most of the NLI datasets is the syntactic correlation between premise-hypothesis pairs and the label (Figure 3). Such spurious cues are discovered by McCoy et al. (2019). Overall, there are three genres of such syntactic correlation: *Lexical Overlap*, where the Jaccard distance between the `premise` and the `hypothesis` is high, *Subsequence*, where the `hypotehsis` contains tokens only from the `premise`, and *Constituent*, where the argument and triggers (verbs) of the `hypothesis` are identical to that of the `premise`. We explore these types of spurious cues by evaluating our remedies also on the HANS dataset (McCoy et al., 2019).

In the following experiments, we try both data-centric and model-centric approaches to remedy the spurious correlation. We measure the effectiveness of each method by evaluating the finetuned model with the HANS dataset, which serves as the modern benchmark for measuring the amount of syntactic spurious correlation models learned for NLI tasks.

5 Data-Centric Remedy

5.1 Data Augmentation

Performing data augmentation to remove spurious correlation for the NLI task involves two steps: first, we need to identify the keywords that lead to cheating during training; second, we need to mutate such spurious words, cut their connection with the labels, and add the newly augmented samples to the original data set to evaluate for the original task again and see if it gains better generalization capabilities. For simplicity, we merge the entailment and neutral label in SNLI to be entailment, and contradiction to be non-entailment to binarize the labels.

Spurious Word Identification For the NLI task, we determine that if training a simple binary classifier (entailment VS. non-entailment) to predict labels with only the hypothesis could lead to an accuracy higher than a random guess, then it means that there are at least some spurious words in the hypothesis that correlate with labels but might not be helpful for learning the relationship between hypothesis and premise (the NLI task itself). The binary classifier we choose in this project is logistic regression. To identify the spurious words, we further apply the LIME algorithm (Ribeiro et al., 2016) to find the keywords in each hypothesis that lead to a confident prediction of the NLI labels. LIME explains the prediction of any classifier by learning the perturbation of the original instances locally around the prediction. We set an arbitrary threshold for the confidence of the prediction given by the logistic regression model, and probe the importance score of each word given by LIME. If a

	Model	Training Data	Testing Data	Accuracy	Macro F1
Majority Class Baseline	Majority Class	–	SNLI-test	0.3368	–
	Majority Class	–	HANS-test	0.5000	–
	Majority Class	–	MNLI-test	0.3522	–
	Majority Class	–	ANLI-test	0.3344	–
SNLI	BERT-base	SNLI-train	SNLI-test	0.8457	0.8311
	BERT-base	SNLI-train	HANS-test	0.5110	0.3522
	BERT-base	SNLI-hypothesis	SNLI-test	0.4952	0.4619
	BERT-base	SNLI-hypothesis	HANS-test	0.5036	0.3754
	BERT-base	SNLI-premise	SNLI-test	0.3237	0.1592
	BERT-base	SNLI-premise	HANS-test	0.5000	0.3285
HANS	BERT-base	HANS-train	SNLI-test	0.3252	0.2406
	BERT-base	HANS-train	HANS-test	0.9999	0.9999
Multi-Genre NLI	BERT-base	MNLI-train	SNLI-test	0.7118	0.6850
	BERT-base	MNLI-train	HANS-test	0.5263	0.3922
	BERT-base	MNLI-train	MNLI-test	0.7817	0.7607
	BERT-base	MNLI-train-hypothesis	SNLI-test	0.4254	0.3885
	BERT-base	MNLI-train-hypothesis	HANS-test	0.4739	0.4450
	BERT-base	MNLI-train-hypothesis	MNLI-test	0.4692	0.4457
Adversarial NLI	BERT-base	ANLI-train	SNLI-test	0.6223	0.5952
	BERT-base	ANLI-train	HANS-test	0.5067	0.4828
	BERT-base	ANLI-train	ANLI-test	0.4415	0.4124
	BERT-base	ANLI-train-hypothesis	SNLI-test	0.4301	0.3773
	BERT-base	ANLI-train-hypothesis	HANS-test	0.4822	0.3916
	BERT-base	ANLI-train-hypothesis	ANLI-test	0.3647	0.3236
Data Augmentation (Section 5.1)	BERT-base	SNLI-AUG	SNLI-test	0.8294	0.7772
	BERT-base	SNLI-AUG	HANS-test	0.4997	0.3287
	BERT-base	SNLI-AUG-hypothesis	SNLI-test	0.6702	0.5319
	BERT-base	SNLI-AUG-hypothesis	HANS-test	0.4860	0.4220
Adversarial Filtering (Section 5.2)	BERT-base	SNLI-AF	SNLI-test	0.3332	0.3172
	BERT-base	SNLI-AF	HANS-test	0.4986	0.3479
	BERT-base	SNLI-AF-hypothesis	SNLI-test	0.3798	0.3494
	BERT-base	SNLI-AF-hypothesis	HANS-test	0.4953	0.4140
Data Fusion (Section 5.3)	BERT-base	SNLI(90%)+HANS(10%)	SNLI-test	0.8327	0.8150
	BERT-base	SNLI(90%)+HANS(10%)	HANS-test	0.5035	0.4462
	BERT-base	SNLI(70%)+HANS(30%)	SNLI-test	0.8370	0.8219
	BERT-base	SNLI(70%)+HANS(30%)	HANS-test	0.6909	0.6530
	BERT-base	SNLI(50%)+HANS(50%)	SNLI-test	0.8328	0.8177
	BERT-base	SNLI(50%)+HANS(50%)	HANS-test	0.8425	0.8286
	BERT-base	SNLI(30%)+HANS(70%)	SNLI-test	0.8084	0.7910
	BERT-base	SNLI(30%)+HANS(70%)	HANS-test	1.0000	1.0000

Table 1: Expeirment results of the BERT-base model on various NLI datasets. Data name with postfix `hypothesis` means only the `hypothesis` is provided to the model during training. Data name with postfix `premise` means only the `premise` is provided to the model during training.

particular word in a hypothesis has such high importance score that it contributes to most of the confidence of the prediction exceeding a random guess, then we determine that it is a spurious word. The threshold we pick for prediction probability is 0.8 (since random guesses should have 0.66 confidence in predicting the correct label), and 0.1 for word importance score (which will help reduce probability from 0.8 to around 0.7, closer to random guessing).

Spurious Word Transformer After identifying the spurious words, we use the following logic

Heuristic	Premise	Hypothesis	Label
Lexical overlap heuristic	The banker near the judge saw the actor.	The banker saw the actor.	E
	The lawyer was advised by the actor.	The actor advised the lawyer.	E
	The doctors visited the lawyer.	The lawyer visited the doctors.	N
	The judge by the actor stopped the banker.	The banker stopped the actor.	N
Subsequence heuristic	The artist and the student called the judge.	The student called the judge.	E
	Angry tourists helped the lawyer.	Tourists helped the lawyer.	E
	The judges heard the actors resigned.	The judges heard the actors.	N
	The senator near the lawyer danced.	The lawyer danced.	N
Constituent heuristic	Before the actor slept, the senator ran.	The actor slept.	E
	The lawyer knew that the judges shouted.	The judges shouted.	E
	If the actor slept, the judge saw the artist.	The actor slept.	N
	The lawyers resigned, or the artist slept.	The artist slept.	N

Figure 3: Examples of the 3 heuristics used in the HANS dataset, namely `Lexical Overlap`, `Subsequence`, and `Constituent`. The Figure is adopted from McCoy et al. (2019).

to augment them: first, we determine the part-of-speech of the word with the nltk package. If the word is a noun, we find its synonym from nltk WordNet, replace the word in the original hypothesis with its synonym, and reverse its label; if the word is an adjective or adverb, we find its antonym from WordNet, replace this word and keep its original label. For instance, if the adjective word 'beautiful' is identified to be spurious and the NLI label corresponding to this hypothesis is 1, we augment the sample by replacing 'beautiful' with its antonym 'ugly' and keep the new sample's label to be 1. In this way, the connection between the spurious word and the label is mitigated. Verbs are much more difficult to augment. We experiment with adding a 'not' indicator in front of verbs and assign the new sample with the original connection. However, such a naive approach would only reinforce the connection between the spurious verb and the original label. Therefore, in this project, we do not modify samples with verbs as spurious words.

Iterative Data Augmentation Ideally, for the logistic regression model that we have trained with only the hypothesis as features should have a much lower accuracy after spurious word identification and transformation, which means that the classifier is becoming worse at predicting the NLI label simply based on the hypothesis. However, as we fuse the augmented data with the original dataset, if the proportion of the newly augmented data is too small, then the effect of data augmentation is likely to not be strong enough. But if we are too generous with the spurious word importance score or confidence score bound, we might accidentally create more noises by killing the benign words and hurt the original NLI task. Since LIME would assign each word with a new importance score for new models, we decide to take an iterative approach of augmentation. After each augmentation, we will take out the augmented sample along with its original sample first. Then, we will train a logistic regression model again with the remaining data and perform spurious data identification and augmentation again. As we train a new logistic regression model with a new subset of data, we might identify new spurious words that are not previously caught. This iterative process will stop if the logistic regression classifier reaches an accuracy close to random guess on the remaining data, or if it exceeds a certain number of iterations to avoid infinite loop. If the latter case happens, we will only use the augmented data and its original sample as our data set for the NLI task.

5.2 Adversarial Filtering

In this study, we created an adversarial dataset (Table 1) by applying an Adversarial Filter on the Stanford NLI dataset. Specifically, we first conduct a round of finetuning using the pretrained BERT-base model. With the finetuned model, we conduct inference and collect data whose labels are not correctly classified. We treat such data as the adversarial training set and used such data to conduct another round of finetuning. While it is possible to conduct multiple runs of AFs, we only conduct AF once due to limited computational resources. A natural issue of doing AF on the NLI dataset is that many of the filtered data contain wrong annotations or are ambiguously annotated due to subjectivity. Such data provide no help with mitigating spurious correlation but bring difficult for the model to comprehend the task. We will reiterate the issue of this approach with the result in Section 8.

5.3 Data Fusion

Data fusion is the suggested remedy from the original HANS paper (McCoy et al., 2019). To mitigate spurious correlations in the syntax, McCoy et al. (2019) mix the Multi-Genre NLI dataset with the HANS dataset and conducted a series of leave-one-correlation-out experiments (see (McCoy et al., 2019) for details). In our study, we mimic a similar approach by fusing the Stanford NLI data with HANS training data for finetuning. We constructed finetuning datasets with different proportions of SNLI and HANS dataset (see Section 8.4)e.

6 Model-Centric Remedy

In this section, we adapt the two methodologies in (Belinkov et al., 2019) to train the model adversarially by using a single encoder. To enable adversarial training, we have to modify the original model architecture a bit and replace Bert with distilled Bert for faster training. In the original model, the intermediate representation is directly output by Bert encoder $R_{PH} = Bert(P, H)$. It is then passed to a classifier to get prediction over all possible labels $P = C_{NLI}(R_{PH})$. Now, we learn an intermediate representation for hypothesis and premise separately and combine them $R_{PH} = combine(Bert(P), Bert(H))$, where we combine the representation by concatenating their vectors, difference, and product following (Mou et al., 2016). We train the model by using the loss

$$L_{NLI} = L(C_{NLI}(R_{PH}), y_{label}) \tag{3}$$

where L is the softmax cross entropy loss. It should be noticed that this method performs significantly worse than the original method, achieving around 0.72 accuracies on SNLI test data. This is reasonable because the original method enables the hypothesis and premise to do cross attention throughout layers in the Bert model. However, in the new architecture, such connections between premise and hypothesis are only made at the few last layers.

6.1 AdvCls: Adversarial Classifier

Our first method, AdvCls follows a common adversarial training paradigm by adding an additional classifier C_H to the model. C_H is used to predict label given only hypothesis $P_l = C_H(Bert(H))$. And the loss function is changed to

$$L = L_{NLI} + \lambda_H LAdv$$
$$L_{Adv} = L(C_H(GRL_{\lambda_a}(Bert(H)), y_{label})$$

where GRL is the gradient reversal layer (Ganin and Lempitsky, 2014) with backward coefficient set to λ_a. In the forward pass, GRL output its input without any modification. While in the backward pass during backpropagation it multiplies the gradient it received by $-\lambda_a$. In a word, the model tries to reduce the correlation between the embedding of hypothesis and labels while optimizing the loss L_{NLI}. See Figure 4 for the model architecture.

6.2 AdvDat: Adversarial Training Data

Our second model AdvDat uses an unchanged general model but is trained with perturbed data. For a fraction of pairs in our training data, we replace (P, H) with (P', H) where P' is randomly sampled from the training data. For these instances, we similarly reverse the gradient through the encoder of hypothesis and block gradient through the encoder of premise by using two gradient reversal layers. The loss now becomes

$$L_{Adv} = L(C_{NLI}([GRL_0(bert(P'));$$
$$GRL_{\lambda_a}(bert(H))]), y)$$

where GRL_0 achieve gradient blocking by setting backward coefficient to 0. At the same time, GRL_{λ_a} work as before with backward coefficient λ_a. The total loss we are optimizing is then

$$L = (1 - \lambda_{Rand})L_{NLI} + \lambda_{Rand}L_{Adv} \tag{4}$$

where the hyper-parameter $\lambda_{Rand} \in [0, 1]$ specify what fraction of training samples have premise perturbed. This method essentially tries to penalize the model for correctly predicting y given an uninformative premise.

7 Experiment Setup

7.1 Model

The model we use for our NLI task is the BERT model available at Hugging Face (`https://huggingface.co/bert-base-uncased`). BERT is a transformer-based model pretrained with the masked language modeling (MLM) and next sentence prediction (NSP) task. It is a popular architecture that has been proven to be superior in

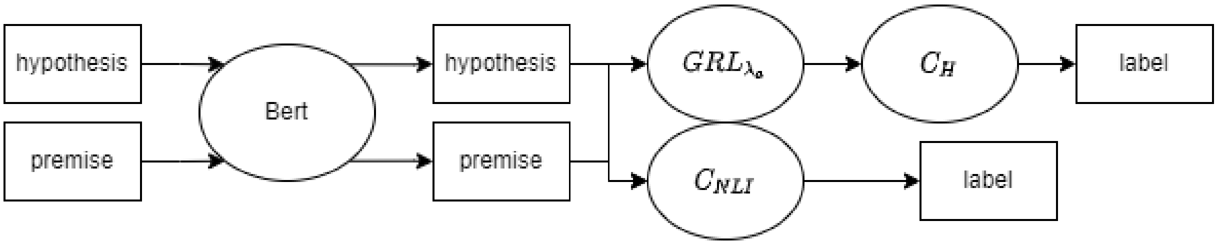

Figure 4: Demonstration of the `AdvCls` model architecture.

Method	Baseline Acc	Val Acc before Aug	Val Acc after Aug	No. Augmented data
Iter.Aug-(n=50k)	66.59	73.14	69.00	10756
Iter.Aug-(n=5k)	66.53	69.09	65.30	448
Aug-(n=50k, train with verb)	66	72.99	73.37	-

Table 2: Data augmentation results.

downstream tasks such as NLI, classification, NER etc. Due to computational constraints, we choose to use the BERT-base model, which uses 12 layers of transformers block and 12 self-attention heads, with around 110M trainable parameters. The uncased version of BERT we use takes in uncased texts and strips out accent markers. We choose the below hyperparameters for the model: `optimizer = "AdamW"; lr_rate = 2e-4, epochs = 10, batch_size = 16; training size = 50000; validation size = 10000.`

7.2 Dataset

We used four different datasets for training and testing to see how well our model generalizes. The SNLI dataset is one of the most commonly used NLI datasets. It has around 570k sentence pairs labeled as entailment, contradiction, and neutral (Bowman et al., 2015). We also include two of its variations. MNLI is a dataset containing around 433k sentence pairs modeled on NLI but it covers a wide range of genres of spoken and written text, such as fiction, letters, and telephone speech (Williams et al., 2018). ANLI contains adversarial human-and-model-in-the-loop data that are difficult to be trained with nie-etal-2020-adversarial. HANS helps test specific hypotheses about invalid heuristics that NLI models are likely to learn well (McCoy et al., 2019). All three of these are used to test the robustness issue of NLI.

8 Experiments

8.1 Evaluation of spurious correlation

The performance of the logistic model trained on the original data and the augmented data is shown in Table 2. After the iterative data augmentation process, we can see that both the cross-validated validation accuracy drops (for the task of predicting NLI labels based on only hypothesis), indicating that first, we are successful in removing the correlation between the spurious words and labels; second, removing the spurious correlation in the train data also helps remove the spurious correlation in the validation data. This confirms that there should be some consistent spurious patterns in the SNLI dataset. Also, we see that if we simply augment spurious verbs by adding 'not' in the font, the validation accuracy actually improves, which means that such a spurious correlation is strengthened. This proposes open questions for researchers as to how to effectively augment verbs for the task.

We also include here a few illustrations of the confidence and importance score generated via LIME (See Appendix for figures). In the first LIME example (Figure 6), the word 'for' is clearly not relevant to the NLI task, however, it gives a very high probability of 0.81 predicting the hypothesis to have the contradiction (non-entailment) label. In the second LIME example ((Figure 7)), the prediction probability is close to the random guessing accuracy of 0.66, and none of the words in the sentence stand out as particularly important for label prediction. This should be the ideal case that we are after if all spurious correlation is removed. In the third LIME example (Figure 8), the word 'outdoors' is having a very high importance score of 0.12 that leads to a high probability of 0.96 to predict the label as contradiction. This is cross-validated in the graph of the top spurious words identified, where the y-axis represents the coeffi-

Figure 5: Training and validation curves of the BERT-base model on AF filtered SNLI dataset.

cient of each word for the logistic regression model. We notice that these words occur frequently in the hypothesis with a contradiction label, which may mislead the model to assign contradiction as long as the sentence contains such words.

8.2 Evaluation of NLI with data augmentation

Due to computational constraints, we only augment the data on around 50k of the SNLI hypothesis. Around 5k of which is identified to contain spurious correlations, and in total the augmented data and original data used for training after de-biasing are 10756. From the result in Table 1, we do not observe an increase in test accuracy for the HANS dataset after data augmentation–its performance is 0.4997. If taking a random guess, we get an accuracy of 0.4996 on HANS-val if trained with SNLI-train. Augmentation only improves its performance by a negligible amount. Also, training with SNLI-train hypothesis will yield a nearly random guessing accuracy on SNLI-test and a slightly worse than random guessing accuracy on HANS-val. There are a few possible reasons for this: first, the distribution of the binary label is not the same for SNLI and HANS. In SNLI the ratio of Entailment to Non-entailment is around 0.66, while in HANS the ratio is around 0.5. This might hurt the HANS-val accuracy that is supposed to be rising if the model is trained with SNLI-train. Second, it is possible that the failure in generalization might not entirely stem from spurious correlation, but from some other inherent difference in the language features in the two datasets.

8.3 Adversarial Filtering

In this section, we discuss the results of applying AF on the Stanford NLI dataset. As discussed in Section 5.2, we leverage a finetuned BERT-base classifier to filter hard-to-learn data, which we then use to conduct another round of finetuning. From the result of Table 1, we see that applying AF in the Stanford NLI dataset harms the task tremendously. From the training curves shown in Figure 5, we see that the model still is able to fit the training dataset (top right plot). However, it does not generalize (bottom right plot), suggesting that the distribution of data after AF filtering is drastically different from the overall distribution of the data in the SNLI dataset. We suspect the cause of such distribution discrepancy is largely due to error and subjectivity in the annotations.

8.4 Data Fusion

Results from Table 1 show the effectiveness of data fusion in mitigating spurious cues. Our empirical result shows that the 50-50 fusion strategy works the best where the model is able to achieve reasonable performances on both SNLI and HANS data. Finetuning with a 50000 dataset consists of 50% SNLI data and 50% HANS data, the resulting BERT-base model is able to get 0.8328 acc on SNLI (0.8457 for training with SNLI alone) and 0.8425 acc on HANS (0.9999 on training with HANS alone). Further, Figure 11 shows that fuse dataset can efficiently reduce the loss and improve the validation accuracy for the NLI tasks, further proving the efficacy of finetuning with multiple datasets of the same task.

In fact, pretraining language models using various datasets of the same task have been proven to be a data-efficient way of training even for LLMs and it is a vital component for preventing models from learning spurious cues in single dataset (Chung et al., 2022; Wang et al., 2022).

8.5 Adversarial Training

For both $AdvDat$ and $AdvCls$ we test the performance of models with different configurations of the two involved hyper-parameters. See Table 3 for the results. For $AdvCls$, the performance of the hypothesis adversarial classifier is highly sensitive to the two hyper-parameters. When either of λ_{Adv} or λ_H is higher than 0.4, the accuracy of hypothesis only classifier suddenly drop below 0.33 as expected. However, adversarial training seems to have little effect when the two hyper-parameters are both set to values below 0.2. For $AdvDat$, the performance of the model is also closely related to the two hyper-parameters λ_{Rand} and λ_a, the effect of adversarial training become obvious when both of them are set to values higher than 0.2. It should be pointed out that for both methods, the accuracy of NLI classifier (C_{NLI}) dropped significantly compared to the baseline where the classifier is trained normally without adversarial training (with accuracy around 0.74). For $AdvCls$, We also observed the phenomenon that retraining a classifier on frozen hypothesis representations boosts accuracy close to the fully trained hypothesis-only baseline. This phenomenon is also observed by other researchers without proper explanation. (Belinkov et al., 2019) argues that even a frozen encoder with random weight is able to capture spurious correlation between hypothesis and labels, as a hypothesis-only classifier trained on that performs fairly well. They suspect that this is caused by the fact that word embeddings were not updated during training and they contain significant information that propagates even through a random encoder. For both methods, the models give a by-chance prediction on HANS dataset, which suggests that there is no improvement in generalization. Overall, we have to summarize that the effectiveness of the adversarial training methods tried is unsatisfying.

9 Conclusion

In this study, we unveiled the existence of spurious correlations in various NLI datasets including SNLI, MNLI, ANLI. We built connections between spurious correlation and models' incapable to generalization. Further, we attempted multiple remedies such as `Data Augmentation`, `Adversarial Filtering`, `Data Fusion`, and `Adversarial Training`, to prevent models from picking up the spurious cues. We have shown that preventing models from taking the shortcut is a challenging task and only data fusion can mitigate the issue. Yet, we are unsure whether the model also learned spurious cues in the fused data. Therefore, we justify the necessity of training language models with excessive data as they may serve as an effective measure of preventing models from learning spurious correlations. In addition, we also justify the current trend of building in-context learning models and focus on building generation models. The powerful LLMs of the in-context learning paradigm enable researchers to approach the issue of mitigating spurious cues with much more natural and effective measures such as asking the LLM to provide rationale in classification tasks like NLI.

Acknowledgements

We show our utmost gratitude to the instructor, Professor Pratik Chaudhari, and the course staff of ESE 546 at the University of Pennsylvania for a great semester. The course deepened our understanding of deep learning and broadened our knowledge in areas such as optimization and generalization in machine learning. Enlightened by this course, all of us wish to continue conducting research, both applied and theoretical, using deep learning technologies.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Yonatan Belinkov, Adam Poliak, Stuart M. Shieber, Benjamin Van Durme, and Alexander M. Rush. 2019. On adversarial removal of hypothesis-only bias in natural language inference.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yahui Chen. 2015. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao,

Method	λ_H	λ_a	C_H Val Acc	C_{NLI} Val Acc	Baseline Val Acc
	0.2	0.2	0.628	0.705	0.742
	0.2	0.4	0.637	0.695	0.742
	0.2	0.8	0.298	0.697	0.742
$AdvCls$	0.4	0.2	0.624	0.716	0.742
	0.4	0.4	0.229	0.683	0.742
	0.4	0.8	0.135	0.673	0.742
	0.8	0.2	0.231	0.705	0.742
	0.8	0.4	0.127	0.677	0.742
	0.8	0.8	0.120	0.677	0.742
	λ_{Rand}	λ_a	P' Val Acc	P Val Acc	P Baseline Val Acc
	0.2	0.2	0.636	0.624	0.742
	0.2	0.4	0.626	0.624	0.742
	0.2	0.8	0.602	0.610	0.742
$AdvDat$	0.4	0.2	0.643	0.626	0.742
	0.4	0.4	0.610	0.634	0.742
	0.4	0.8	0.645	0.601	0.742
	0.6	0.2	0.614	0.597	0.742
	0.6	0.4	0.580	0.595	0.742
	0.6	0.8	0.534	0.610	0.742

Table 3: Experiment results of the BERT-base model on various NLI datasets. Data name with postfix `hypothesis` means only the `hypothesis` is provided to the model during training. Data name with postfix `premise` means only the `premise` is provided to the model during training.

Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain. Association for Computational Linguistics.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. 2020. Underspecification presents challenges for credibility in modern machine learning.

Anupam Datta. Gender bias in neural natural language processing. *Logic, Language, and Security*, page 189.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

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

Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 1681–1691.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Michael I Jordan. 1997. Serial order: A parallel distributed processing approach. In *Advances in psychology*, volume 121, pages 471–495. Elsevier.

Michael Kearns. 1988. Thoughts on hypothesis boosting. *Unpublished manuscript*, 45:105.

Michael Kearns and Leslie Valiant. 1994. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint arXiv:2201.05955*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

Emily McMilin. 2022. Selection bias induced spurious correlations in large language models.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. 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.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun

Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

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

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020b. An empirical study on robustness to spurious correlations using pre-trained language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: generalization via declarative instructions on 1600+ tasks. EMNLP.

Zhao Wang and Aron Culotta. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14024–14031.

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

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

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Appendix

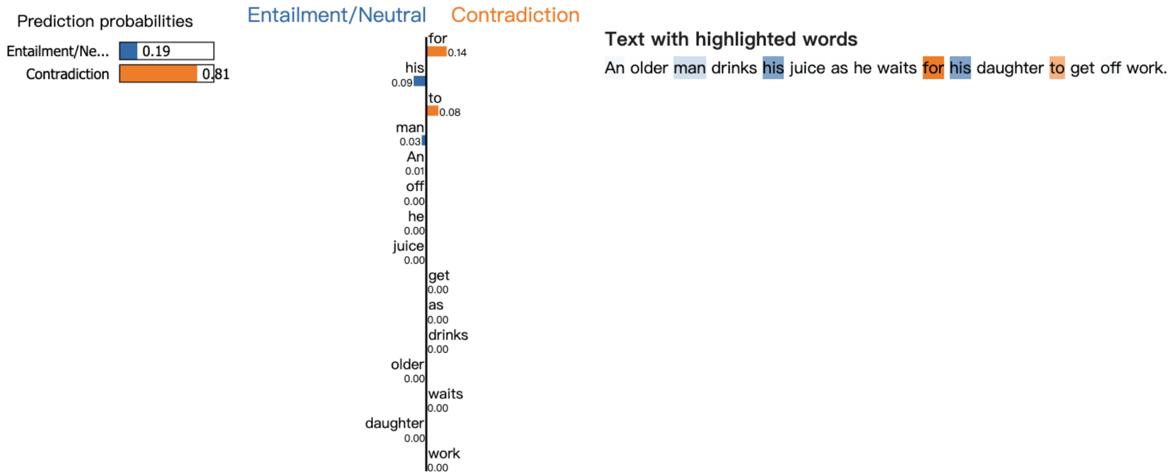

Figure 6: LIME Example 1

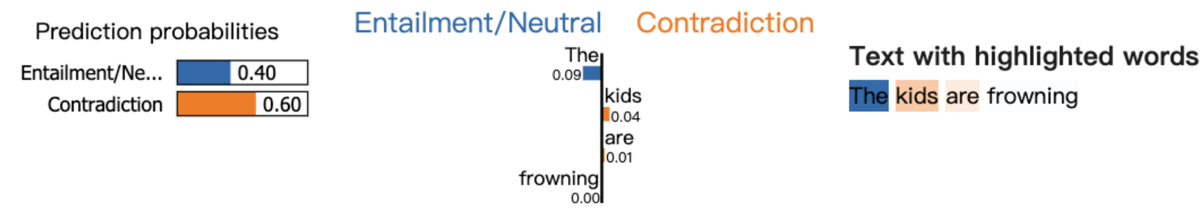

Figure 7: LIME Example 2

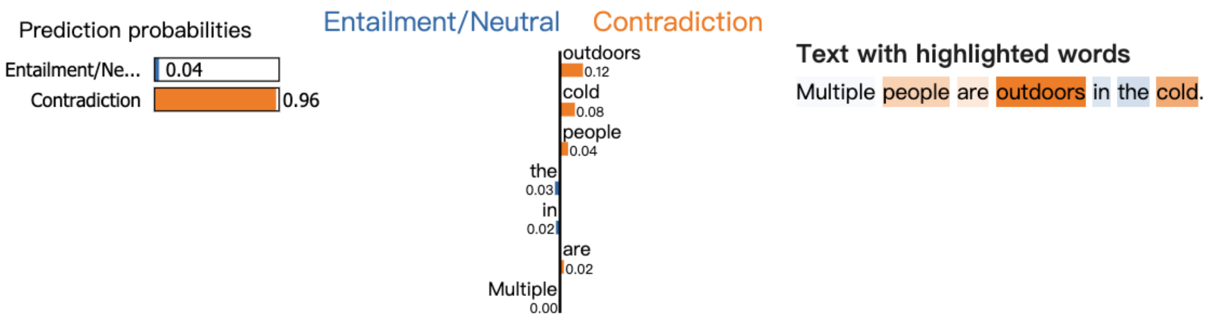

Figure 8: LIME Example 3

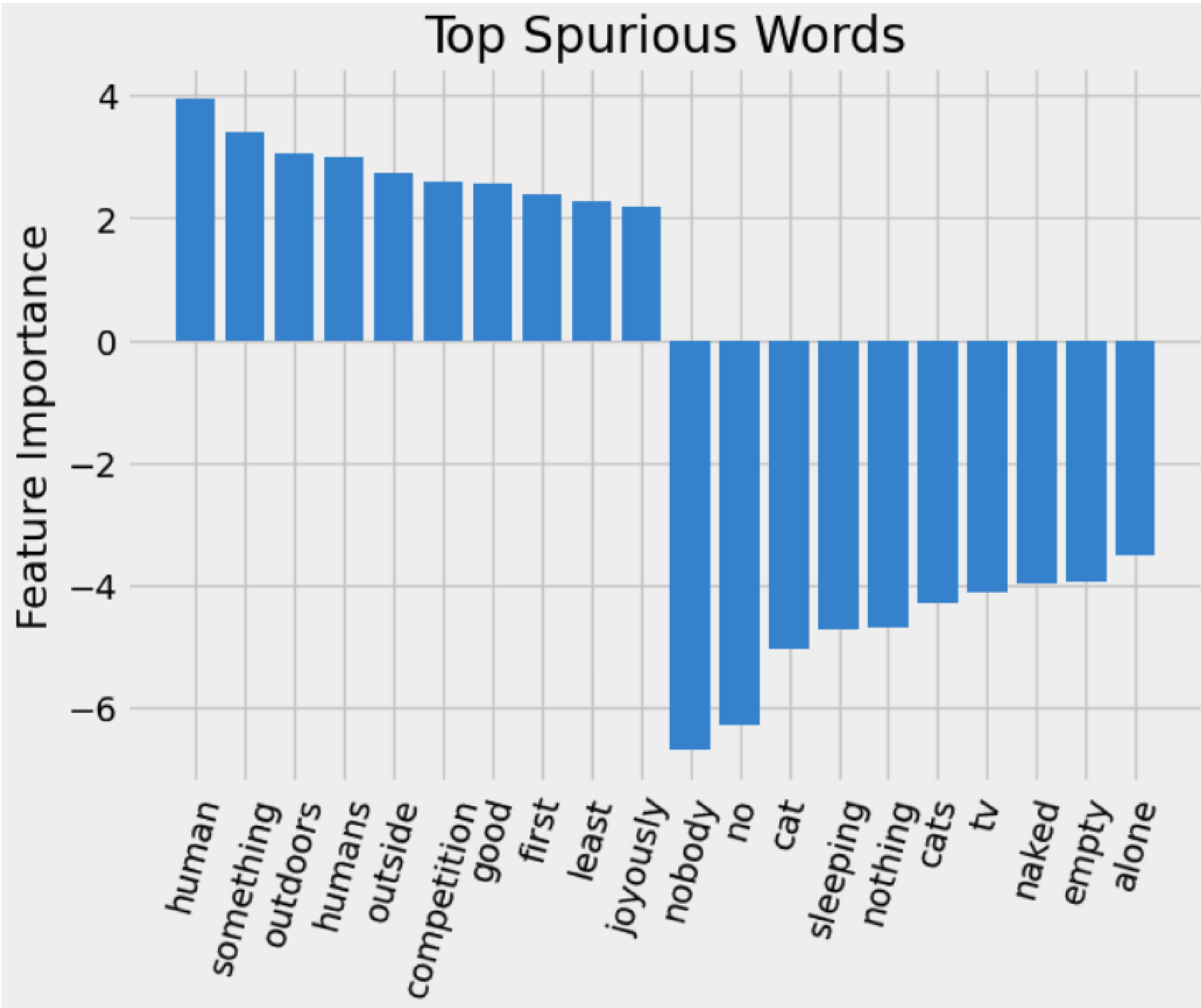

Figure 9: Top Spurious Words Identified (in the first 67k SNLI data)

Figure 10: The training curves of BERT-base model using various forms of the SNLI dataset. `hans_frac_(frac)` represents the percentage of data that contains data from the HANS dataset.

Figure 11: The training curves of BERT-base model using various forms of the Multi-Genre NLI dataset.