We Need to Talk About Robustness to Adversarial Attacks While Removing Spurious Dataset Biases

by

Bhavdeep Singh Sachdeva

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2021 by the
Graduate Supervisory Committee:

Chitta Baral, Chair
Huan Liu
Yezhou Yang

ARIZONA STATE UNIVERSITY

August 2021

ABSTRACT

Machine learning models can pick up biases and spurious correlations from training data and projects and amplify these biases during inference, thus posing significant challenges in real-world settings. One approach to mitigating this is a class of methods that can identify filter out bias-inducing samples from the training datasets to force models to avoid being exposed to biases. However, the filtering leads to a considerable wastage of resources as most of the dataset created is discarded as biased. This work deals with avoiding the wastage of resources by identifying and quantifying the biases. I further elaborate on the implications of dataset filtering on robustness (to adversarial attacks) and generalization (to out-of-distribution samples). The findings suggest that while dataset filtering does help to improve OOD(Out-Of-Distribution) generalization, it has a significant negative impact on robustness to adversarial attacks. It also shows that transforming bias-inducing samples into adversarial samples (instead of eliminating them from the dataset) can significantly boost robustness without sacrificing generalization.

TABLE OF CONTENTS

Page

LIST OF TABLES iv

LIST OF FIGURES v

CHAPTER

1 INTRODUCTION 1

2 BACKGROUND AND RELATED WORK 5

2.1 Machine Learning Bias 5

2.2 Dataset Pruning 7

2.3 Adversarial Filtering 8

2.4 Robustness and Generalization Evaluation 14

2.4.1 Generalization 15

2.4.2 Independently and Identically Distributed Dataset 19

2.4.3 Out of Domain Datasets 20

2.4.4 Robustness 26

2.5 Neural Language Models 37

2.5.1 BERT 37

2.6 Dataset Cartography 38

3 EXPERIMENTAL SETUP 41

3.1 Identifying biased Data Samples 41

3.2 Evaluation Metrics 41

3.3 In domain accuracies 43

3.4 Robustness 44

3.4.1 Robustness Scores 44

3.4.2 Model free Robustness 44

3.5 Generalization 45

CHAPTER Page

4 RESULTS 47

4.1 In domain accuracies 47

4.2 Robustness 48

4.3 Generalization 49

4.4 SNLI-R 51

4.5 Data maps 53

REFERENCES 58

LIST OF TABLES

Table Page

2.1 Sentences Generated Using Eda. Sr: Synonym Replacement. Ri: Random Insertion. Rs: Random Swap. Rd: Random Deletion (Wei and Zou, 2019) 32

2.2 Adversarial Examples from B.A.E. 37

4.1 Snli Good Data I.I.D. Accuracy with Bert Model 48

4.2 Snli Robustness Scores for Different Quantities of Good and Bad Data. 51

4.3 Model Free Text Attack 51

4.4 Zero-shot Snli Accuracy on Ood Datasets with Roberta Based Trained Model. 52

4.5 Snli-r Helps Improve the Robustness. 54

4.6 Snli-r Helps Improve the Generalization. 54

LIST OF FIGURES

Figure Page

2.1 Spurious Correlations 6
2.2 A Few Examples from Winogrande Dataset Sakaguchi *et al.* (2020) Showing Selected Biased Samples Using Aflite 11
2.3 Roberta's Pre-computed Embedding Visualized in 2d and 1d 13
2.4 Wastage of resources with pruning 14
2.5 A Few Examples from the Snli Dataset (Bowman *et al.*, 2015a) 21
2.6 One example for each miss-classified error category. 23
2.7 The Process of Generating Adversarial Samples Through Human and Model in the Loop Strategy Nie *et al.* (2020) 24
2.8 Augmentation Is Done on the Train Set. 27
2.9 Illustration of Clare. 28
2.10 High Level Description of Transformer Encoder. 38
2.11 Data Map Showing Different Types of Samples Present in Winogrande Dataset. 40
3.1 Procedure for Calculating in Domain Accuracy. 43
3.2 Procedure for Calculating Average Robustness. 45
3.3 Procedure for Model Free Text Attack. 46
4.1 Visualization for Model Free Robustness Accuracy and Robustness Scores. 50
4.2 Visualization for Zero-shot Snli Accuracy on Ood Datasets 52
4.3 Repairing the Bad or Biased Samples to Recover Robustness and Avoid Wastage of Resources. 53
4.4 Snli-r Helps Improve the Robustness. 54
4.5 Snli-r Helps Improve the O.O.D. Genearlization. 55

Figure Page

4.6 Data Map for Snli Dataset Showing Unique Hard to Learn, Easy to Predict and Ambiguous Samples. 56

4.7 Data Map for "Good" Part of Snli Dataset as Identified by Aflite. 56

4.8 Data Map for the Adversaries Generated for the "Bad" Part of Snli Dataset as Identified by Aflite. 57

Chapter 1

INTRODUCTION

Machine learning models are universal function approximators (Hornik *et al.*, 1989), and in theory, with enough data (Hoeffding, 1963), can generalize to diverse data distributions. However real-world data, however large, tends to be biased (Torralba and Efros, 2011; Wang *et al.*, 2019; Jiang and Nachum, 2020). When trained on biased data, models can replicate these biases during prediction Bolukbasi *et al.* (2016), and in many cases also amplify biases (Zhao *et al.*, 2017; Bender *et al.*, 2021). These biases can be attributed to several causes such as historical, social, representational, algorithmic, observer and annotator bias and so on (Mehrabi *et al.*, 2019).

In natural language processing, Neural language models, empowered by recent advances in machine learning have achieved human-level performances in various held out datasets such as SQuAD Rajpurkar *et al.* (2016) and SNLIBowman *et al.* (2015b). However, similar model performance is not reflected in "data in the wild", Hendrycks and Dietterich (2019); Eykholt *et al.* (2018); Jia and Liang (2017a) i.e, Out of Distribution (OOD) and Adversarial Datasets. A growing number of studies are now raising concerns about using accuracy on the held out set as the single evaluation metric due to the potential model performance inflation Bras *et al.* (2020a); Sakaguchi *et al.* (2020) accuracy produces; accuracy's large scale negative impact in overestimating capabilities of AI systems hinders the adoption of AI in many promising applications.

A key source of the model performance inflation with accuracy on held out dataset is spurious bias, the unintended correlation between model input and output Torralba and Efros (2011). For e.g., crowdworkers created a large number of contradiction samples with the word 'NOT' in the hypothesis while creating SNLI. This has given rise to spurious bias and the model trained on these datasets tend to label any sample with the word 'NOT' in hypothesis as contradiction Gururangan *et al.* (2018). Held out datasets, being part of I.I.D, carry the same bias as the training dataset, so the model succeeds in solving the heldout dataset, even with its misconception of the word 'NOT'. However, as expected, model performance drops for "data in the wild" because of overreliance of models on these spurious biases. Biases manifest as spurious correlations between input features and output variables (Poliak *et al.*, 2018; Gururangan *et al.*, 2018; Kaushik and Lipton, 2018), thus posing a significant obstacle in the reliability of machine learning models in the real-world setting.

Solutions designed to mitigate the risk of spurious correlations can be categorized as:

1. data augmentation – rule-based, adversarial, or counterfactual augmentation of existing datasets to increase diversity,
2. *model de-biasing* – algorithms that seek to exploit prior knowledge of existing biases in order to learn robust models,
3. *dataset filtering* – algorithmic methods that seek to filter bias-inducing samples from training datasets.

Several approaches have been proposed to remove data samples containing spurious biases Bras *et al.* (2020a); Wu *et al.* (2020); Utama *et al.* (2020); Li and Vasconcelos (2019). The goal in all these works is to improve the zeroshot OOD generalization performance on several independently created datasets where the spurious biases in the SNLI training set are less likely to hold. However, to our surprise, none of these works talk about robustness to adversarial attack Jin *et al.* (2019a)Morris *et al.* (2020) while removing dataset biases. *We want our model to generalize better but not at the cost of losing robustness to adversarial attacks, another important capability essen-*

tial for models to succeed in real world. We argue that such a study is important, specifically when the correlation between robustness to adversarial attack and OOD generalization is unknown.

When we learn from a large set of very similar samples, the risk of spurious bias increases which provides shortcuts to models that subsequently negatively impact generalization [1]. On the other hand, a set of very similar samples are still distinct among each other and may represent word/phrase perturbed forms of others. Learning from those samples makes models stay aware of various perturbations and increase its robustness to adversarial attacks which are essentially perturbations. This intuitive understanding compels us further to conduct this study regarding the effect of spurious bias on robustness to adversarial attacks, along with OOD generalization.

In this work, we seek to study the effect of dataset filtering methods such as AFLite (Bras *et al.*, 2020b) on two metrics – robustness to adversarial attacks (**Adv-Rob**) and generalization to out-of distribution samples (**OOD-Gen**). This study is done in comparison with vanilla model training, data augmentation, and model debiasing, for the task of natural language inference (NLI).

In summary, we make the following contributions while addressing the aforementioned gap in model evaluation.

- We show that deletion of datasets containing spurious biases hurts model robustness to adversarial attacks, advocating to conduct robustness study along with generalization while analyzing/removing spurious biases.
- Even though the addition of data samples containing spurious biases improves robustness, it hurts generalization, and inflates accuracy on the held out set. So,

[1] Henceforth generalization implies zeroshot OOD generalization, and robustness implies robustness to adversarial attack

we propose a data augmentation mechanism using adversarial transformation and show that our mechanism results in higher boost in robustness (than addition of biased samples without transformation), with competitive generalization performance and reduced inflation in model performance. In this process we also address the issue of data deletion prevalent in most approaches to remove spurious bias; deletion of data samples wastes the resources invested in their creation.

- We propose a framework to repair legacy datasets and revamp existing benchmarks which are not really solved (in contrast to inflated leaderboard performance). We demonstrate our framework by proposing the RSNLI (repaired SNLI) dataset.

- Our data quality analysis shows that RSNLI has higher diversity and contains lesser number of artifacts such as word overlap, sentence length variation across labels than SNLI, etc.; this justifies the efficacy of our framework to revamp legacy datasets.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Machine Learning Bias

The term ”bias” in machine learning, sometimes referred to as algorithmic bias or AI bias, was introduced by (Mitchell, 1980). As stated, it means ”any basis for choosing one generalization [hypothesis] over another, other than strict consistency with the observed training instances.” Simply put, machine learning bias is the phenomenon when the model chooses one hypothesis over another due to erroneous assumptions. AI bias can be introduced at any stage of the machine learning pipeline. The basis of these erroneous assumptions helps us classify these biases into various categories. For example, a race-agnostic loan approval would be undesirable; therefore, if a model predicted based on race for such a scenario, it would have a racial bias. Another example is exclusion bias, where a fraction of the real data is not represented either knowingly or unknowingly.

Spurious bias, in short, is the unintended correlation between the input and the output. See figure 2.1 one example from Vigen (2015), a collection of spurious correlations. This example shows a high correlation(99.79%) between the ”US spending on science, space, and technology” and ”suicides by hanging, strangulation, and suffocation” for a given duration of time. Even though common sense dictates that we should not relate these two entities if a model trained on a set of features(including the US spending on science) to predict the suicides in the same time period. The model will likely depend on only this feature to make the prediction.

When we do not know the actual ”cause” of a result, a ”leakage” from any direction in the data will lead the model to gold prediction. These unwanted correlations are the primary reason for the recent success of machine learning models, where we solve a dataset but fail at the task. For example, numerical reasoning in natural language is still a challenge (Mishra *et al.*, 2020d) even though we have an accuracy of 90% on datasets like DROP (Dua *et al.*, 2019), which are not easy for an average human.

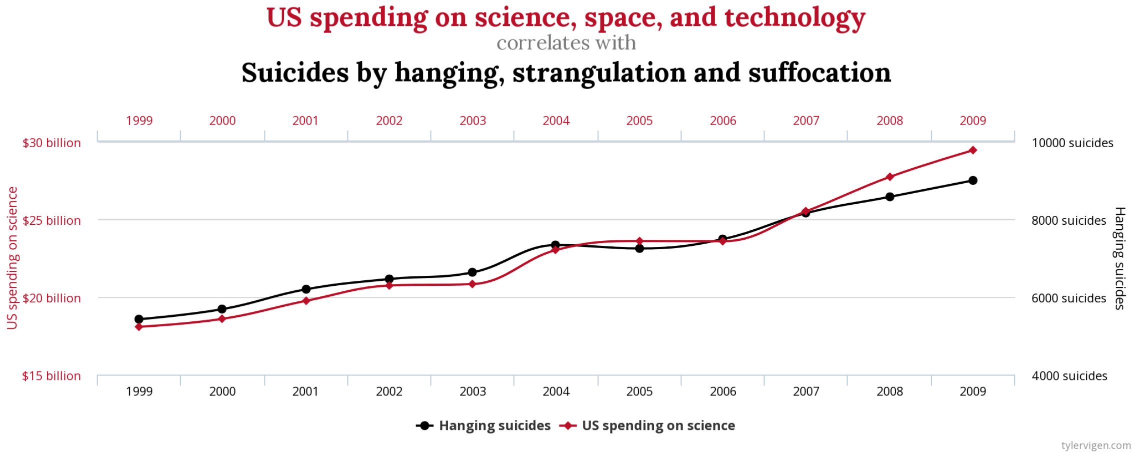


Figure 2.1: Spurious Correlations

Several studies Gururangan *et al.* (2018); Poliak *et al.* (2018); Mishra *et al.* (2020b); Bras *et al.* (2020a); Kaushik and Lipton (2018) have identified that language models are overfitting to various benchmark idiosyncrasies and do not truly learn the underlying task. Spurious bias, the unintended correlation between model input and output, is the carrier of these benchmark idiosyncrasies. In case of NLI, the association of the word 'not' with the contradiction label is an example of spurious bias. Spurious bias has even higher implications as it gives rise to overestimation of AI capabilities Bras *et al.* (2020a); Hendrycks *et al.* (2020b); Mishra *et al.* (2020a). This further hinders the adoption of AI systems in safety critical applications like healthcare. While inductive bias is wanted in machine learning, spurious bias is unwanted and needs to be avoided. Due to the observation on spurious bias, data quality Rogers (2021); Sambasivan *et al.* (2021); Mishra *et al.* (2020c) is also getting attention recently similar to the attention model development has received over past several years. Our primary objective in this work is to get rid of the spurious bias in data. In contrast to other works, we are also interested in studying several different aspects such as generalization, robustness, model accuracy along with spurious bias.

2.2 Dataset Pruning

Dataset pruning removes strategically selected samples from the dataset to improve the model's performance. The concept of dataset pruning was first defined in (Angelova *et al.*, 2005) as the process of noise removal. In the same article, the measure of success of these models was defined by SVC and AdaBoost algorithm. In the modern context, noise can be used as an umbrella term for annotator artifacts and various other kinds of biases that negatively affect the performance of a model. Dataset pruning has been performed across various works to identify the representative samples that can replace the bigger dataset. Smaller dataset requires lesser computation and time to train model and are also easier to control quality. Also, the pruning process can unfold various characteristics of the necessary samples compared to the redundant samples. Dataset Distillation Wang *et al.* (2018b) shows that ten synthetically created samples can potentially replace the MNIST dataset. Adversarial Filtering was first proposed by (Zellers *et al.*, 2018) for constructing the SWAG question-answering dataset. Adversarial Filtering iteratively trains an ensemble of classifiers to identify and easy samples and replace them with adversarial samples. Versions of AF have also been used to construct challenging datasets Zellers *et al.* (2019b); Bhagavatula *et al.* (2020) or to prevent models from answering multiple-choice questions by elimination Zellers *et al.* (2019a); Fang *et al.* (2020). These methods adversarially perturb instances and require re-training of the model at every iteration of dataset filtering. AFLite (Sakaguchi *et al.*, 2020; Bras *et al.*, 2020b) does not rely on curated strategies or rules for generating perturbations and does not need model re-training. Other approaches such as REPAIR (Li and Vasconcelos, 2019), RESOUND (Li *et al.*, 2018), (Sagawa *et al.*, 2020) suggest re-sampling or sub-sampling of datasets reduce biases.

2.3 Adversarial Filtering

The concept of adversarial filtering was introduced in (Zellers *et al.*, 2018) to construct the SWAG dataset. The corpus consisted of 113k multiple-choice questions derived from ActivityNet Captions (Krishna *et al.*, 2017). The goal in the conception of the dataset was to achieve diversity while minimizing the annotation artifacts, conditional stylistic patterns such as sentence similarity and word preference biases.

What is an adversarial dataset? If a model M is trained on a dataset, the dataset is said to be adversarial if it will not generalize even if evaluated on the same distribution. The dataset is adversarial for a model M if we expect high empirical error I overall leave-one-out train/test splits (Vapnik, 2013)

$$I(\mathcal{D}, M) = \frac{1}{N} \sum_{i=1}^{N} L\left(M_{\theta_i^\star}, \{(x_i, y_i)\}\right) \tag{2.1}$$

$$\text{where } \theta_i^\star = \underset{\theta}{\text{argmin}} L\left(M_\theta, \mathcal{D} \backslash \left\{(x_i, y_i)\right\}\right), \tag{2.2}$$

How to generate an adversarial dataset? Let us assume we have N contexts for a given problem statement, each having one positive example $\left(x_i^+, 1\right) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and the $\mathcal{Y}$ is the output space. Similarly, we have a range of negative examples corresponding to each context $\left(x_{i,j}^-, 0\right) \in \mathcal{X} \times \mathcal{Y}$, where $1 \leq j \leq N^-$ for each i. The goal of adversarial filtering in this setting is to identify a subset of negative examples for each context instance i to a minimal set $k \ll N^-$. If we for the sake of this argument identify the returned subset as $\mathcal{A}$ the filtered dataset would be :

$$\mathcal{D}^{AF} = \left\{ (x_i, 1), \left\{ \left(x^-_{i,j}, 0\right) \right\}_{j \in \mathcal{A}_i} \right\}_{1 \leq i \leq N} \tag{2.3}$$

Unfortunately, this problem is non-tractable. The adversarial filtering algorithm as shown in Algorithm 1 is proposed solution for this problem. The idea is to split the existing dataset iteratively and train model f to remove the sample that model classifies correctly and replace them with the samples which the model is not able to identify correctly.

Algorithm 1: Adversarial Filtering (Zellers *et al.*, 2018)

```
Adversarial filtering (AF) of negative samples.
Inputs: $N^{\text{easy}} = 2$ for refining a population of $N^- = 1023$ negative examples to $k = 9$
while convergence not reached do
    - Split the dataset $\mathcal{D}$ randomly up into training and testing portions $\mathcal{D}^{tr}$ and $\mathcal{D}^{te}$.
    - Optimize a model $f_\theta$ on $\mathcal{D}^{tr}$. for index $i$ in $\mathcal{D}^{te}$ do
        - Identify easy indices: $\mathcal{A}_i^{\text{easy}} = \left\{ j \in \mathcal{A}_i : f_\theta\left(x_i^+\right) > f_\theta\left(x_{i,j}^-\right) \right\}$
        - Replace $N^{\text{easy}}$ easy indices $j \in \mathcal{A}_i^{\text{easy}}$ with adversarial indices $k \notin \mathcal{A}_i$ satisfying $f_\theta\left(x_{i,k}^-\right) > f_\theta\left(x_{i,j}^-\right)$
    end for
end while
```

Lightweight adversarial filtering(AFLite) (Sakaguchi *et al.*, 2020) is an improved version of adversarial filtering(AF). It is an improvement in two key aspects.

- avoiding overgeneration of data thus more generally applicable.

- it is lightweight as it does not require the model to be retrained at each iteration of the filtering.

The problem with the generation of samples at this point is the risk of distributional bias. The model identifies the artificial samples as separate entities in distributional bias and learns to solve them separately. To operate filtering by training neural language models like BERT is an extensive computation process. The AFLite works with a slightly different setting. Instead of using models like BERT, it uses an ensemble of linear and basic non-linear models such as logistic regression and support vector machines(SVMs). In addition to that, rather than using manually selected lexical features, the model uses RoBERTa based sentence embeddings as features. The complete algorithm is shown in Algorithm 2.

To demonstrate the identified bias using the AFLite algorithm, we see a few examples in figure 2.2 from the Winogrande dataset created using the AFLite algorithm from the Winogrande schema challenge. The Winogrande task is formulated as a fill-in-the-blank question with binary answers. The goal is to choose the right option for a given sentence which requires commonsense reasoning. In figure 2.2, the first two examples, there is an undesirable correlation between the sentiment between the answer option and the target pronoun. Given the correlation, the problem can be solved by exploiting the polarity pattern (positive or negative) in the sentence. The model would not be using any extra commonsense knowledge, which is required part of the task's goal. One more thing to be noted here is that the bias is present at the structural level and not at the token level, which is hard to detect using heuristics such as PMI-filtering.

Are these filtering algorithms really working? To answer this question, we

	Twin sentences	Options (**answer**)
✗	The monkey loved to play with the balls but ignored the blocks because he found **them** *exciting*.	**balls** / blocks
	The monkey loved to play with the balls but ignored the blocks because he found **them** *dull*.	balls / **blocks**
✗	William could only climb begginner walls while Jason climbed advanced ones because **he** was very *weak*.	**William** / Jason
	William could only climb begginner walls while Jason climbed advanced ones because **he** was very *strong*.	William / **Jason**
✓	Robert woke up at 9:00am while Samuel woke up at 6:00am, so **he** had *less* time to get ready for school.	**Robert** / Samuel
	Robert woke up at 9:00am while Samuel woke up at 6:00am, so **he** had *more* time to get ready for school.	Robert / **Samuel**
✓	The child was screaming after the baby bottle and toy fell. Since the child was *hungry*, **it** stopped his crying.	**baby bottle** / toy
	The child was screaming after the baby bottle and toy fell. Since the child was *full*, **it** stopped his crying.	baby bottle / **toy**

Figure 2.2: A Few Examples from Winogrande Dataset Sakaguchi *et al.* (2020) Showing Selected Biased Samples Using Aflite

see an analysis of AFlite on two relative baselines.

- Random data reduction
- PMI based filtering

Figure 2.3 shows the plots of RoBERTa based pre-computed embeddings visualized in 2 Dimensions and 1 Dimension with the help of PCA dimension reduction. The figure suggests that the distinction between the two labels is quite prominent for other datasets except for the debiased one. For the debiased Winogrande, the distinction between the labels is not very distinct. This indicates the reduction of spurious correlation in the dataset. Another interesting finding here is that even though there is a high separation between the two labels after principal component analysis on PMI-filtered subsets there is a minimal reduction in the KL divergence.

Algorithm 2: AFLite Sakaguchi *et al.* (2020)

```
Input : Dataset $D = (X,Y)$,
Hyper-Parameters: Model Family $M$, Number of random partitions $m$,target
  dataset size $n$, training set size $t < n$,slice size $k < n$ and early stopping
  thresh-hold $tau$
Output : Pruned dataset $S$
$S = D$
while $|S| > n$ do
    // Filtering Phase
    forall $i \in S$ do
        Initialize multiset of out-of-sample predictions E(i) = $\emptyset$
    end
    forall $j \in m$ do
        Randomly partition $S$ into $(T_j, S \backslash T_j)$ s.t. $|S \backslash T_j| = t$
        Train a classifier $\mathcal{L} \in \mathcal{M}$ on $\{(\Phi(x), y) \mid (x, y) \in S \backslash\ T_j\}$ ($\mathcal{L}$ is typically a linear classifier)
        forall  $i = (x, y) \in T_j$ do
            Add the prediction $\mathcal{L}\ (\Phi(x))$ to $E(i)$
        end
    end
    forall $i = (x, y) \in S$ do
        Compute the predictability score $\tilde{p}(i) =| \{\hat{y} \in E(i) \text{ s.t. } \hat{y} = y\}|/|E(i)$
    end
    Select up to $k$ instances $S'$ in $S$ with the highest predictability scores subject to $\tilde{p}(i) \geq \tau$
    $S = S \backslash S'$
    if $|S'| < k$ then
        break
    end
    return $S$
end
```

What are the issues with Adversarial filtering techniques?

Even though Adversarial filtering seems to be an ideal approach for pruning, there are certain drawbacks to the existing approach:

- In algorithm 2, there is no restriction on the number of samples belonging to a

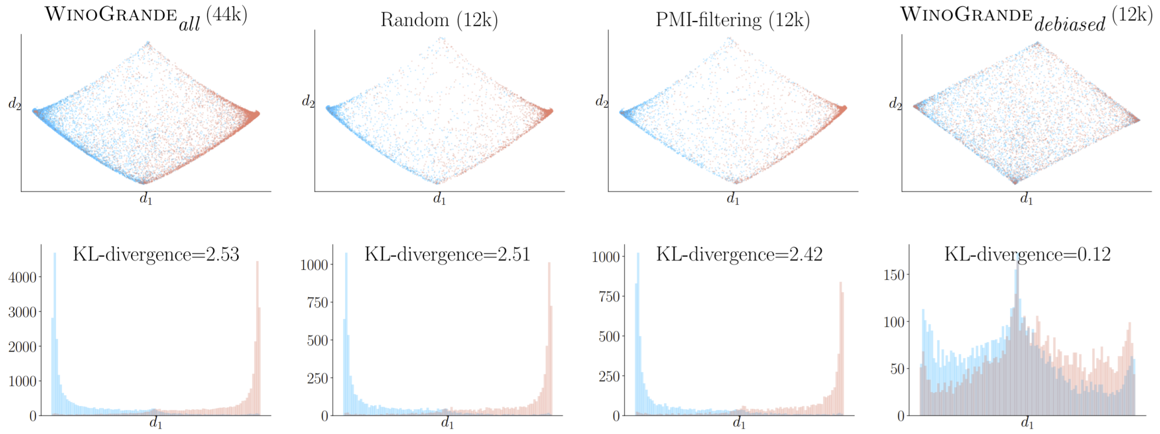


Figure 2.3: Roberta's Pre-computed Embedding Visualized in 2d and 1d

specific class; therefore, entire classes can also vanish.

- Due to the lack of similar restriction samples, though easy, might be outliers and be representative of a context $\mathcal{N}$.

- The whole algorithm depends heavily on the parameter m, the number of iteration is needed to reach a conclusive debiased dataset, causes computational issues. For example, to give bird eye perspective to this problem for a dataset like SNLI (Bowman *et al.*, 2015a) having a data size of 550k for every 10k iterations if linear or a non-linear model has to run 64 times, there are 3520 models trained. Even if it takes 5 minutes to run a normal SVM for such big data, it will take 12 days to complete the filtering process.

Apart from these technical fallacies, one major issue with the pruning methodologies, in general, is the wastage of resources in creating the data in the first place. In a recent field survey Alegion (2019) it was noted that it needs around 1,00,000 data samples to perform well. Using a service like Amazon's Mechanical Turk software to crowdsource the data creation of 100000 samples takes around $70000 on average. Further to annotate the same data based on its complexity costs anywhere from

$8000 to $80000. If, at the end of the day, after pruning, one is throwing 80 percent of the data created, it is a waste of 80 percent of the money spent on the creation.The estimates for the cost per sample is shown in the Figure 2.4. Resolution for this flaw is the primary motivation behind our overall research.

Another major issue with the pruning methodologies is lack of a feedback mechanism. There is no feedback provided to a dataset creator therefore no further restriction on creating these biased samples again in future Arunkumar *et al.* (2020).

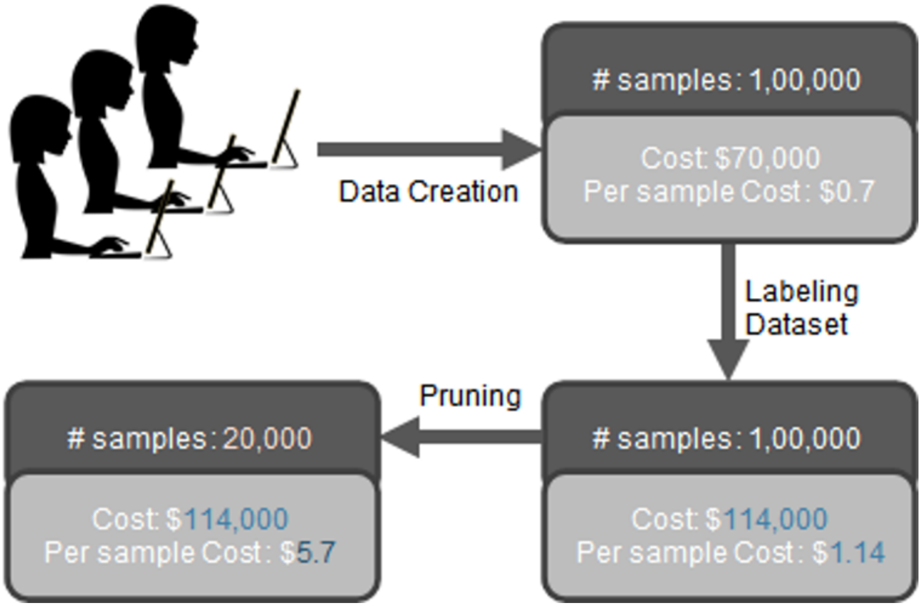


Figure 2.4: Wastage of resources with pruning

2.4 Robustness and Generalization Evaluation

Accuracy has been shown to be not a reliable indicator of model capabilities Ribeiro *et al.* (2020a) and various alternate metrics have been proposed. Robustness Hendrycks *et al.* (2020b,a); Mishra and Arunkumar (2021) is an important aspect of model evaluation beyond accuracy. Various kinds of robustness metrics have been proposed in literature that measure model performance under perturbed instances Jia *et al.* (2019); Jones *et al.* (2020). We incorporate several types of such perturbation based robustness evaluation metrics in our work. However, perturbation accuracy based evaluation does not incorporate ordering of samples, unlike in a typical interview setup where the interviewer gauge a candidate's potential by asking followup question to a given question in a particular order ie. easy to hard. We bring in order of samples and propose new evaluation metrics to study robustness. In this process, we expand on a recent work Jin *et al.* (2019a) which has used 'query number' as the average number of queries to fool a model.

Generalization on the other hand, refers to how model's learning transfers from the training to the test split. Recently, with the IID generalization achieving super-human performance across various benchmarks, focus has been shifted towards evaluating Out of Distribution (OOD) generalization Hendrycks *et al.* (2020b); Talmor and Berant (2019); Bras *et al.* (2020a); Mishra *et al.* (2020b); Mishra and Sachdeva (2020). Often these two terms "robustness" and "generalization" are used interchangeably in literature, we study the interplay between robustness and generalization to see if they are positively correlated or not.

2.4.1 Generalization

What is supervised learning? Many of the success in Deep learning can be attributed to the supervised learning framework that leverages large scale datasets such as Imagenet Deng *et al.* (2009) in vision and SQuAD Rajpurkar *et al.* (2016) and SNLIBowman *et al.* (2015a) in NLP. Large scale neural models such as Efficient Net Tan and Le (2019) in Vision and BERT Devlin *et al.* (2019), RoBERTA Liu *et al.* (2019) in NLP have achieved super-human performance leveraging the supervised learning setup. Supervised learning typically involves fine-tuning of models on task-specific data. Note that, these models are already pre-trained on large training corpus in a self-supervised manner such as mask language modelling and next sentence prediction.

Even though supervised learning is very successful, there are several issues attached with it. First, it requires a large labelled dataset of the downstream task. Second, it requires some time and computation for the model to get trained on. Each of these is associated with various disadvantages. For example, getting a large labelled dataset is an expensive process and various data creation approaches like crowdsourcing contain the risk of creating biased data. Requirement of computational time for training hinders application of supervised systems in real world applications e.g. in conversational agents the inference time has to be real time and users can not wait for a system to get trained on before answering a question.

Generalization is one of the critical problems in the field of supervised learning. The supervised learning task is to learn a mapping function given training data of input-output pairs. With training data, the outputs are already known. The initial success of the model is measured in terms of how many of the inputs successfully map to the correct outputs in a held-out part of the original dataset. This measure is known as accuracy. Generalization refers to the success of the model on new data. After training on the train set, how much the model can digest the new data for the most accurate predictions.

In the generalization literature, apart from data augmentation, many other strategies are used to avoid overfitting the Deep Learning models. One central focus idea is to improve the model architecture itself, which led to the development of more complex architectures such as RNN, LSTM, Word2Vec, Glove and Transformers(Wolf *et al.*, 2019).

To avoid overfitting and achieve more generalization, there have been more functional solutions as well, such as:

- **Dropout**: Dropout is used in Deep Learning to reduce the overfitting in large neural networks. Previous regularization methodologies such as L1 and L2 weight penalties could not wholly solve the overfitting issue due to Co-adaptation. Therefore, it became difficult to expand a neural network's size, and consequently, its accuracy was limited. Dropout is a regularization approach that solved the issue of co-adaptation. It is a technique in which neurons are randomly ignored during the training process. Weights were now learned infractions instead of learning all the wights in each iteration. This results in learning more critical features that are useful with different random subsets of other neurons.

- **Batch Normalization**: Dealing with Deep Neural Networks can be a challenging task, especially when it comes to handling data with the addition of extra layers. Training the deeper layers can be tricky as they can be sensitive to initial random weights and the configuration used in the learning algorithm. The distribution of data to the inner layers might be affected by a small delta change in the weights introduced by the mini-batch as they pass deeper into the network. So, one of the best solutions to handle such a distortion in the data is by using Batch Normalization. Normalization is the process of data pre-processing that is used to bring the data to a common ground without distorting the shape of the data. It's a method used by the model to generalize the data when we input data to the machine learning or deep learning model. Similarly, Batch Normalization is the process to enhance of the performance of the Deep Neural Networks by the inclusion of extra layers. The newly added layers are expected to normalize the data coming from the previous layers and hence help the model in generalizing the data. Since this normalization happens over a set of data provided to the model, hence the name Batch Normalization. This process is proved to increase the speed of the training process of the model, handles the internal covariate shift, and smoothens the loss function

- **Transfer Learning**:Transfer Learning is a process where we leverage the labeled data from a pre-existing model trained on more generic tasks and have a sizeable dataset. Apply this knowledge to a different model designed for a similar/related task with a more specific domain. For example, To detect pedestrians on night-time images, we can pre-trained model designed for a similar domain, i.e., daytime images.

- **Pre-training**: Pre-training is a process in which we use weights from a previously trained model as initial weights to perform a task with different but related testing datasets. As the previous model has already optimized it on the training dataset, the weights are more inclined than the random weights, which have nothing in common with the dataset. For example, training on a dataset of images will require a large number of resources and time for a car recognition model. So, we can use a pre-trained model in this case to transfer the optimized weights.

- **Zero-Shot evaluation** Owing to the disadvantages of the supervised learning setup that limits adoption of Machine Learning sytems in real-world application, Zero-shot evaluation Radford *et al.* (2019) has become popular, where model is not trained on downstream task and instead, is directly evaluated. Since zero data is used for training, its called zero shot. There are several variants of zeroshot learning such as one-shot and the recently popular few-shot learning Brown *et al.* (2020) and Mishra *et al.* (2021). We also adopt zero shot evaluation in our work, similar to the zeroshot learning paradigm set up by prior works

Bras *et al.* (2020a) and Hendrycks *et al.* (2020b).

2.4.2 Independently and Identically Distributed Dataset

In mathematical statistics, an **independent** event is one that is not influenced by the chance of any other event happening or not happening. The probability of occurrence of that event is not dependent on any other event. There is no connection between various observations. An excellent example of independent events is flipping a coin. As one flips the coin, the result is not dependant on the flip of another coin or subsequent coin flips.

Identically Distributed relates to the probability distribution that describes the characteristic you are measuring. Specifically, one probability distribution should adequately model all values you observe in a sample. Consequently, a dataset should not contain trends because they indicate that one probability distribution does not describe all the data. Developing further on the coin toss example, if one tossed the coin 100 times and got 80 times head and 20 times tail. On the 101st toss, will the outcome be dependant on the previous tosses? The probability of getting a head or a tail remains the same, i.e., 0.5. Therefore the outcomes we get from flipping a coin are independent and identical. It is independent because one outcome does not depend on the other outcome. It is identical because every sample comes from the same distribution.

why is IID necessary? The independent and identical distribution of data is essential for the stability of the results. The identical data is created by random sampling from the whole feature space. For the data to be IID, there should not be any trend in the dataset. The In-domain or Independent and identically distributed(IID) performance is evaluated because of the assumption that the data distribution will remain same across training and evaluation spits. This thesis shows an experiment to understand the contribution of biased or bad samples on model performance.

NLI

Natural Language Inference is a subtask of the Natural language processing domain. It involves two given statements, usually referred to as a premise and a hypothesis. The inference task is to tell the hypothesis is going to be true or ”entailed”; not true or ”contradiction”; cannot say or ”neutral” given the premise is true. Large scale NLI datasets such as SNLI Bowman *et al.* (2015b), MNLI Williams *et al.* (2018) have accelerated the development of NLI Research. Variants of these such as Adversarial NLI Nie *et al.* (2019) and Uncertain NLI Chen *et al.* (2020) have been recently proposed to study NLI in various circumstances. In our setup, we use a mix of NLI datasets, some of which we use to train models and some to evaluate models in OOD setup.

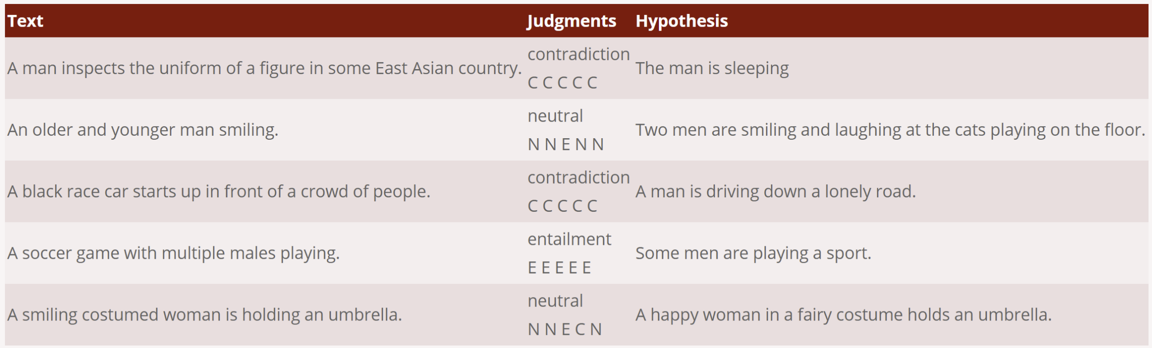
SNLI

SNLI or Stanford Natural Language Inference dataset is a collection of 570k sentence pairs in the NLI setting. All the sentence pairs are labeled as either entailment, contradiction, or neutral. The premises are initially taken from the image captioning dataset Flickr30k; while human annotators created the corresponding hypothesis.

2.4.3 Out of Domain Datasets

Conventionally, Machine Learning operates on Independently and Identical Distribution (IID), where the training and test sets are selected from the same distribution. However, in the real world, the test set distribution may differ from the training set. This difference can be due to various reasons:

- it is hard for a test set to characterize the entire distribution Torralba and Efros



Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Figure 2.5: A Few Examples from the Snli Dataset (Bowman *et al.*, 2015a)

(2011),

- because of various environmental reasons test distribution might change over time Hendrycks *et al.* (2020b). For example; Covid-19 may be an Out of distribution (OOD) sample for a general model trained on various diseases.

One straightforward solution is to create a new training set for everything the test distribution changes. However, it gets exponentially expensive to catch a data distribution that evolves quickly over time. It will almost always be the case that the models will encounter an unexpected situation at test time. Ideally, models should generalize to the OOD samples, but often this is hard. Humans, on the other hand, prefer to abstain from answering when they detect OOD samples and are not confident about answering Kamath *et al.* (2020); Varshney *et al.* (2020). Considering the importance of OODs, we use several OOD datasets in our experimental setup and evaluate the capability of models in distributions that are different from the training set.

Stress NLI dataset:

This work proposes an evaluation methodology consisting of automatically constructed ”stress tests” that allow us to examine whether systems can make accurate inferential decisions.

For the creation of the stress NLI dataset Naik *et al.* (2018) sampled 100 misclassified examples from each category of genre-matched and mismatched sets, analyzed their potential sources of errors, and grouped them into a typology of common reasons for the error. The causes for errors can broadly be divided into categories shown below:

- Word Overlap (29%): Large word-overlap between premise and hypothesis sentences causes wrong entailment prediction, even if they are unrelated. Minimal word overlap causes a prediction of neutral instead of entailment.

- Negation (13%): Strong negation words (”no”, ”not”) cause the model to predict contradiction for neutral or entailed statements.

- Antonymy (5%): Premise-hypothesis pairs containing antonyms (instead of explicit negation) are not detected as a contradiction by the model.

- Numerical Reasoning (4%): For some premise-hypothesis pairs, the model cannot perform reasoning involving numbers or quantifiers for correct relation prediction.

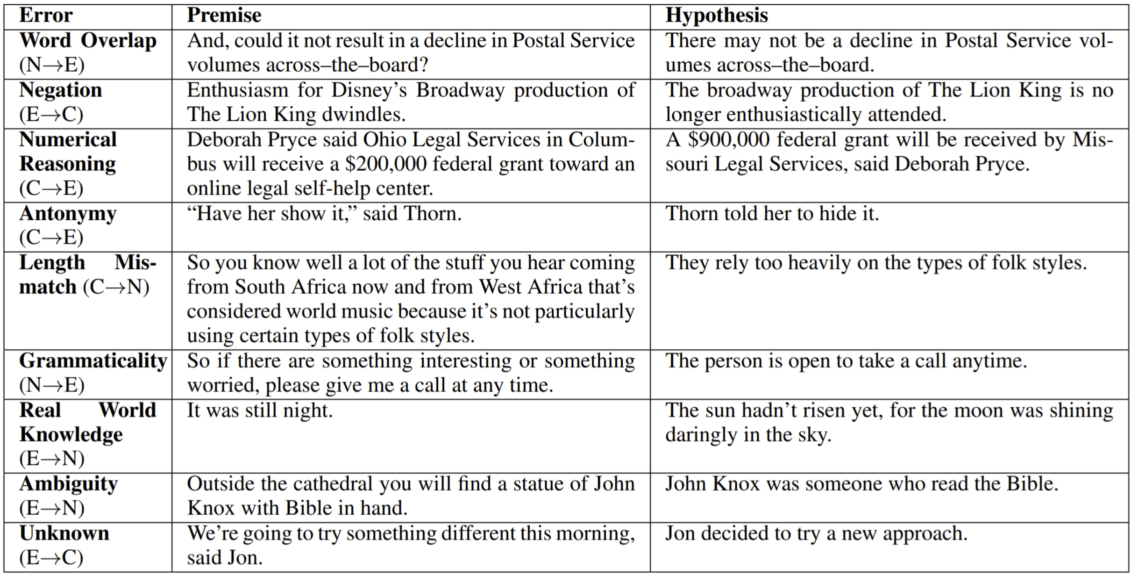
- Length Mismatch (3%): The premise is much longer than the hypothesis, and this extra information could act as a distraction for the model.

- Grammaticality (3%): The premise or the hypothesis is ill-formed because of spelling errors or incorrect subject-verb agreement.

- Real-World Knowledge (12%): These examples are hard to classify without some real-world knowledge.

- Ambiguity (6%): For some instances, the correct answer is unclear to humans. These are the most difficult cases.

- Unknown (26%): No obvious source of error is discernible in these samples



Error	Premise	Hypothesis
Word Overlap (N→E)	And, could it not result in a decline in Postal Service volumes across–the–board?	There may not be a decline in Postal Service volumes across–the–board.
Negation (E→C)	Enthusiasm for Disney's Broadway production of The Lion King dwindles.	The broadway production of The Lion King is no longer enthusiastically attended.
Numerical Reasoning (C→E)	Deborah Pryce said Ohio Legal Services in Columbus will receive a $200,000 federal grant toward an online legal self-help center.	A $900,000 federal grant will be received by Missouri Legal Services, said Deborah Pryce.
Antonymy (C→E)	"Have her show it," said Thorn.	Thorn told her to hide it.
Length Mismatch (C→N)	So you know well a lot of the stuff you hear coming from South Africa now and from West Africa that's considered world music because it's not particularly using certain types of folk styles.	They rely too heavily on the types of folk styles.
Grammaticality (N→E)	So if there are something interesting or something worried, please give me a call at any time.	The person is open to take a call anytime.
Real World Knowledge (E→N)	It was still night.	The sun hadn't risen yet, for the moon was shining daringly in the sky.
Ambiguity (E→N)	Outside the cathedral you will find a statue of John Knox with Bible in hand.	John Knox was someone who read the Bible.
Unknown (E→C)	We're going to try something different this morning, said Jon.	Jon decided to try a new approach.

Figure 2.6: One example for each miss-classified error category.

Adversarial NLI

ANLI Nie *et al.* (2020) uses a human and model in the loop approach to manually generate adversarial examples with the help of a model with an increasing level of complexities. Each subset of the ANLI OOD test is generated by a model trained on the previously generated samples, thus making it more difficult in every iteration. The process of generating the samples consist of four steps.

- Data Collection, the premise is taken from a previously defined corpus such as Wikipedia. A Human writer is asked to write the hypothesis given a target label and premise or context from corpora.

- Get model Feedback, the generated sample consisting or a premise and a hypothesis are given to a model.

- Verify Samples and make splits, the samples are compared with the help of different human if the human verifies the model is correct and the human is also correct the samples contributes to the train set otherwise if the model is wrong and human was correct the sample goes to the validation or the test set.

- Retrain the model, in this step we perform the adversarial training in order to generate even more difficult samples in the future.

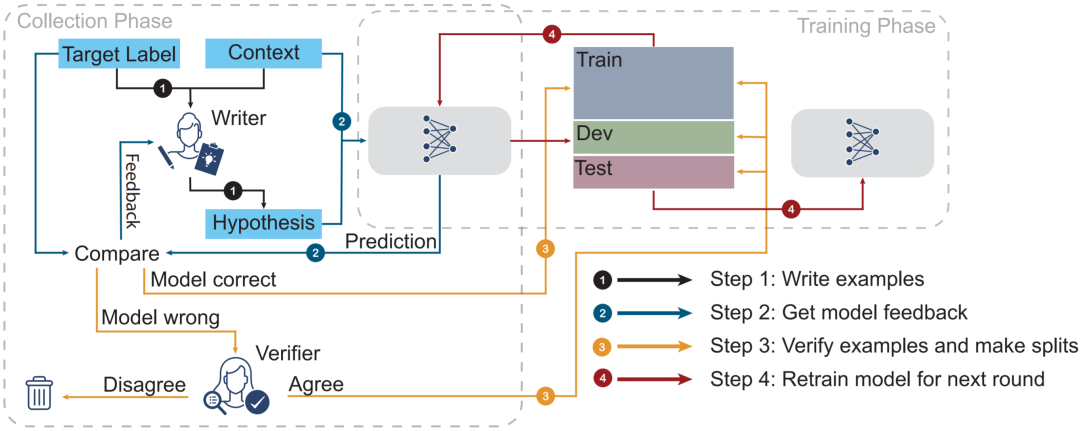


Figure 2.7: The Process of Generating Adversarial Samples Through Human and Model in the Loop Strategy Nie *et al.* (2020)

See figure 2.7 for more detail for the process of adversaries generation using human and model in the loop methodology.

NLI Diagnostics

This dataset is used to analyse the performance of model on the language. It uses Natural Language Inference (NLI) problems to analyse the given sentence. It supports from straightforward evaluation of the sentence to resolving high-level reasoning and syntatic ambiguity.

Lexical Semantics: This phenomena is used to analyse the sentences on the basis of the word meaning.

This includes words which are related to each other. For eg. dog lexically entails animal because any characterics that applies to dog also apply to animal but it contradicts cat because it's impossible to be both at once

Words which are condradicting in meaning but one word is derived from another. For eg. affordable and unaffordable, agree to disagree, ever and never.

Context of the words appearing the sentence and thier position in relation to lexical triggers (verbs and adverbs) Symmetry relations where both words can be treated as one subject and both are connected to each other. Reduncy where words that can be removed from the sentence without changing its meanings.

Predicate-Argument Structure: This entails to how the different parts of the sentence come together to form the whole sentence. These are issues with this phenomena. For eg. Syntatic ambiguity, prepositional phrases, core arguments where verbs used for subject and object etc.

Logic: After getting the structure of the sentence, we draw more conclusions from it using logical operators. These are operators which are commonly used to understand the sentence: Negation, Double Negation, Conjunction, Disjunction, Conditionals. There are other ways to analyse the sentence using natural language analog of universal and existential quantifiers for eg. all, some, many and no.

Knowledge and Common Sense: World knowledge and common sense is required to understand the language. For Instance, In world knowledge, we focus on knowledge that be expressed as facts, as well as common geological, legal and political, technical and cultural knowledge. Common sense is the knowledge that is expected to be possessed by most people independant of their educational and cultural background. This includes understading of basic common social and physical dynamics as well as lexical meaning.

2.4.4 Robustness

Data Augmentation

Deep Neural Networks(DNNs) have shown high performance on various vision and natural language tasks. The success of DNNs has been fueled by advances in model architectures, intense computation capabilities, and access to big data. However, sometimes the required data to learn a task without overfitting is limited. Overfitting refers to the concept of learning a function with high variance. It perfectly fits the training data to the finest detail, which includes the noise as well.

A practical example of limited data is the medical domain, where the amount of crucial and verified data is scarce.

Data augmentation circumscribes a collection of the techniques used to increase the amount of data by slight ”addition to,” ”subtraction from,” or ”modification of” existing data.

Data augmentation for vision vs NLP Data augmentation is an integral part of vision applications. Images are relatively easy to augment by small perturbations such as cropping, rotating, zooming, noise injection, etc., without changing the overall sense of the image.

For natural language processing, augmentation is much more complicated due to the high complexity of the language and the grammatical structure of the text. The augmented dataset is generated with the train set as shown in the figure 2.8

Augmentation Techniques Used

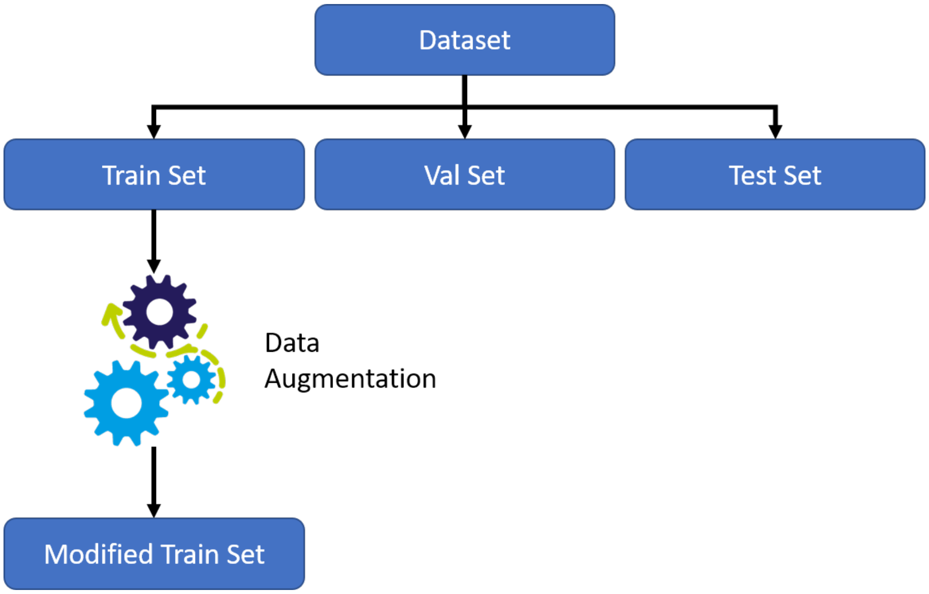


Figure 2.8: Augmentation Is Done on the Train Set.

1. **Contextualized Perturbation for Textual Adversarial Attack**

   CLARE, short for Contextualized Perturbation for Textual Adversarial Attack, was introduced in (Li *et al.*, 2020a). CLARE follows a mask-the-infill procedure. It masks a word in the sentence and then predicts that word using a pre-trained masked language model. In the sentence for every chosen position, either of the three perturbation actions is performed: Merge, Replace, and Insertion. The process is shown in figure 2.9

   - **Replace:** A word is selected in the sentence, and another word replaces that keyword of similar meaning. For example, "The trailer of the movie was scary." can be changed to "The trailer of the movie was horrifying.", changing the word "scary" to "horrifying."

     To "replace," first, the word w_i will be masked, from the candidate list Z a word z will be chosen and will be placed at the position of the mask.

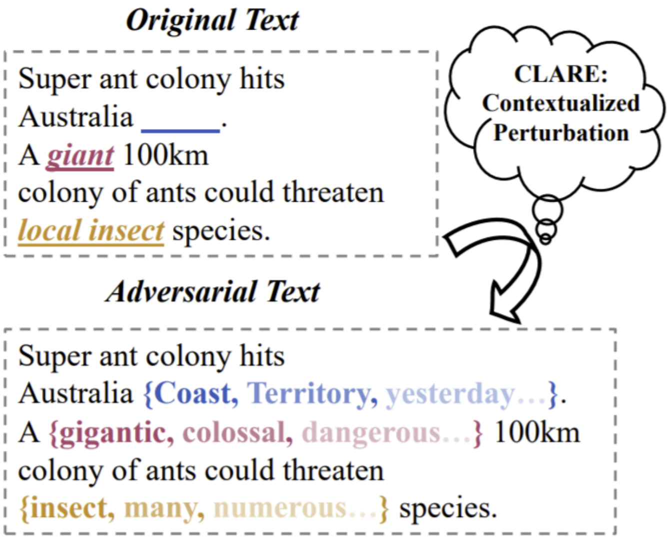

Original Text
Super ant colony hits
Australia ____.
A giant 100km
colony of ants could threaten
local insect species.
CLARE:
Contextualized
Perturbation
Adversarial Text
Super ant colony hits
Australia {Coast, Territory, yesterday...}.
A {gigantic, colossal, dangerous...} 100km
colony of ants could threaten
{insect, many, numerous...} species.

Figure 2.9: Illustration of Clare.

This process is summarized in equation 2.4

$$\begin{aligned} \widetilde{\mathbf{x}} &= x_1 \ldots x_{i-1}[\text{MASK}]x_{i+1} \ldots x_n, \\ \text{replace}\left(\mathbf{x}, i\right) &= x_1 \ldots x_{i-1} z x_{i+1} \ldots x_n. \end{aligned} \tag{2.4}$$

- **Insert:** "Insert" differs from the "Replace" regarding how the mask is placed in the sentence. In replace, we put the mask in place of a word $\mathrm{w_i}$. However, in case of insertion, we put the mask after the word $\mathrm{w_i}$. For example, "I insist that..." can be changed to "I highly insist...", "highly" being the keyword added to the sentence. The equation 2.5 shows how the mask is added after $\mathrm{w_i}$, and a new word is predicted by the masked language model to be added at that location.

$$\begin{aligned} \widetilde{\mathbf{x}} &= x_1 \ldots x_i[\text{MASK}]x_{i+1} \ldots x_n \\ \text{insert}(\mathbf{x}, i) &= x_1 \ldots x_i z x_{i+1} \ldots x_n. \end{aligned} \tag{2.5}$$

- **Merge:** For the merge operation, a bigram is chosen from the sentence

$\mathrm{x_i x_{i+1}}$ and is replaced by a single mask. The masked language model tries to predict a single word in place of two words, thus merging them. This process is further explained in the equation **?**

$$\begin{aligned} \widetilde{\mathbf{x}} &= x_1 \ldots x_{i-1}[\text{MASK}]x_{i+2} \ldots x_n \\ \text{merge}(\mathbf{x}, i) &= x_1 \ldots x_{i-1} z x_{i+2} \ldots x_n \end{aligned} \tag{2.6}$$

2. **CharSwap**

Traditionally words could be represented as vectors using word-level or character-level embedding. For word embeddings, each word is mapped into low dimensional dense vectors directly from a lookup table. Character embeddings are usually obtained by applying neural networks on the character sequence of each word, and the hidden states are used to form the representation.

There are several strategies to integrate word-level and character-level embedding for a fine-grained word representation. The following techniques can be used to augment data at the character level.

- OCR Augmenter: Substitute character by pre-defined OCR error.
- Keyboard Augmenter: Substitute character by keyboard distance.
- Random Augmenter: Insert character randomly.
- Substitute character randomly from a position in the word another character.
- Swap character randomly from a given position in the word with another position.
- Delete character randomly from the given word.

3. **CheckList**

A CheckList is a task agnostic methodology for testing NLP models. A CheckList proposes a general framework for writing behavioral tests for any NLP model and task.

The idea behind CheckList is a conceptual matrix that is composed of linguistic capabilities like Named Entity Switching, Negations, Robustness to Typos as rows and test types like Minimum Functionality Test (MFT), Directional Expectation (DIR), and Invariance Test (INV) as columns (Ribeiro *et al.*, 2020b). It facilitates extensive test ideation to generate a large, diverse number of test cases easily. The components behind the conceptual matrix are described below.

(a) Test Types: These are the columns in the CheckList matrix. There are three test types proposed by the CheckList framework.

   i. Minimum Functionality Testing: A collection of (text, expected label) pairs is built from scratch and the model on this collection. The goal of this test is to make sure the model is not taking any shortcuts and possesses linguistic capabilities.

   ii. Invariance Test: In this test, we perturb our existing training examples in a way that the label should not change. Then, the model is tested on this perturbed example and the model passes the test only if its prediction remains the same (i.e invariant).

   iii. Directional Expectation Test: This test is similar to the invariance test but here we expect the model prediction to change after perturbation.

(b) Linguistic Capabilities: These are the rows in the CheckList matrix. Each row contains a specific linguistic capability that applies to most NLP tasks.

   i. Vocabulary and POS: We want to ensure the model has enough vocabulary knowledge and can differentiate words with a different part of speech and how it impacts the task at hand.

ii. Named Entity Recognition: It tests the capability of the model to understand named entities and whether it is important for the current task or not.

iii. Temporal: Here we want to test if the model understands the order of events in the text.

iv. Negation: This ensures the model understands negation and its impact on the output.

v. Semantic Role Labeling: This ensures the model understands the agent and the object in the text.

4. **Easy Data Augmentation**

EDA, short for Easy Data Augmentation (Wei and Zou, 2019), is a combination of simple yet powerful operations inspired by computer vision-based augmentation techniques, namely:

- Synonym Replacement (SR): In synonym replacement, the idea is to choose 'n' words from all the words in the sentence except the stop words and replace them in place with their synonyms.
- Random Insertion (RI): In random insertion, the idea is to choose a word from the sentence except for the stop words. Pick a random synonym of the chosen word and insert it randomly somewhere in the sentence. This process has to be repeated n times.
- Random Swap (RS): In the random swap, the idea is to choose two words n times and swap them with each other.

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
SR	A lamentable, superior human comedy played out on the back roads of life.
RI	A sad, superior human comedy played out on funniness the back roads of life.
RS	A sad, superior human comedy played out on roads back the of life.
RD	A sad, superior human out on the back roads of life.

Table 2.1: Sentences Generated Using Eda. Sr: Synonym Replacement. Ri: Ran-dom Insertion. Rs: Random Swap. Rd: Random Deletion (Wei and Zou, 2019)

- Random Deletion (RD): Randomly remove each word in the sentence with probability p.

5. **Embedding based augmentation**

   Augments text by transforming words with their embeddings.

6. **WordNet** In WordNet augmentation, the text is augmented by replacing it with synonyms from the WordNet thesaurus. WordNet-based augmentation methodology randomly selects n words from the sentence based on the parts-of-speech tag. A Geometric Distribution calculates the probability of choosing a word in the sentence. This distribution, in turn, is also dependent on the probability of success. Even for the synonym selection, a similar geometric distribution-based approach is followedMarivate and Sefara (2020). The augmentations are performed on the nouns or verbs or event their combinations by replacing them with their synonyms.

Adversarial Examples

Given the recent progress in machine learning and natural language processing methods, there has been an increasing need to generate models that can generalize better on various data and are robust. These requirements came because the data used in developing such models are filled with various biases. For example, natural language processing models are prone to be influenced by various societal biases like gender bias while dealing with a corpus.

There have been several techniques proposed to deal with this situation. One of those techniques is to use adversarial examples to train the model. An adversarial example generally refers to a set of inputs specifically designed to fool a machine learning model. The modulation in input is an adversarial perturbation when the minimal change in the input causes the output to change completely.

There exist two types of adversarial examples - Black-Box examples and White-box examples. Black-box examples are those where information about the model, like gradients or parameters, is unknown while creating the adversarial examples. In contrast, for the White-box examples, we have access to the model and its parameters. When it comes to algorithms that generate adversarial examples for NLP, most of them deal with character/word/sentence level perturbations. For example, (Jia and Liang, 2017b) added extra sentences to fool comprehension models without altering the answer of the question, whereas (Jin *et al.*, 2019b) identified important words and replaced them with their synonyms while maintaining the true meaning and essence of the sentence. Adversarial perturbations are plugged into the model in two ways. First, re-training the original model using some adversarial examples that have successfully fooled the model. The second is to incorporate perturbations in the model training process. Even though adversarial training improves the robustness of models and makes them less vulnerable to adversarial attacks, there exists a trade-off between the generalization of a model (i.e., the standard test accuracy) and its robustness (accuracy on an adversarial dataset). Models trained with an adversarial objective often show an increase in the robustness but a decrease in the standard accuracy. (Min *et al.*, 2020) showed that the trade-off between generalization and robustness exists even in infinite data limit.

Adversarial attacks

Adversarial attacks, on the other hand, refer to the process of generating adversarial perturbations. The perturbations to the original input are minimalistic because if we consider too large perturbations, the output would be different for the two completely different inputs. A major reason that limits the application of Machine Learning systems in various applications is the security aspect. For example, a patient can die in a healthcare application if the system is not reliable. Similarly, an economy can go down if the security system gets broken in a Machine Learning application. Adversarial attacks where models are probed against various perturbations have been the most prevalent approach to evaluating systems' security aspects. Robustness to Adversarial attacks is well-studied literature in Vision Kurakin *et al.* (2016); Zhao *et al.* (2018)– is still under-explored in NLP. The discrete space in NLP makes it hard to maintain semantic coherence and language fluency while generating adversarial data. In our study, we adopt recent adversarial attack approaches Garg and Ramakrishnan (2020a), Li *et al.* (2020b), Jin *et al.* (2019a) which have reduced these issues to a large extent.

TextAttack

Textattack is a framework built using python language for streamlining the recent development in robustness and generalization literature of natural language processing. It provides us with the most recent implementation for adversarial attacks, adversarial training, and data augmentation in NLP. To make the process more streamlined, the TextAttack library provides essential components of NLP such as sentence encoding, language model training on standard datasets, grammar checking, and word replacement strategies. In the following subsections, we will be explaining the components of the TextAttacj used in this research. **Importance of words for attacking** Given a sentence of n words $X = \{w_1, w_2, \ldots, w_n\}$ only some of the words are influential signals for the model $\mathcal{F}$ to make a prediction for a task. Niven and Kao (2019) reached a similar conclusion that BERT attends to statistical cues of some words. Therefore we need some kind of selection mechanism to identify these specific words. The idea of selecting important words is only applicable in black box scenario, as the words can be identified by the gradients of model $\mathcal{F}$. In most of the recent black box adversarial attack techniques the importance score I_{w_i} of a word w_i towards a classification result $F(X) = Y$ is used to identify which words are to be replaced to possibly create adversarial examples. This was introduced by Jin *et al.* (2019a). The I_{w_i} score is defined as the reduction in the confidence of a result after deleting the concerned word. Once we have the importance score as the difference in the predictability score with and without that word we sort them based on importance and filter out the stop words.

$$I_{w_i} = \begin{cases} F_Y(X) - F_Y\left(X_{\backslash w_i}\right), \quad \text{if } F(X) = F\left(X_{\backslash w_i}\right) = Y \\ \left(F_Y(X) - F_Y\left(X_{\backslash w_i}\right)\right) + \left(F_{\bar{Y}}\left(X_{\backslash w_i}\right) - F_{\bar{Y}}(X)\right) \\ \text{if } F(X) = Y, F\left(X_{\backslash w_i}\right) = \bar{Y}, \text{ and } Y \neq \bar{Y} \end{cases} \tag{2.7}$$

BAE ”Bert-based Adversarial Examples for Text Classification” (Garg and Ramakrishnan, 2020b) is one of the state-of-the-art methods that help generate adversarial examples for the downstream task of text classification. The central idea of BAE is based on a black-box attack that uses contextual perturbations obtained from a BERT masked language model. It tries to replace tokens in the original text by masking a portion of it and leveraging BERT-MLM to generate alternatives for the masked portion. It first calculates each token's importance by selectively removing each token and noting the decrease in prediction probability. Then in the decreasing order of the token importance, it masks them and predicts top k tokens for the masked portion. After obtaining the top k tokens, the model uses a sentence similarity scorer to ensure the generated sentence is semantically identical to the original sentence. Suppose multiple sentences can successfully flip the label. In that case, the model chooses those sentences that are most similar to the input sentence. Suppose none of the generated sentences can change the prediction. In that case, the model selects the sentence that decreases the probability of prediction the most. The paper proposes four forms of attack:

- BAE-R: Replace tokens to generate perturbations,
- BAE-I: Insert tokens around the important tokens to generate perturbations,
- BAE-R/I: Either replace or insert a token
- BAE-R+I: First, replace a token and then insert a token to the left or right of the original token.

They evaluate their proposed method on benchmark datasets such as IMDB for sentiment classification, TREC for question type classification, and MPQA datasets for online polarity detection.

Original samples	**Adversarial Examples**
P: A young boy with a blue **fin** with red stripes sitting on a concrete ledge. **H:** The boy is swimming.	**P:** A young boy with a blue **cap** with red stripes sitting on a concrete ledge. **H:** The boy is swimming.
P: Two **men**, both wearing bright yellow vests and jeans, are working on a roof. **H:** Two men are working outside.	**P:** Two **boys**, both wearing bright yellow vests and jeans, are working on a roof. **H:** Two men are working outside.

Table 2.2: Adversarial Examples from B.A.E.

2.5 Neural Language Models

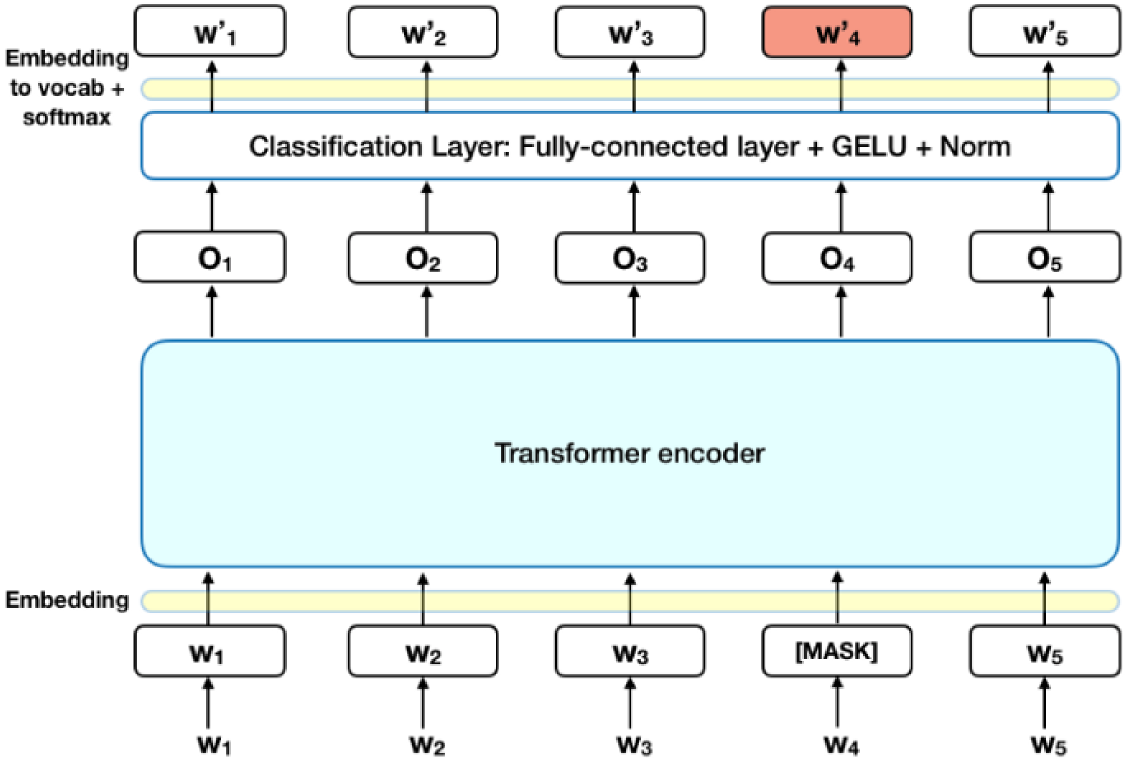
In NLP, a language model is a probability distribution over sequences on an alphabet of tokens. A central problem in language modeling is to learn a language model from examples, such as a model of English sentences from a training set of sentences.

2.5.1 BERT

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary.

Unlike directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it is non-directional. This characteristic allows the model to learn the context of a word based on its surroundings (left and right of the word).

The chart below is a high-level description of the Transformer encoder. The input is a sequence of tokens, which are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors of size H, in which each vector corresponds to an input token with the same index.



W'1
W'2
W'3
W'4
W'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
W1
W2
W3
[MASK]
W5
W1
W2
W3
W4
W5

Figure 2.10: High Level Description of Transformer Encoder.

2.6 Dataset Cartography

Large datasets are a significant prerequisite for the latest neural models. As the size of the data increases, it becomes harder to evaluate these datasets. Swayamdipta *et al.* (2020) provides a way of automatically characterizing data instances concerning their role in achieving good performance on IID as well as OOD. In this study, training dynamics, i.e., the model's behavior as the training progresses, is used to contextualize examples in a dataset creating data maps. The training dynamics include confidence and variability, i.e., the mean and standard deviation of the gold label probabilities for individual samples.

Triaining Dynamics Let us assume we have a train set of size $N, \mathcal{D}$ defined as

$$\mathcal{D} = \{(\boldsymbol{x}, y^*)_i\}_{i=1}^{N} \tag{2.8}$$

Here, the ith instance consists of the observation, $\boldsymbol{x}_i$ and its true label under the task, y_i^*. The training is performed for $\mathcal{E}$ epochs. To calculate the confidence the mean probability of the true label (y_i^*) across $\mathcal{E}$ epochs is defined as:

$$\hat{\mu}_i = \frac{1}{E} \sum_{e=1}^{E} p_{\boldsymbol{\theta}^{(e)}} \left(y_i^* \mid \boldsymbol{x}_i\right) \tag{2.9}$$

In the equation 2.9, $p_{\boldsymbol{\theta}^{(e)}}$ is probability given by the model $\mathcal{F}$ with parameters $\boldsymbol{\theta}^{(e)}$ at the end of the e^{th} epoch. The number of times the models predicts the correct label is defined as the correctness of the model at instance i.

The variability is defined as the spread of $p_{\boldsymbol{\theta}^{(e)}} \left(y_i^* \mid \boldsymbol{x}_i\right)$ across $\mathcal{E}$ epochs, using the standard deviation.

$$\hat{\sigma}_i = \sqrt{\frac{\sum_{e=1}^{E} \left(p_{\boldsymbol{\theta}^{(e)}\left(y_i^* | \boldsymbol{x}_i\right) - \hat{\mu}_i)^2} \right.}{E}} \tag{2.10}$$

Based on these two metrics, the model identifies various regions of the dataset. The samples are ”easy to learn” if the model consistently predicts such instances correctly with high confidence. The samples are ”hard to learn” if they have low variability and low confidence. The third notable group contains ”ambiguous” examples or those with very high variability. The model tends to be indecisive about these instances, as they do not show high confidence or correctness.

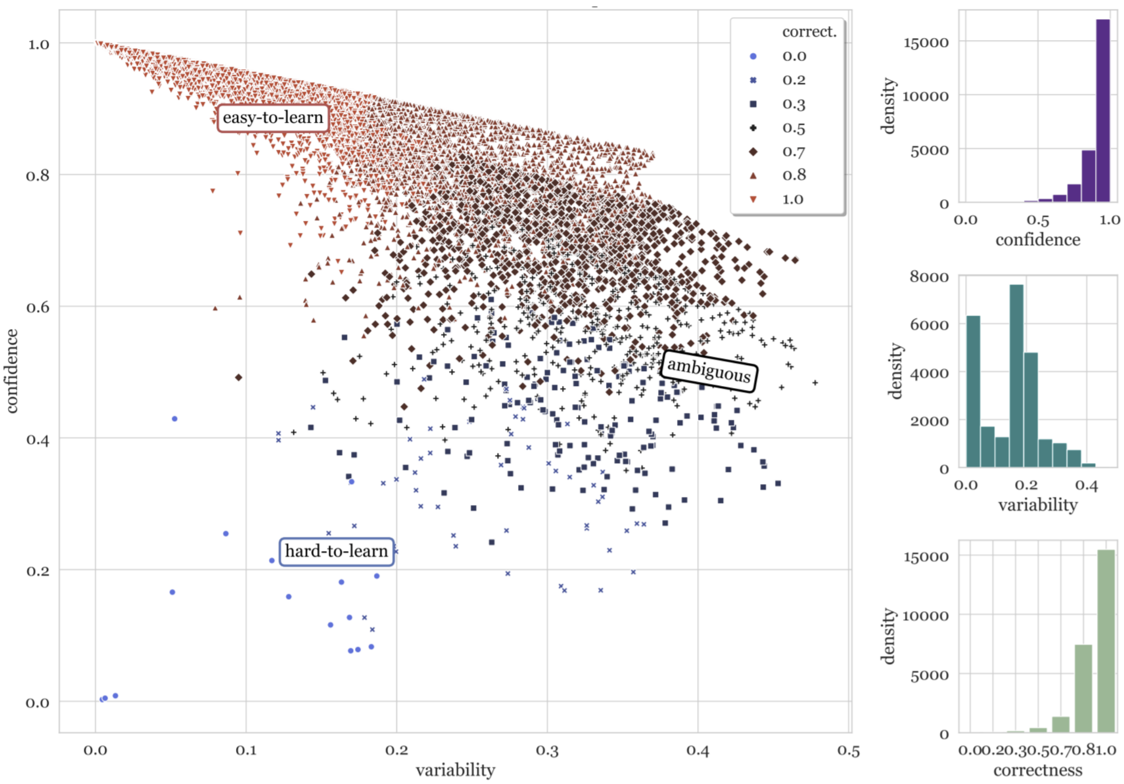


Figure 2.11: Data Map Showing Different Types of Samples Present in Winogrande Dataset.

Chapter 3

EXPERIMENTAL SETUP

3.1 Identifying biased Data Samples

To analyse relationship between the biased and non biased part of the dataset, first we have to identify them. To do so we leverage AFLite Bras *et al.* (2020a); Sakaguchi *et al.* (2019), a recent and successful approach for adversarial filtering of dataset. The following nomenclature is used in the literature and for rest of the discussion.

- biased data samples (which are deleted by AFLite) i.e. 'bad data' and
- non biased data samples (which are retained by AFLite) i.e. 'good data'.

3.2 Evaluation Metrics

For evaluating the **in domain and out of domain performance** we count **the number of correct predictions divided by the total number of predictions** which is defined in the equation 3.1. In the given equation, 'TP' stands for True Positive, 'TN' strands for True negative, 'FP' stands for False positive and 'FN' stands for False negative.

$$Accuracy = \frac{\text{TP} \ + \ \text{TN}}{\text{TP} \ + \text{TN} + \ \text{FP} \ + \text{FN}} \tag{3.1}$$

With the advent of various adversarial attacks, the notion of a metric to quantify the robustness of a model also came into existence. The initial work focused on generating adversarial examples and model accuracy on these examples. The issue with that was that it provides little to no information about the level of perturbations required to attack the model. Goodfellow *et al.* (2015) proposed to use adversarial examples to retrain a model, and the difference in the number of adversarial examples generated is regarded as the level of robustness of the model. However, adversarial training, in general, increases the tolerance towards adversarial examples. Measuring robustness using this methodology would be unproductive. Additionally, suppose there is a high number of adversarial examples. In that case, the model runs a risk of overfitting, which will skew the metrics mentioned above.

In natural language literature, a recent work Jin *et al.* (2019a) used 'query number' as the average number of queries to fool a model, more formally defined as in equation 3.2. This measure gives us an upper bound to the number of attacks, which is useful in many situations but not always ideal.

$$\text{Query Number} = \frac{\text{successful attacks}}{\text{successful attack } + \text{number of unsuccessful attacks}} \tag{3.2}$$

Based on the recent development of various attack methodologies in the natural language domain. **We propose the Robustness Score as the average number of queries before the first successful attack or, more simply put, the minimum number of perturbations required for a successful attack as shown in equation 3.3.** Even though the equation 3.2 is a good measure of the number of perturbations, however, it is not practical because, in an ideal scenario, we are concerned about attacking a sample if it is successful once that is enough. Still, both the formulas have their advantage. For example, consider software to suggest how to improve your credit score by explaining why you were rejected and suggest improvement using adversarial examples. In that case, one might have to see more adversarial generations until a specific adversarial example makes sense. In this case, the equation 3.2 might be a better metric, but for training a model to be more robust, we do not create specific examples.

$$\text{Robustness Score} = \text{Count the number of queries before successful attack} \tag{3.3}$$

The idea behind defining this new robustness metric

3.3 In domain accuracies

To calculate the in domain accuracies we first segregate the biased samples from the non biased samples using AFlite. Once we have good data and bad data we train BERT models by sampling n bad samples and m good samples. The resultant model is tested on the good part of the evaluation set. The value of n and m vary from 0 with a step size of 18000 till 90000. We choose 90k as an upper bound because the final data size for the SNLI dataset after pruning is 92k.

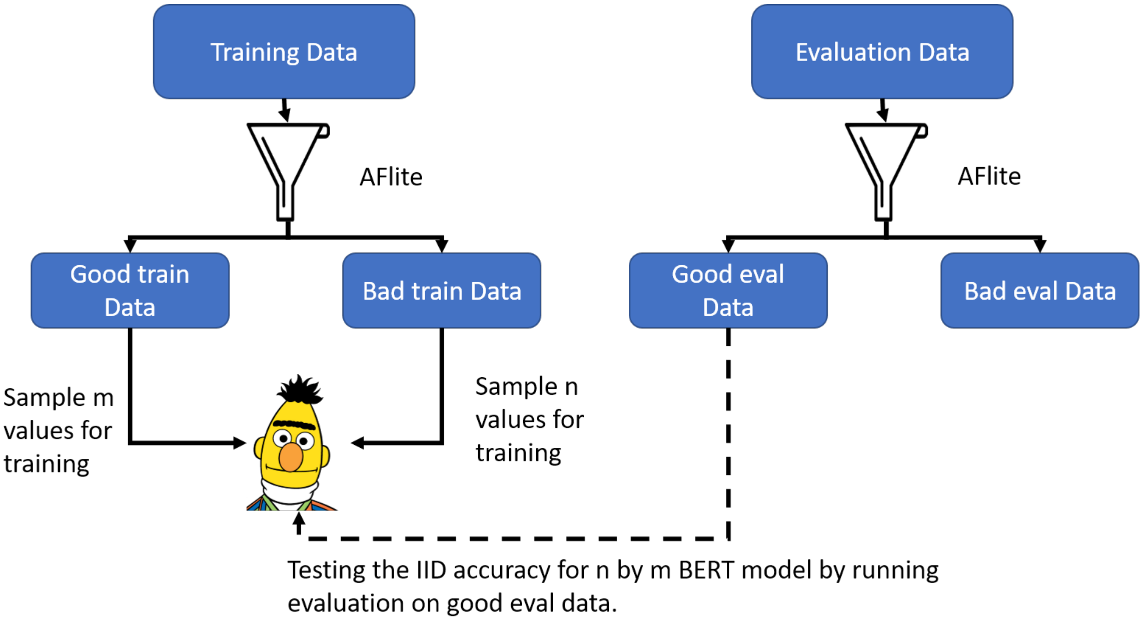

Training Data
Evaluation Data
AFlite
AFlite
Good train Data
Bad train Data
Good eval Data
Bad eval Data
Sample m values for training
Sample n values for training
Testing the IID accuracy for n by m BERT model by running evaluation on good eval data.

Figure 3.1: Procedure for Calculating in Domain Accuracy.

3.4 Robustness

For evaluating effect of biased samples, unbiased samples and repaired samples on robustness we perform two sets of experiments.

3.4.1 Robustness Scores

For comparing the robustness in one scenario we keep the same setting for the models as before, i.e., a BERT model trained on n bad samples and m good samples. With this model we attack the evaluation set using the given attack strategy, in our case as stated before we are using BAE, Bert-based Adversarial Examples for Text Classification. for every sample we calculate the robustness score as the number of queries taken to reach the first successful attack. The total score for a given pair of model and dataset is average of robustness scores of individual samples.

3.4.2 Model free Robustness

The central ideology behind model-free robustness experiments is to corroborate the results of robustness metrics by not using the same model agnostic approach for evaluation. We take BERT models trained on Randomly sampled original dataset, Randomly sampled original dataset and randomly augmented data using easy data augment methodologies Wei and Zou (2019), AFLite identified good dataset, and randomly sampled repaired dataset. All the experiments with randomness were performed 5 times and averaged. All the datasets used with exception of full data baselines are trained of data size of 92k to keep the comparisons fair. Once we have all these model ready we evaluate them on the augmented evaluation set. The augmentations vary from most basic character manipulation with char swap to masked language model based augmentation with CLARE.The higher value of accuracy on

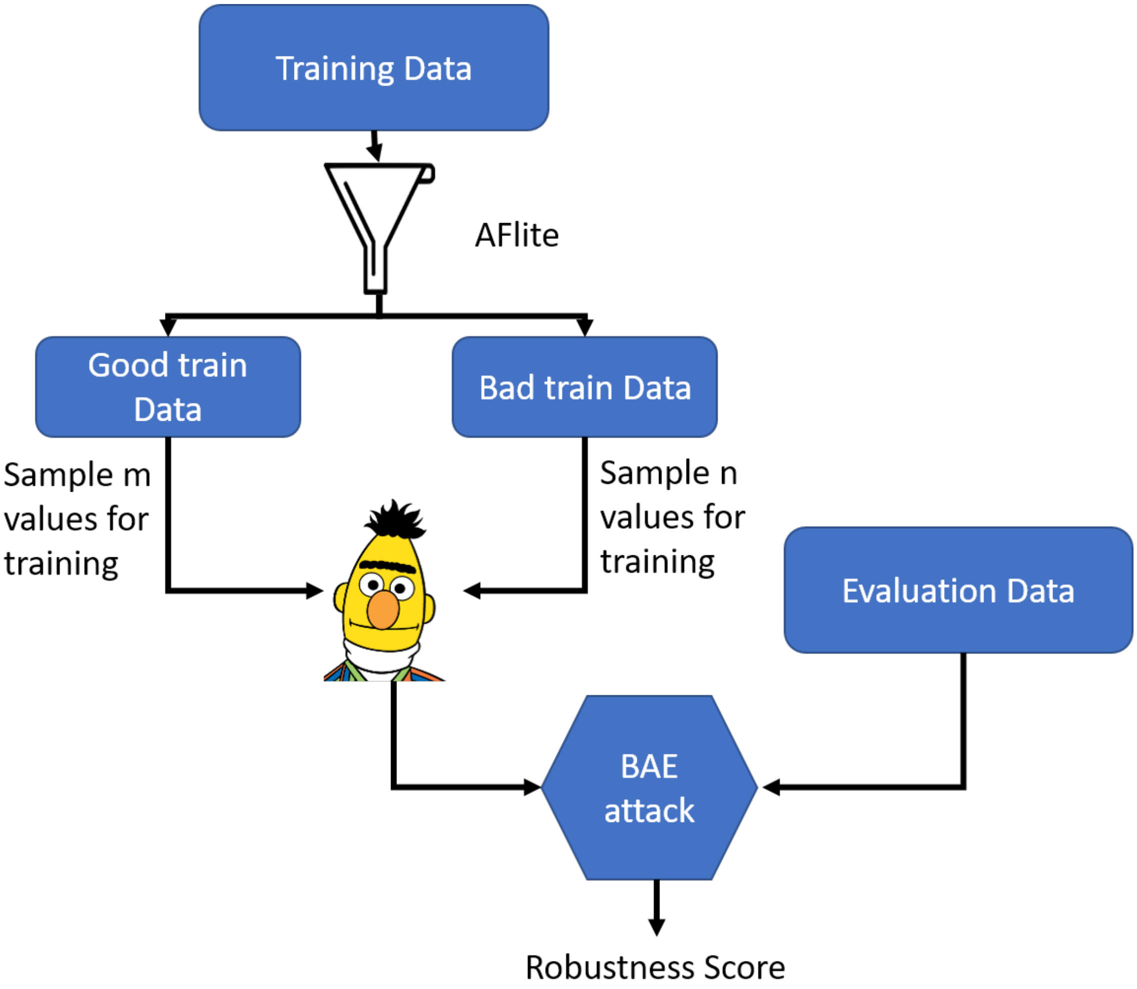


Figure 3.2: Procedure for Calculating Average Robustness.

the augmented evaluation sets indicate more robustness.

3.5 Generalization

For measuring the generalization capabilities, using BERT and $RoBERTa$ model trained on a given size of data, we run evaluations on the Out Of Domain(OOD) datasets defined in section 2.4.3. Both the models are trained on different datasets of size 92k and 182k such as AFLite identified good samples, randomly selected samples, and augmented samples.

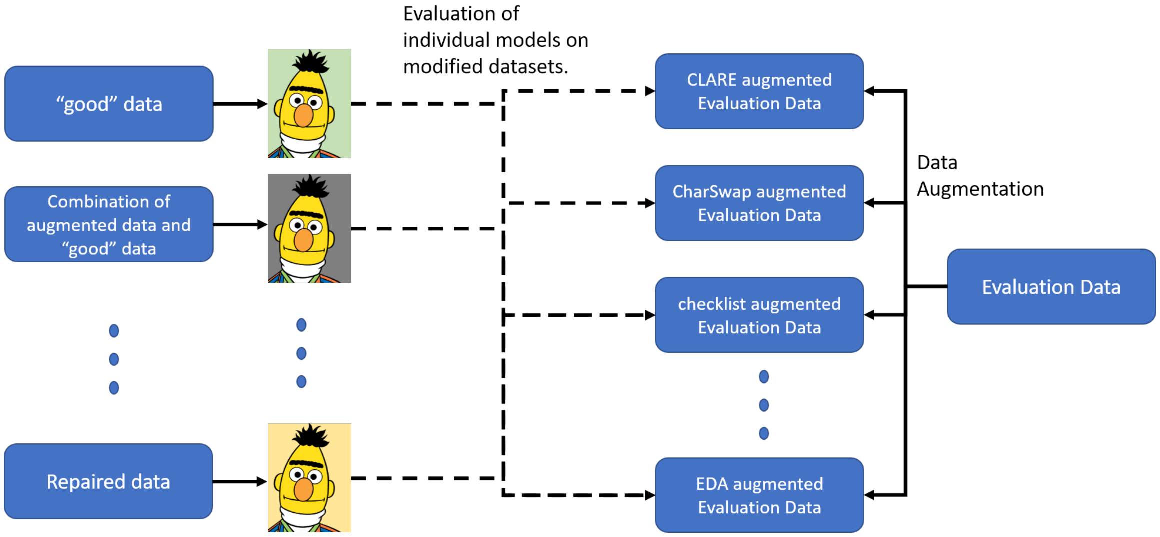

Evaluation of individual models on modified datasets.
"good" data
Combination of augmented data and "good" data
Repaired data
CLARE augmented Evaluation Data
CharSwap augmented Evaluation Data
checklist augmented Evaluation Data
EDA augmented Evaluation Data
Data Augmentation
Evaluation Data

Figure 3.3: Procedure for Model Free Text Attack.

Chapter 4

RESULTS

4.1 In domain accuracies

Unlike conventional software engineering algorithms, machine learning algorithms are not reliable. In machine learning, the results rely on training data and overfit to underlying correlations. However, the model's association to underlying biases should adversely affect the out-of-distribution examples.

As we remove such artifacts, the in-domain accuracy should decrease because the trained model would not have access to spurious correlations. We compare the samples selected with adversarial filtering against randomly chosen samples from the datasets to strengthen our hypothesis. The results are shown in Table 4.1 and the setup is highlighted previously in section 3.3.

The model trained just on the non-biased dataset shows a drop of 18.51% accuracy on the non-biased evaluation set of the SNLI dataset compared to the model trained on only the biased part of the dataset. When the model trained on 90k good samples is compared with a model with 90k good samples and 90k bad samples, we see a drop of 17.78% accuracy. Overall, we see that the significant contribution to accuracy comes from a minimal amount of "biased" or "bad" data. A small set of biased samples are the primary source of accuracy is shown by the first entry in the biased dataset row. **A model trained on only 18k data samples shows an accuracy of 83.81% of the filtered dev set, which is only 4.77 % less than the model's accuracy trained on complete SNLI data.**

Good Data Size -> / Bad Data Size	0	18000	36000	54000	72000	90000
0		66.65	65.85	66.71	67.99	68.11
18000	83.81	83.26	83.57	81.61	81.61	79.05
36000	85.95	84.54	85.16	84.85	82.59	83.57
54000	86.01	85.28	85.95	84.12	85.09	85.64
72000	87.17	87.97	86.81	86.32	85.89	86.19
90000	86.62	86.62	87.48	86.13	86.68	85.89

Table 4.1: Snli Good Data I.I.D. Accuracy with Bert Model

These findings shows biased samples inflate model's performance that leads ti an overestimation of their capabilities. By seeing near-total data accuracy on a small number of samples and as the number of unbiased samples increases, the model's accuracy decreases but not significantly. We can say that spurious correlations are easy to learn but hard to forget.

4.2 Robustness

For an ideal model, we expect it to be both generalizable and robust. For example, consider a hypothetical situation where you need to make payments using face recognition. The model should be generalizable enough to work for different races and communities. It should be robust enough to avoid changing decisions if you wear glasses or a turban.

So far, we have seen neural models are vulnerable to various sample-level attacks. Broadly, it can be model in the loop, i.e., small perturbations to the data sample, and continuously feed to the model to check if the label is flipped. The other way model-free using predefined augmentation techniques, as explained in section 2.4.4.

We hypothesize that when we prune the data, we lose some information in the feature space when we drop samples to avoid biases. This lost information should be reflected in some way in the model.

The expected negative effect is reflected in robustness. We calculate the robustness based on equation 3.3. Table 4.2 shows how the robustness scores vary with different sizes of good(non-biased) and bad(biased) datasets. The results show that we have less robustness in the non-biased or good dataset. **We see a total decrease of 4.57 on 90k samples**. **We also see 2.02 less robustness when comparing the robustness of models trained on 90k good samples and 90k good + 90k bad samples respectively.** Therefore, in the process of pruning the dataset, we are losing robustness.

The model-free experiments further strengthen the notion that non-biased samples show low robustness. **We drop in an average accuracy of 10.97% for the AFLite identified non-biased samples**. The robustness scores reinforce even further the loss of robustness. The results for model free experiment is shown in Table 4.3 and are visualized in Figure 4.1

4.3 Generalization

Inspired by recent identifications of various spurious artifacts in the SNLI dataset, which makes it considerably simpler. This discovery impelled the development of different Out-of-Distribution datasets. Some of these datasets include HANS McCoy *et al.* (2019), NLI Diagnostics Wang *et al.* (2018a), Stress tests Naik *et al.* (2018) and Adversarial NLI Nie *et al.* (2020). The detailed explanation of these datasets are given in section 2.4.3.

In Table 4.4 and Figure 4.2 we see the zero-shot evaluation on three out-of-

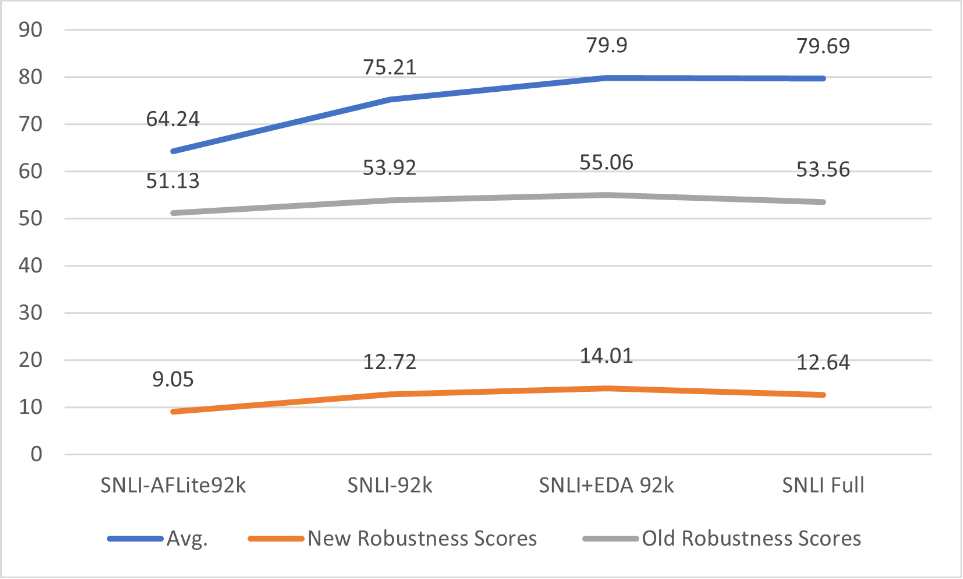


Figure 4.1: Visualization for Model Free Robustness Accuracy and Robustness Scores.

distribution evaluation tasks using *RoBERTa* model trained on original SNLI data, AFLite Filtered data from SNLI and same size Random data. The reported accuracy is averaged across 5 random seeds, and the subscript denotes standard deviation. On the HANS dataset, all models are evaluated on the non-entailment cases of the three syntactic heuristics (Lexical overlap, Subsequence, and Constituent). The NLI-Diagnostics dataset is broken down into the instances requiring world and common-sense knowledge (Knowl.), logical reasoning (Logic), predicate-argument structures (PAS), or lexical semantics (LxS.). Stress tests for NLI are further categorized into Competence, Distraction, and Noise tests. Bras *et al.* (2020a).**We observe that models trained on AFLite non-biased datasets perform better than the same size random data baseline and full data uniformly.** The trend we see in the baselines is consistent with the recent study Hendrycks *et al.* (2020b).

We control the size of the datasets on which these models are trained to make them comparable. The non-biased samples based model, as denoted by SNLI AFLite,

Good Data Size -> / Bad Data Size	0	18000	36000	54000	72000	90000
0	0	9.56	9.36	9.72	9.47	9.65
18000	16.4	12.95	12.1	11.07	11.19	10.92
36000	15.23	13.5	12.22	12.19	11.7	11.25
54000	15.2	13.35	12.69	12.13	11.71	11.62
72000	14.27	13.39	12.66	12.06	12.02	11.52
90000	14.22	13.92	12.99	12.66	12.19	11.67

Table 4.2: Snli Robustness Scores for Different Quantities of Good and Bad Data.

Train Data/Method	Model Free Text Attack(Accuracy)						Avg.	New Robustness Scores	Old Robustness Scores
	CLARE	Charswap	CheckList	EDA	Emb	Word Net			
SNLI-AFLite92k	62.45	65.15	72.29	56.81	66.2	62.55	64.24	9.05	51.13
SNLI-92k	72.83	75.85	86.57	68.09	76.91	71.01	75.21	12.72	53.92
SNLI+EDA 92k	80.45	77.37	86.57	74.09	80.71	80.22	79.9	14.01	55.06
SNLI Full	76.34	81.33	89.41	72.05	81.94	77.07	79.69	12.64	53.56

Table 4.3: Model Free Text Attack

reports higher zero-shot generalization accuracy.

4.4 SNLI-R

It can be concluded from Figure 4.2 and Figure 4.1, even though AFLite improves OOD generalization, it reduces robustness. To solve the problem for recovering the lost robustness and avoid the wastage of resources problem, we repair the datasets using the adversarial attack as shown in Figure 4.3.

Is SNLI-R helping with robustness? In comparison with SNLI AFLite, i.e., the unbiased part of the dataset SNLI-R shows an increase in robustness through both

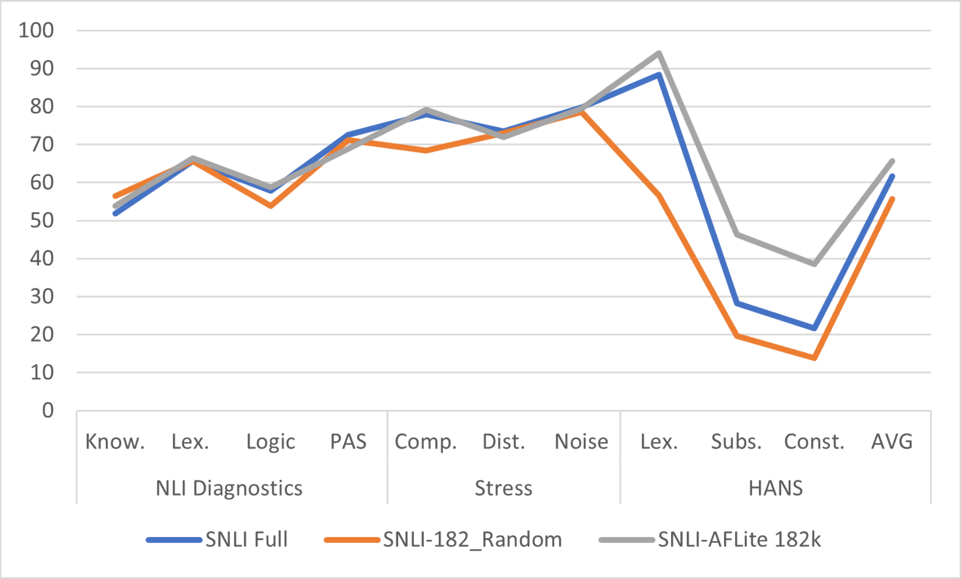


Figure 4.2: Visualization for Zero-shot Snli Accuracy on Ood Datasets

	NLI Diagnostics				Stress			HANS			
	Know.	Lex.	Logic	PAS	Comp.	Dist.	Noise	Lex.	Subs.	Const.	AVG
SNLI Full	51.8	65.7	57.8	**72.6**	77.9	**73.5**	**79.8**	88.4	28.2	21.7	61.74
SNLI-182_Random	**56.4**	65.6	53.9	71.2	68.4	73	78.6	56.6	19.6	13.8	55.71
SNLI-AFLite 182k	53.9	**66.5**	**58.7**	68.9	**79.1**	72	79.5	**94.1**	**46.3**	**38.5**	**65.75**

Table 4.4: Zero-shot Snli Accuracy on Ood Datasets with Roberta Based Trained Model.

model-based and model-free experiments. The results are shown in Table 4.5 and are summarized in Figure 4.4

Is SNLI-R helping with OOD generalization? In comparison with SNLI AFLite, i.e., the unbiased part of the dataset, SNLI-R shows an overall increase in generalization. We observe an increase in average accuracy by 2.66%. The results are shown in Table 4.6 and are summarized in Figure 4.5

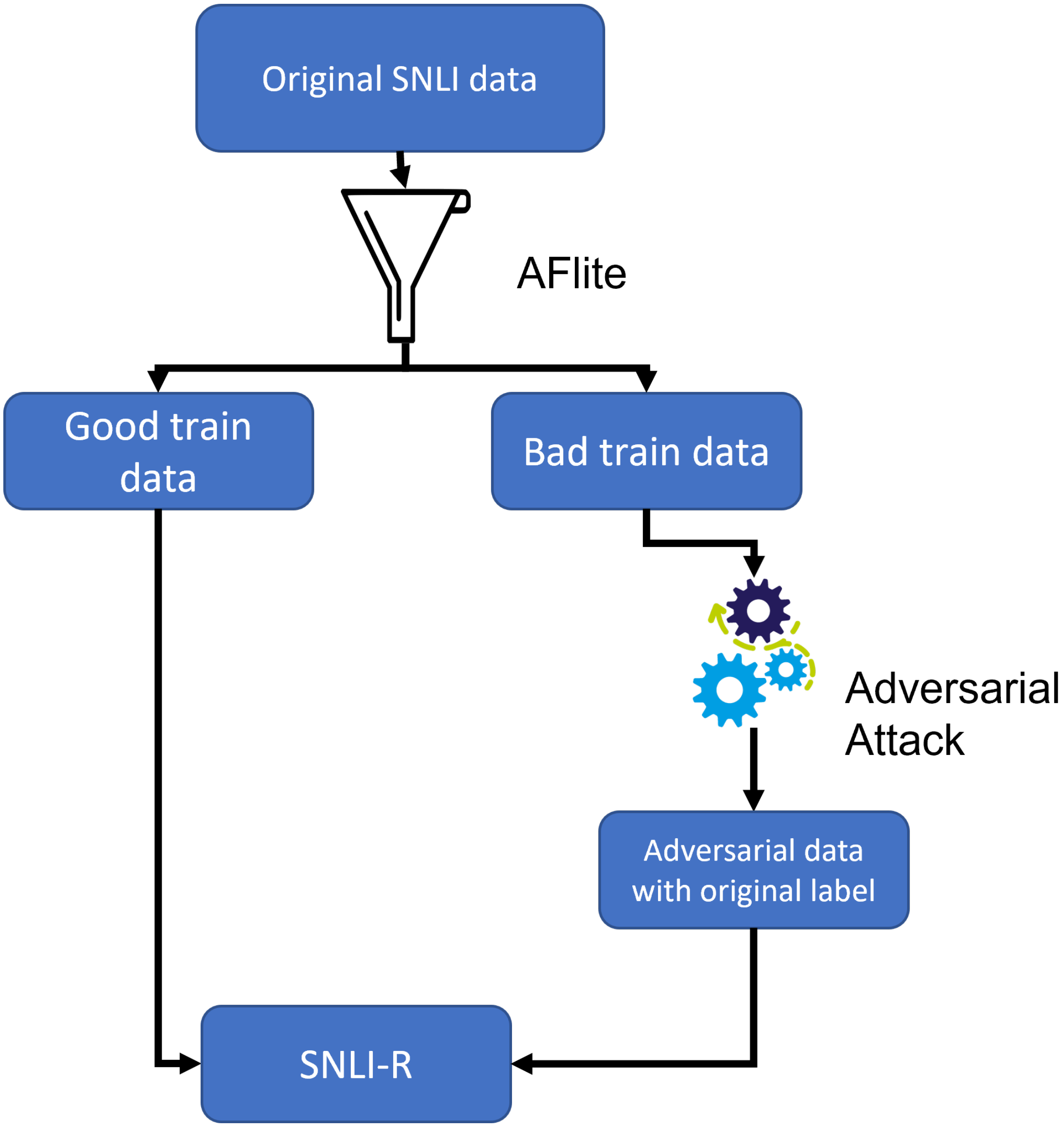


Figure 4.3: Repairing the Bad or Biased Samples to Recover Robustness and Avoid Wastage of Resources.

4.5 Data maps

We further analyse our datasets using the training dynamics i.e. confidence, variability and correctness and creating data maps as defined in section 2.6. The Figure 4.8 shows the original distribution of the SNLI dataset. We can spot a set of hard-

Train Data/Method	Model Free Text Attack(Accuracy)						Model Free Avg	New Robustness Scores	Old Robustness Scores
	CLARE	Charswap	CheckList	EDA	Emb	Word Net			
SNLI-AFLite 92k	62.45	65.15	72.29	56.81	66.2	62.55	64.24	9.05	51.13
SNLI-R 92k	**72.59**	**75.67**	**82.76**	**63.95**	**77.4**	**73.3**	**74.28**	**22.98**	**61.28**

Table 4.5: Snli-r Helps Improve the Robustness.

	ANLI			NLI Diagnostics				Stress			HANS		
	R1	R2	R3	Know.	Lex.	Logic	PAS	Comp.	Dist.	Noise	Lex.	Subs.	Const.	AVG
SNLI-AFLite 92k	30.2	33.2	34.83	39.79	36.68	41.48	45.99	30.93	48.51	52.73	51.8	53.9	51.74	42.44
SNLI-R 92k	28.9	30.6	30.92	45.42	41.58	43.41	49.53	46.65	49.91	57.34	57.63	52.04	52.38	45.1

Table 4.6: Snli-r Helps Improve the Generalization.

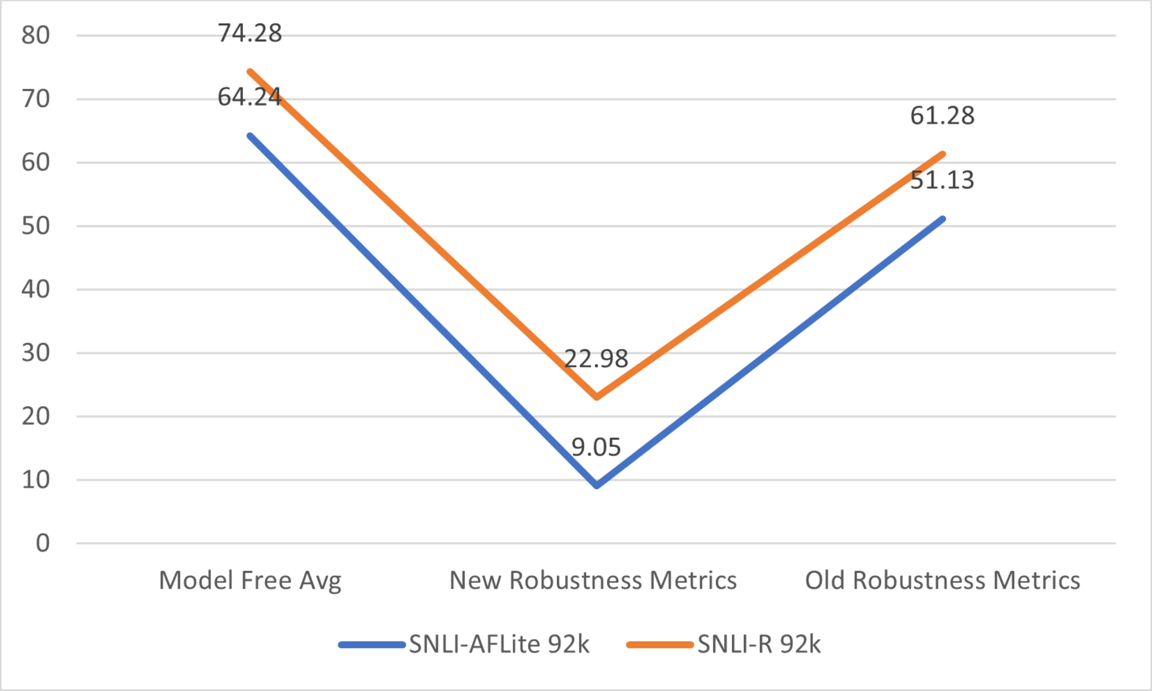


Figure 4.4: Snli-r Helps Improve the Robustness.

to-learn, easy-to-learn and ambiguous samples as defined by the different variations in the training dynamics. The Figure 4.6 corresponds to the non-biased samples as generated by AFLite. Here we see very distinct structure as all the samples come in

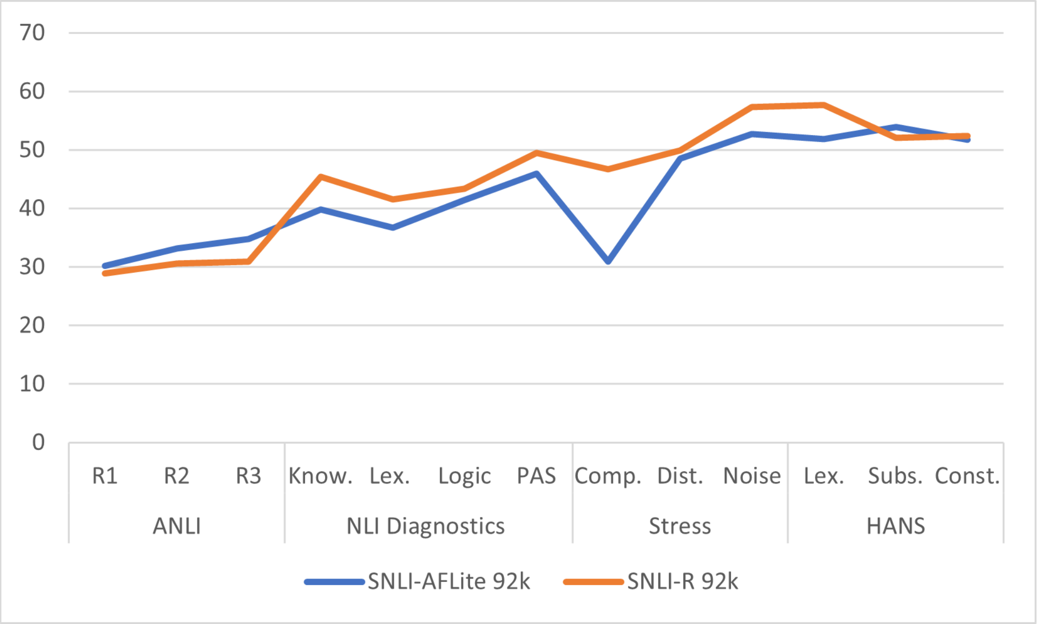


Figure 4.5: Snli-r Helps Improve the O.O.D. Genearlization.

the hard to learn category even though there are relatively easier samples but not ambiguous samples. The figure 4.8 shows the data map for equal number of samples taken from both the adversaries and the non biased samples. We see that the model has more hard to learn samples now along with many easy to learn sample but these samples don't have high correctness, i.e. in each epoch the model's predictions are not stable for these samples.

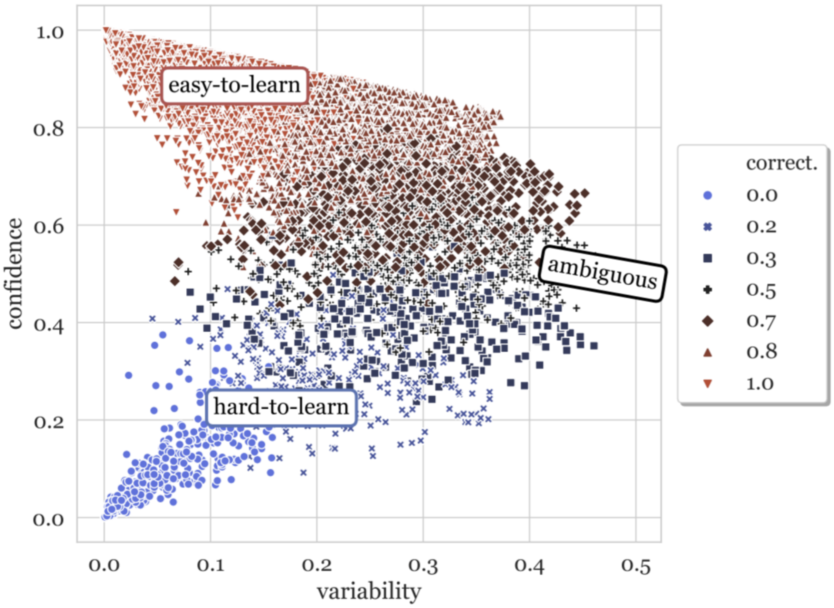


Figure 4.6: Data Map for Snli Dataset Showing Unique Hard to Learn, Easy to Predict and Ambiguous Samples.

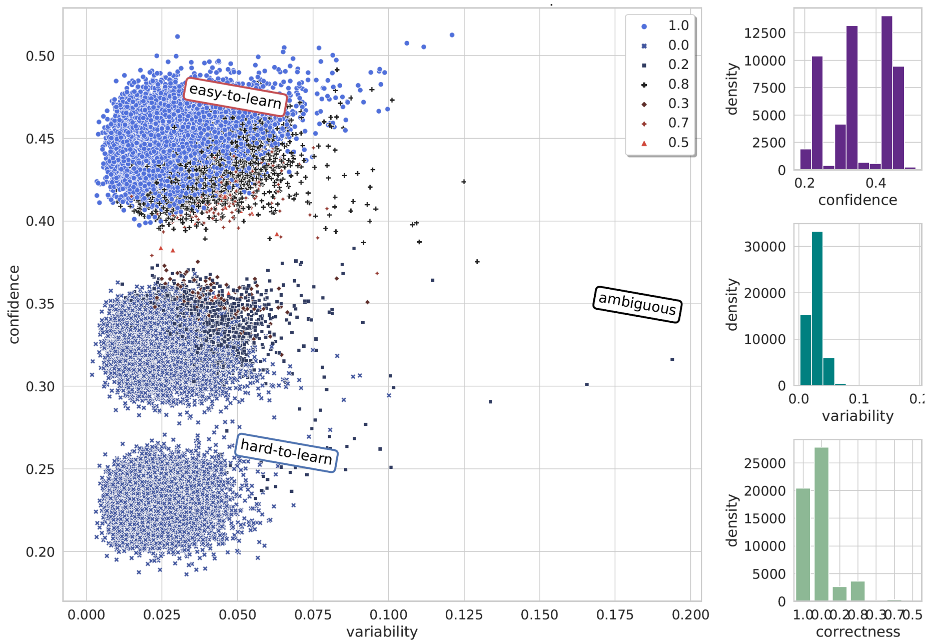


Figure 4.7: Data Map for ”Good” Part of Snli Dataset as Identified by Aflite.

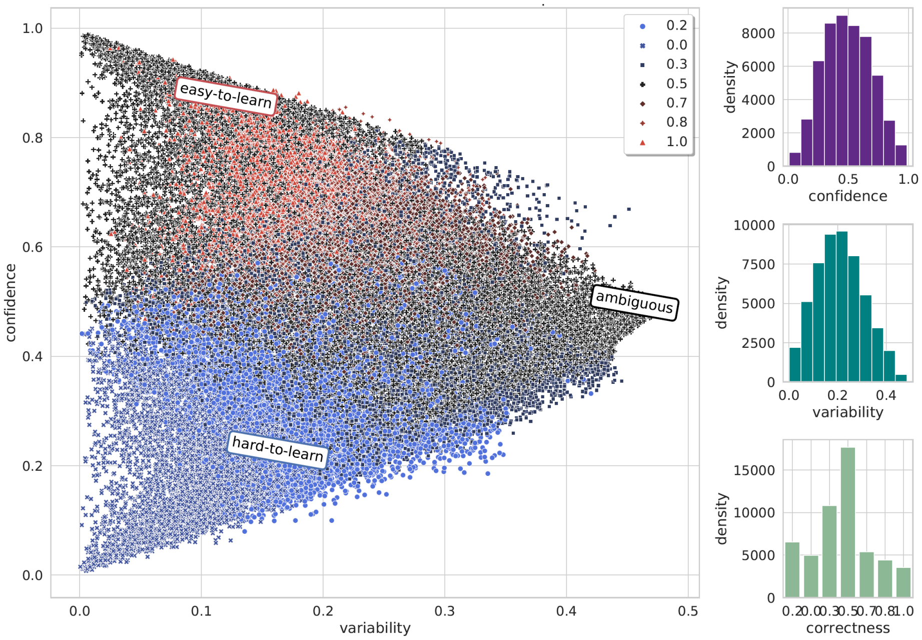


Figure 4.8: Data Map for the Adversaries Generated for the "Bad" Part of Snli Dataset as Identified by Aflite.

REFERENCES

Alegion, "Artificial intelligence and machine learning projects are obstructed by data issues", https://cdn2.hubspot.net/hubfs/3971219/Survey%20Assets%201905/Dimensional%20Research%20Machine%20Learning%20PPT%20Report%20FINAL.pdf (2019).

Angelova, A., Y. Abu-Mostafam and P. Perona, "Pruning training sets for learning of object categories", in "2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)", vol. 1, pp. 494–501 vol. 1 (2005).

Arunkumar, A., S. Mishra, B. Sachdeva, C. Baral and C. Bryan, "Real-time visual feedback for educative benchmark creation: A human-and-metric-in-the-loop workflow", (2020).

Bender, E. M., T. Gebru, A. McMillan-Major and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?", in "Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency", pp. 610–623 (2021).

Bhagavatula, C., R. L. Bras, C. Malaviya, K. Sakaguchi, A. Holtzman, H. Rashkin, D. Downey, W. Yih and Y. Choi, "Abductive commonsense reasoning", in "8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020", (OpenReview.net, 2020), URL https://openreview.net/forum?id=Byg1v1HKDB.

Bolukbasi, T., K. Chang, J. Y. Zou, V. Saligrama and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings", in "Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain", edited by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon and R. Garnett, pp. 4349–4357 (2016).

Bowman, S. R., G. Angeli, C. Potts and C. D. Manning, "A large annotated corpus for learning natural language inference", in "Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)", (Association for Computational Linguistics, 2015a).

Bowman, S. R., G. Angeli, C. Potts and C. D. Manning, "A large annotated corpus for learning natural language inference", in "Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing", pp. 632–642 (Association for Computational Linguistics, Lisbon, Portugal, 2015b), URL https://www.aclweb.org/anthology/D15-1075.

Bras, R. L., S. Swayamdipta, C. Bhagavatula, R. Zellers, M. E. Peters, A. Sabharwal and Y. Choi, "Adversarial filters of dataset biases", arXiv preprint arXiv:2002.04108 (2020a).

Bras, R. L., S. Swayamdipta, C. Bhagavatula, R. Zellers, M. E. Peters, A. Sabharwal and Y. Choi, “Adversarial filters of dataset biases”, in “Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event”, vol. 119 of *Proceedings of Machine Learning Research*, pp. 1078–1088 (PMLR, 2020b), URL http://proceedings.mlr.press/v119/bras20a.html.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, “Language models are few-shot learners”, in “Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual”, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin (2020).

Chen, T., Z. Jiang, A. Poliak, K. Sakaguchi and B. Van Durme, “Uncertain natural language inference”, in “Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics”, pp. 8772–8779 (Association for Computational Linguistics, Online, 2020), URL https://www.aclweb.org/anthology/2020.acl-main.774.

Deng, J., W. Dong, R. Socher, L. Li, K. Li and F. Li, “Imagenet: A large-scale hierarchical image database”, in “2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA”, pp. 248–255 (IEEE Computer Society, 2009), URL https://doi.org/10.1109/CVPR.2009.5206848.

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding”, in “Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)”, pp. 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), URL https://www.aclweb.org/anthology/N19-1423.

Dua, D., Y. Wang, P. Dasigi, G. Stanovsky, S. Singh and M. Gardner, “DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs”, in “Proc. of NAACL”, (2019).

Eykholt, K., I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno and D. Song, “Robust physical-world attacks on deep learning visual classification”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 1625–1634 (2018).

Fang, Z., T. Gokhale, P. Banerjee, C. Baral and Y. Yang, “Video2Commonsense: Generating commonsense descriptions to enrich video captioning”, in “Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)”, pp. 840–860 (Association for Computational Linguistics, Online, 2020), URL https://www.aclweb.org/anthology/2020.emnlp-main.61.

Garg, S. and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification", Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) URL `http://dx.doi.org/10.18653/v1/2020.emnlp-main.498` (2020a).

Garg, S. and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification. arxiv 2020", arXiv preprint cs.CL/2004.01970 (2020b).

Goodfellow, I. J., J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples", (2015).

Gururangan, S., S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman and N. A. Smith, "Annotation artifacts in natural language inference data", in "Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)", pp. 107–112 (Association for Computational Linguistics, New Orleans, Louisiana, 2018), URL `https://www.aclweb.org/anthology/N18-2017`.

Hendrycks, D., S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo *et al.*, "The many faces of robustness: A critical analysis of out-of-distribution generalization", arXiv preprint arXiv:2006.16241 (2020a).

Hendrycks, D. and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations", arXiv preprint arXiv:1903.12261 (2019).

Hendrycks, D., X. Liu, E. Wallace, A. Dziedzic, R. Krishnan and D. Song, "Pretrained transformers improve out-of-distribution robustness", arXiv preprint arXiv:2004.06100 (2020b).

Hoeffding, W., "Probability inequalities for sums of bounded random variables", Journal of the American Statistical Association **58**, 301, 13–30 (1963).

Hornik, K., M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators", Neural networks **2**, 5, 359–366 (1989).

Jia, R. and P. Liang, "Adversarial examples for evaluating reading comprehension systems", in "Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing", pp. 2021–2031 (Association for Computational Linguistics, Copenhagen, Denmark, 2017a), URL `https://www.aclweb.org/anthology/D17-1215`.

Jia, R. and P. Liang, "Adversarial examples for evaluating reading comprehension systems", CoRR **abs/1707.07328**, URL `http://arxiv.org/abs/1707.07328` (2017b).

Jia, R., A. Raghunathan, K. Göksel and P. Liang, "Certified robustness to adversarial word substitutions", in "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)", pp. 4129–4142 (Association for Computational Linguistics, Hong Kong, China, 2019), URL `https://www.aclweb.org/anthology/D19-1423`.

Jiang, H. and O. Nachum, “Identifying and correcting label bias in machine learning”, in “International Conference on Artificial Intelligence and Statistics”, pp. 702–712 (PMLR, 2020).

Jin, D., Z. Jin, J. T. Zhou and P. Szolovits, “Is bert really robust? natural language attack on text classification and entailment”, arXiv preprint arXiv:1907.11932 (2019a).

Jin, D., Z. Jin, J. T. Zhou and P. Szolovits, “Is BERT really robust? natural language attack on text classification and entailment”, CoRR **abs/1907.11932**, URL `http://arxiv.org/abs/1907.11932` (2019b).

Jones, E., R. Jia, A. Raghunathan and P. Liang, “Robust encodings: A framework for combating adversarial typos”, in “Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics”, pp. 2752–2765 (2020).

Kamath, A., R. Jia and P. Liang, “Selective question answering under domain shift”, arXiv preprint arXiv:2006.09462 (2020).

Kaushik, D. and Z. C. Lipton, “How much reading does reading comprehension require? a critical investigation of popular benchmarks”, in “Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing”, pp. 5010–5015 (Association for Computational Linguistics, Brussels, Belgium, 2018), URL `https://www.aclweb.org/anthology/D18-1546`.

Krishna, R., K. Hata, F. Ren, L. Fei-Fei and J. C. Niebles, “Dense-captioning events in videos”, in “International Conference on Computer Vision (ICCV)”, (2017).

Kurakin, A., I. J. Goodfellow and S. Bengio, “Adversarial examples in the physical world”, CoRR **abs/1607.02533**, URL `http://arxiv.org/abs/1607.02533` (2016).

Li, D., Y. Zhang, H. Peng, L. Chen, C. Brockett, M.-T. Sun and B. Dolan, “Contextualized perturbation for textual adversarial attack”, arXiv preprint arXiv:2009.07502 (2020a).

Li, L., R. Ma, Q. Guo, X. Xue and X. Qiu, “Bert-attack: Adversarial attack against bert using bert”, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) URL `http://dx.doi.org/10.18653/v1/2020.emnlp-main.500` (2020b).

Li, Y., Y. Li and N. Vasconcelos, “Resound: Towards action recognition without representation bias”, in “Proceedings of the European Conference on Computer Vision (ECCV)”, pp. 513–528 (2018).

Li, Y. and N. Vasconcelos, “Repair: Removing representation bias by dataset resampling”, in “Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition”, pp. 9572–9581 (2019).

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach", arXiv preprint arXiv:1907.11692 (2019).

Marivate, V. and T. Sefara, "Improving short text classification through global augmentation methods", in "Machine Learning and Knowledge Extraction", edited by A. Holzinger, P. Kieseberg, A. M. Tjoa and E. Weippl, pp. 385–399 (Springer International Publishing, Cham, 2020).

McCoy, R. T., E. Pavlick and T. Linzen, "Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference", CoRR **abs/1902.01007**, URL `http://arxiv.org/abs/1902.01007` (2019).

Mehrabi, N., F. Morstatter, N. Saxena, K. Lerman and A. Galstyan, "A survey on bias and fairness in machine learning", arXiv preprint arXiv:1908.09635 (2019).

Min, Y., L. Chen and A. Karbasi, "The curious case of adversarially robust models: More data can help, double descend, or hurt generalization", CoRR **abs/2002.11080**, URL `https://arxiv.org/abs/2002.11080` (2020).

Mishra, S. and A. Arunkumar, "How robust are model rankings: A leaderboard customization approach for equitable evaluation", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 35, pp. 13561–13569 (2021).

Mishra, S., A. Arunkumar, C. Bryan and C. Baral, "Our evaluation metric needs an update to encourage generalization", ArXiv **abs/2007.06898** (2020a).

Mishra, S., A. Arunkumar, B. Sachdeva, C. Bryan and C. Baral, "Dqi: A guide to benchmark evaluation", arXiv preprint arXiv:2008.03964 (2020b).

Mishra, S., A. Arunkumar, B. S. Sachdeva, C. Bryan and C. Baral, "Dqi: Measuring data quality in nlp", ArXiv **abs/2005.00816** (2020c).

Mishra, S., D. Khashabi, C. Baral and H. Hajishirzi, "Natural instructions: Benchmarking generalization to new tasks from natural language instructions", arXiv preprint arXiv:2104.08773 (2021).

Mishra, S., A. Mitra, N. Varshney, B. S. Sachdeva and C. Baral, "Towards question format independent numerical reasoning: A set of prerequisite tasks", ArXiv **abs/2005.08516** (2020d).

Mishra, S. and B. S. Sachdeva, "Do we need to create big datasets to learn a task?", in "Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing", pp. 169–173 (Association for Computational Linguistics, Online, 2020), URL `https://aclanthology.org/2020.sustainlp-1.23`.

Mitchell, T. M., *The need for biases in learning generalizations* (Department of Computer Science, Laboratory for Computer Science Research . . . , 1980).

Morris, J., E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin and Y. Qi, "Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp", Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations URL http://dx.doi.org/10.18653/v1/2020.emnlp-demos.16 (2020).

Naik, A., A. Ravichander, N. Sadeh, C. Rose and G. Neubig, "Stress test evaluation for natural language inference", in "Proceedings of the 27th International Conference on Computational Linguistics", pp. 2340–2353 (Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018), URL https://www.aclweb.org/anthology/C18-1198.

Nie, Y., A. Williams, E. Dinan, M. Bansal, J. Weston and D. Kiela, "Adversarial nli: A new benchmark for natural language understanding", arXiv preprint arXiv:1910.14599 (2019).

Nie, Y., A. Williams, E. Dinan, M. Bansal, J. Weston and D. Kiela, "Adversarial NLI: A new benchmark for natural language understanding", in "Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics", pp. 4885–4901 (Association for Computational Linguistics, Online, 2020), URL https://aclanthology.org/2020.acl-main.441.

Niven, T. and H. Kao, "Probing neural network comprehension of natural language arguments", CoRR **abs/1907.07355**, URL http://arxiv.org/abs/1907.07355 (2019).

Poliak, A., J. Naradowsky, A. Haldar, R. Rudinger and B. Van Durme, "Hypothesis only baselines in natural language inference", in "Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics", pp. 180–191 (Association for Computational Linguistics, New Orleans, Louisiana, 2018), URL https://www.aclweb.org/anthology/S18-2023.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language models are unsupervised multitask learners", (2019).

Rajpurkar, P., J. Zhang, K. Lopyrev and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text", in "Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing", pp. 2383–2392 (Association for Computational Linguistics, Austin, Texas, 2016), URL https://www.aclweb.org/anthology/D16-1264.

Ribeiro, M. T., T. Wu, C. Guestrin and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with CheckList", in "Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics", pp. 4902–4912 (Association for Computational Linguistics, Online, 2020a), URL https://www.aclweb.org/anthology/2020.acl-main.442.

Ribeiro, M. T., T. Wu, C. Guestrin and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with checklist", CoRR **abs/2005.04118** (2020b).

Rogers, A., “Changing the world by changing the data”, ArXiv **abs/2105.13947** (2021).

Sagawa, S., A. Raghunathan, P. W. Koh and P. Liang, “An investigation of why overparameterization exacerbates spurious correlations”, in “International Conference on Machine Learning”, pp. 8346–8356 (PMLR, 2020).

Sakaguchi, K., R. L. Bras, C. Bhagavatula and Y. Choi, “Winogrande: An adversarial winograd schema challenge at scale”, arXiv preprint arXiv:1907.10641 (2019).

Sakaguchi, K., R. L. Bras, C. Bhagavatula and Y. Choi, “Winogrande: An adversarial winograd schema challenge at scale”, in “AAAI”, (2020).

Sambasivan, N., S. Kapania, H. Highfill, D. Akrong, P. K. Paritosh and L. Aroyo, ““everyone wants to do the model work, not the data work”: Data cascades in high-stakes ai”, Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (2021).

Swayamdipta, S., R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith and Y. Choi, “Dataset cartography: Mapping and diagnosing datasets with training dynamics”, arXiv preprint arXiv:2009.10795 (2020).

Talmor, A. and J. Berant, “Multiqa: An empirical investigation of generalization and transfer in reading comprehension”, in “Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics”, pp. 4911–4921 (2019).

Tan, M. and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks”, in “International Conference on Machine Learning”, pp. 6105–6114 (PMLR, 2019).

Torralba, A. and A. A. Efros, “Unbiased look at dataset bias”, in “CVPR 2011”, pp. 1521–1528 (IEEE, 2011).

Utama, P. A., N. S. Moosavi and I. Gurevych, “Towards debiasing nlu models from unknown biases”, in “Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)”, pp. 7597–7610 (2020).

Vapnik, V., *The nature of statistical learning theory* (Springer science & business media, 2013).

Varshney, N., S. Mishra and C. Baral, “It’s better to say ”i can’t answer” than answering incorrectly: Towards safety critical nlp systems”, ArXiv **abs/2008.09371** (2020).

Vigen, T., *Spurious correlations* (Hachette Books, 2015).

Wang, A., A. Singh, J. Michael, F. Hill, O. Levy and S. R. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding”, CoRR **abs/1804.07461**, URL `http://arxiv.org/abs/1804.07461` (2018a).

Wang, T., J. Zhao, M. Yatskar, K.-W. Chang and V. Ordonez, “Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations”, in “Proceedings of the IEEE/CVF International Conference on Computer Vision”, pp. 5310–5319 (2019).

Wang, T., J.-Y. Zhu, A. Torralba and A. A. Efros, “Dataset distillation”, arXiv preprint arXiv:1811.10959 (2018b).

Wei, J. and K. Zou, “EDA: Easy data augmentation techniques for boosting performance on text classification tasks”, 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)”, pp. 6382–6388 (Association for Computational Linguistics, Hong Kong, China, 2019), URL `https://www.aclweb.org/anthology/D19-1670`.

Williams, A., N. Nangia and S. Bowman, “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)”, pp. 1112–1122 (Association for Computational Linguistics, 2018), URL `http://aclweb.org/anthology/N18-1101`.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, “Huggingface’s transformers: State-of-the-art natural language processing”, arXiv preprint arXiv:1910.03771 (2019).

Wu, M., N. S. Moosavi, A. Rücklé and I. Gurevych, “Improving qa generalization by concurrent modeling of multiple biases”, arXiv preprint arXiv:2010.03338 (2020).

Zellers, R., Y. Bisk, A. Farhadi and Y. Choi, “From recognition to cognition: Visual commonsense reasoning”, in “IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019”, pp. 6720–6731 (Computer Vision Foundation / IEEE, 2019a).

Zellers, R., Y. Bisk, R. Schwartz and Y. Choi, “Swag: A large-scale adversarial dataset for grounded commonsense inference”, (2018).

Zellers, R., A. Holtzman, Y. Bisk, A. Farhadi and Y. Choi, “Hellaswag: Can a machine really finish your sentence?”, in “Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics”, pp. 4791–4800 (2019b).

Zhao, J., T. Wang, M. Yatskar, V. Ordonez and K.-W. Chang, “Men also like shopping: Reducing gender bias amplification using corpus-level constraints”, in “Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing”, pp. 2979–2989 (Association for Computational Linguistics, Copenhagen, Denmark, 2017), URL `https://www.aclweb.org/anthology/D17-1323`.

Zhao, Z., D. Dua and S. Singh, “Generating natural adversarial examples”, (2018).