

# A Survey on Out-of-Distribution Evaluation of Neural NLP Models

Xinzhe Li<sup>1</sup>, Ming Liu<sup>1</sup>, Shang Gao<sup>1</sup> and Wray Buntine<sup>2</sup>

<sup>1</sup>Deakin University

<sup>2</sup>VinUniversity

{lixinzhe, m.liu, shang.gao}@deakin.edu.au, wray.b@vinuni.edu.vn

## Abstract

Adversarial robustness, domain generalization and dataset biases are three active lines of research contributing to out-of-distribution (OOD) evaluation on neural NLP models. However, a comprehensive, integrated discussion of the three research lines is still lacking in the literature. In this survey, we 1) compare the three lines of research under a unifying definition; 2) summarize the data-generating processes and evaluation protocols for each line of research; and 3) emphasize the challenges and opportunities for future work.

## 1 Introduction

Neural NLP models have been reported for their superhuman performance on many language understanding and generation tasks, such as sentiment analysis and machine reading comprehension (MRC). Recent studies show that these models lack human-level understanding of language since they are vulnerable beyond in-distribution (ID) test data and fail to generalize on out-of-distribution (OOD) data, such as perturbed examples under adversarial attacks [Ebrahimi *et al.*, 2018], text from different domains [Hendrycks *et al.*, 2020] and examples against dataset biases [Gururangan *et al.*, 2018], which lie in the three mainstream lines of research: adversarial robustness, domain generalization and dataset biases.

Despite recent progress in each of the three research lines, there has not been a survey that comprehensively reviews and investigates the connections between these lines. We have the following motivations for such an investigation: 1) explaining some phenomena across the three lines of research. For example, methods designed to improve model generalization on one OOD type can defend against other types of OOD data [Yi *et al.*, 2021] but sometimes degrade model performance on other OOD types [Bras *et al.*, 2020; Gokhale *et al.*, 2022]; 2) encouraging future researchers to perform a comprehensive OOD evaluation while proposing their new methods; 3) suggesting that all the OOD types be addressed thoroughly in future research.

In this survey, we first unify the three research lines as the study of distribution shift, which gives us a starting point to find connections between them in terms of shifted features

(§2). Furthermore, we summarize their differences in OOD data generation and evaluation protocols (§3). In particular, we categorize different methods to generate OOD data, including data with natural domain shift (NDS) for domain generalization, debiased data for evaluating dataset biases and adversarial examples for adversarial robustness. We then categorize evaluation protocols for the OOD data into two classes: data-based and method-based evaluation. Finally, we identify opportunities based on the discussion of connections and differences between the three OOD types, including developing a comprehensive benchmark for OOD evaluation, advising caution about breaking the assumptions of covariate shift, improving general OOD performance and using the detection methods across different OOD types. Also, we demonstrate the gap in adversarial robustness for future work (§4).

## 2 Definition

This section first introduces a comprehensive definition of distribution shift across the three research lines and then shows how the definition aids in elucidating the interconnections regarding the shifted features.

### 2.1 Distribution Shift

Deep Neural Network (DNN)  $f(\theta)$  can achieve remarkable or even superhuman performance on NLP benchmarks, such as GLUE with in-distribution (ID) test data  $\mathcal{D}_{\text{test}} \sim \mathbb{P}_0$ , when  $f(\theta)$  is trained on data sampled from the data-generating distribution (an unknown distribution under a data-generating process)  $\mathcal{D}_{\text{train}} \sim \mathbb{P}_0$ . However, the well-trained model often fails to generalize to various unforeseen examples. Formally, the unforeseen data (or OOD data)  $\tilde{\mathcal{D}}$  can be characterized by a shifted distribution  $\tilde{\mathbb{P}}$ . The distribution shift includes covariate shifts (input distribution shifts)  $\mathbb{P}_0(\mathcal{X}) \neq \tilde{\mathbb{P}}(\mathcal{X})$  and label distribution shifts  $\mathbb{P}_0(\mathcal{Y}) \neq \tilde{\mathbb{P}}(\mathcal{Y})$ , where  $\mathcal{X}$  is the input space, and  $\mathcal{Y}$  is the ground-truth output space. We unify the three lines of research as a study of covariate shifts. For brevity, the input notation  $\mathcal{X}$  is sometimes omitted when referring to it, as exemplified by the use of  $\mathbb{P}_0$  to represent  $\mathbb{P}_0(\mathcal{X})$ .

**Domain generalization and dataset biases.** Real-life documents in different domains, e.g., news and fiction text [Hendrycks *et al.*, 2020], are typically characterized by systematically distinct properties and originate in different data-

generating distributions. Domain generalization investigates such natural domain shifts, and relevant literature often uses domain shift and distribution shift interchangeably. Some literature [Arora *et al.*, 2021; Gokhale *et al.*, 2022] also defines that data with particular biases and those without such biases come from different domains. Therefore, the studies of domain generalization and dataset biases can be intuitively defined in the framework of distribution shifts.

**Adversarial robustness: from robustness to distribution shift.** Initially, adversarial methods evaluate model robustness against small perturbations in the worst-case scenarios, especially for continuous data like images [Goodfellow *et al.*, 2015]. Hence, prior studies typically separate adversarial robustness from the study of domain generalization and distribution shift. We can unify adversarial robustness as a study of distribution/domain shift. Conceptually, we can define the shifted distribution in the adversarial setting as a distribution around the original distribution  $\mathbb{P}_0$ . We perform a perturbation on text  $x \in \mathbb{P}_0(\mathcal{X})$  to simulate OOD data from the shifted distribution. In other words, adversarial perturbations encode subtle domain shifts, as discussed in §2.3, and the shifted domains are characterized by the adversarial methods summarized in §3.3.

## 2.2 Shifted Features

To concretize the shifted features, we first define task-relevant features  $s_{\text{rel}}$  and task-irrelevant features  $s_{\text{ir}}$ , which satisfy  $\mathbb{P}(s_{\text{rel}}|y) \neq \mathbb{P}(s_{\text{rel}})$  and  $\mathbb{P}(s_{\text{ir}}|y) = \mathbb{P}(s_{\text{ir}})$ , respectively.  $\mathbb{P}$  is an arbitrary distribution.

**Background features and semantic features.** When  $\mathbb{P}$  is the ground-truth distribution  $\mathbb{P}_{\text{true}}$ , Arora *et al.*, [2021] name  $s_{\text{rel}}$  as *semantic features* since most realistic NLP tasks address the semantics of text, and  $s_{\text{ir}}$  as *background features*, which can be the syntax, writing style or task-irrelevant text.

**Biased features are task-irrelevant features under  $\mathbb{P}_{\text{true}}$  despite being learned as task-relevant features from  $\mathbb{P}_0$ .**  $\mathbb{P}_0(\mathcal{X})$  can be factorized into the conditional distributions of generalized features  $g(\mathcal{X})$  and **biased features**  $b(\mathcal{X})$ .

$$\mathbb{P}_0(\mathcal{X}) = \sum_y \mathbb{P}_0(g(\mathcal{X})|y)\mathbb{P}_0(b(\mathcal{X})|y)\mathbb{P}_0(y), \quad (1)$$

assuming conditional independence between  $b(\mathcal{X})$  and  $g(\mathcal{X})$  given  $y$ . By definition,  $g(\mathcal{X})$  should always be task-relevant features in the true distribution  $\mathbb{P}_{\text{true}}$ , i.e.,  $g(\mathcal{X}) \in s_{\text{rel}}$ , while biased features  $b(\mathcal{X})$  are only predictive for ID dataset  $\mathcal{D} \sim \mathbb{P}_0$ . We can express  $b(\mathcal{X})$  as  $\{s|s \in \mathbb{P}_0 \text{ and } s \notin \mathbb{P}\}$ .

## 2.3 Shifted Features in Three OOD Types

This section shows how each OOD type can potentially shift the three kinds of features.

**Shifted features on adversarial examples.** Shifted features of adversarial examples depend on the attack methods. White-box and grey-box methods compute gradients or output logits, respectively, while black-box methods can only query final predictions. With the guidance of gradients, white-box methods tend to shift the biased features since the

input gradients can help identify the biased features for perturbations [Ilyas *et al.*, 2019; Wallace *et al.*, 2019a]. Perturbation types largely affect whether it is a semantic shift or background shift (See §3.3). For example, character editing may lead to unseen words, e.g., “wonderful” to “wonderful”, which belongs to semantic shift. In contrast, black-box methods explicitly change background features at the sentence level. For example, Qi *et al.*, [2021] paraphrase with different writing styles.

**Shifted features on debiased data.** Debiased data, by definition, is generated by removing biased features. However, the generation process often encodes either background features  $s_{\text{ir}}$  or semantic features  $s_{\text{rel}}$ . For example, debiased data generated by syntactic templates [McCoy *et al.*, 2019] encode background shift, while MRC data requiring numerical reasoning to avoid dataset biases [Naik *et al.*, 2018] change semantic features.

**Shifted features on NDS data.** We classify natural domain shift into two categories, namely genres and sub-populations. Both background and semantic shifts can occur in both categories, such as different vocabularies associated with sub-populations or genres [Barrett *et al.*, 2019]. For example, for sentiment analysis, Tweets tend to express sentiments with emotions compared to the genre of movie reviews, and sub-populations can differ in the way to express emotions [Oprea and Magdy, 2020]. Furthermore, DNNs can easily learn background features for each sub-population or genre as biased features. Writing styles can also serve as another background feature that changes across genres [Arora *et al.*, 2021], e.g., text from fiction books to Wikipedia content.

## 3 OOD Performance Evaluation

This section compares the concrete methodologies of OOD data generation and evaluation protocols for each line of research. Their differences are summarized in Table 1.

### 3.1 NDS Data Generation

NDS data can be generated from different genres or sub-populations. The former can be acquired from various data sources, whereas the latter involves partitioning data into sub-populations based on attributes of entities or individuals. Diverse sets of data genres may display a fusion of unique textual styles, syntactic structures, and lexicons.

**Genres.** Different genres refer to text written by different authors or annotators or for different audiences, or come from different data sources. For example, academic papers are written in formal language while online reviews on Yelp contain non-standard orthography. Since a training dataset typically comes from a specific data source and contains only one genre, even well-trained models generalize poorly to different genres. Although some datasets are created to include multiple genres of text, models still perform worse beyond the coverage of these genres. For example, MultiNLI for NLI consists of text from ten distinct genres. However, even the large pretrained language models cannot generalize well to a different genre [Arora *et al.*, 2021], such as SNLI from image captions or WNLI from fiction books. Take text classification as another example. Classifiers cannot perform well on

	OOD Data Generation		Evaluation	
	Generated by	Required Knowledge	Types	Metrics
Adversarial robustness	Auto	Model	Method-based	ASR; query #
Adversarial transferability	Auto/human	Base model	Data-based	ID metrics
Domain generalization	Existing	Domain	Data-based	ID metrics
Dataset biases	Auto/human	Biases/NA	Data-based	ID metrics

Table 1: Comparison of OOD data generation and evaluation. ASR: attack success rate; query #: number of queries; ID metrics: the same aggregate metrics for ID test data, e.g., accuracy for classifiers.

Biases	Tasks	Related Works
Lexical correlations	TC	Geva et al., [2019]
	NLI	Hypothesis-only bias [Gururangan et al., 2018]
	MRC	Question word matching [Sugawara et al., 2018]
	Fact checking	Claim-only bias [Schuster et al., 2019]
Lexical overlap	Machine Translation	Word disambiguation bias [Emelin et al., 2020]
	NLI	Premise-hypothesis overlap [McCoy et al., 2019]
	MRC	Context matching [Sugawara et al., 2018]
Positional bias	Paraphrase	Zhang et al., [2019]
	Summarization	Kedzie et al., [2018]

Table 2: Dataset biases for various NLP tasks. TC: Text Classification; NLI: Natural Language Inference; MRC: Machine Reading Comprehension; Paraphrase: Paraphrase Identification.

Methods	Tasks	Debiased Datasets
Biased inspired	TC	c-IMDB [Kaushik et al., 2020]
	NLI	HANS [McCoy et al., 2019], Stress Text [Naik et al., 2018]
	MRC	Para-SQUAD [Lai et al., 2021] Adv-SQUAD [Jia and Liang, 2017]
	Paraphrase	PAWS [Zhang et al., 2019]
	Fact Checking	FEVER-Symmetric [Schuster et al., 2019]
Systematic	NLI	SNLI-AFLITE [Bras et al., 2020]
	Reasoning	Winogrande [Sakaguchi et al., 2020]
	CGI	Swag [Zellers et al., 2018]

Table 3: Debiased datasets for various NLP tasks. TC: Text Classification; NLI: Natural Language Inference; MRC: Machine Reading Comprehension; Paraphrase: Paraphrase Identification; Reasoning: Commonsense Reasoning; CGI: Commonsense Grounded Inference.

text from disjoint annotators, e.g., SST data from experts v.s. lengthy IMDB reviews from laymen [Hendrycks et al., 2020; Arora et al., 2021]. Also, MRC models trained on Wikipedia hardly generalize on data from New York Times articles, Reddit posts, and Amazon product reviews.

**Sub-populations.** Multiple studies identify NDS data by dividing data into sub-populations according to different attributes of objects or individuals obtained from metadata. Arora et al., [2021] select reviews from different businesses, e.g., restaurant reviews v.s. movie reviews. For the same business, Hendrycks et al., [2020] split them according to different types of products (e.g., women’s clothing, shoes) or restaurants (e.g., American v.s. Chinese restaurants). Furthermore, model performance can vary on data with different demographics, which is closely connected to another line of

research about fairness. For example, Borkan et al., [2019] find the worst test accuracy on non-toxic comments from the black population.

### 3.2 Debiased Data Generation

In this section, we first divide dataset biases into two categories and introduce a text summarization dataset bias that does not fall under either one. We then classify data-generating methods into two categories according to the knowledge of dataset biases. Table 2 and 3 summarize dataset biases and debiased datasets for various NLP tasks, respectively.

#### Dataset Biases

**Lexical correlations.** Words frequently appearing in the training examples of a particular class can be predictive bi-

ases for that class. For example, when NLI annotators create a majority of contradictory hypotheses by negating premises, models learn the spurious correlation between negation words (‘no’, ‘never’) and labels (*hypothesis-only reliance* [Gururangan *et al.*, 2018]). The success of class-wise adversarial attacks comes from exploring these predictive words to some extent [Wallace *et al.*, 2019a]. Similarly, claim-only models can perform well on ID test data without the context of evidence for fact-checking due to predictive words [Schuster *et al.*, 2019]. MRC models can find the answer in a given paragraph by matching the question type, e.g., “November 2014” for a “when” question (question word matching [Lai *et al.*, 2021] or entity type matching [Sugawara *et al.*, 2018]), and hence they are easily distracted to the wrong answers on OOD data [Jia and Liang, 2017]. Emelin *et al.*, [2020] observe that machine translation systems have a tendency to disambiguate words based on the sense that occurs most frequently in the training data.

**Lexical overlap.** Lexical overlap is another predictive indicator for sentence-pair classification. For example, the high overlap between the premise and hypothesis leads to the “entailment” prediction from NLI models (premise-hypothesis overlap [McCoy *et al.*, 2019]). The classifiers for paraphrase identification also tend to predict highly overlapping sentences as paraphrases [Zhang *et al.*, 2019]. Besides, MRC models can locate the sentence with maximum overlap words in the paragraph (context matching [Sugawara *et al.*, 2018]) and then use question word matching to extract the correct answer.

**Positional biases for text summarization.** News or journal articles tend to summarize content in the lead paragraphs (positional bias or layout bias). Specifically, Kedzie *et al.*, [2018] show that 88.6% reference summaries from the training examples of NYT come from the first half of documents (69% for DUC, 71.7% for CNN/DM).

### Methodologies

There are two types of methods to generate debiased data: 1) breaking the spurious correlations; 2) filtering the examples containing superficial patterns automatically without recognizing any specific dataset biases.

**Bias-informed approaches.** Many studies generate debiased data by reversing the correlations between dataset biases and labels for text classification. For natural language inference (NLI), McCoy *et al.*, [2019] specify linguistic phenomena (constituent and subsequent heuristics) behind lexical overlap. They devise heuristics-based syntactic templates to create sentence pairs that have high lexical overlap but contradict each other. Kaushik *et al.*, [2020] construct counterfactual examples by breaking spurious correlations in sentiment analysis datasets (i.e., IMDB) and NLI datasets. They annotate the text into the targeted label by largely keeping the original text. In this way, the dataset biases of the original label are kept. To break the lexical overlap between paraphrase pairs, Zhang *et al.*, [2019] create non-paraphrase sentences via word scrambling. This method changes the meanings of paraphrases but keeps the overlapping words. For MRC, Jia and Liang [2017] add a sentence containing words overlap-

ping with the question into paragraphs and find that models tend to select answers from the sentence. They also combine the bias-inspired approach with a grey-box attack to generate adversarial examples. Gardner *et al.*, [2020] rely entirely on experts who have the knowledge of dataset biases to generate debiased data.

**Systematic approaches.** Debiased data can also be generated by bias mitigation techniques without knowing dataset biases. For example, Bras *et al.*, [2020] use a simple classifier to identify biased examples and generate debiased datasets for NLI, commonsense reasoning and grounded commonsense inference, respectively. Although some works try to train robust models with such filtered data, filtering always leads to significant drops on ID data and even OOD data [Gokhale *et al.*, 2022]. Therefore, it can be more practical to use them for OOD evaluation.

### 3.3 Adversarial Example Generation

Given a well-trained model, adversarial examples are generated by modifying the given text (or reference data) to make the model output wrong predictions. Typically, the generation process satisfies semantics-preserving and label-preserving assumptions. Mostly, semantics preservation can guarantee the same label for a perturbed text. Thus, a robust model should generate invariant outputs  $\arg \max_{y^*} f(x) =$

$\arg \max_{y^*} f(\tilde{x}) = y$  under semantics-preserving perturbations,

where  $y$  is the ground-truth label and  $\tilde{x}$  is the perturbed text. Adversarial attacks have been widely studied to automatically generate adversarial examples. We categorize the attack methods according to their perturbation types, perturbation space and adversary’s knowledge. Table 4 exemplifies typical methods for each category.

#### Perturbation Types

There are four common perturbation types that exhibit different granularities, namely, character editing, word substitution, paraphrasing and universal adversarial perturbation (UAP).

Character editing mimics real-life accidental typos or spelling variants in social media via character swapping (e.g., “place”  $\Rightarrow$  “plcae”), deletion (e.g., “artist”  $\Rightarrow$  “arlist”), insertion (e.g., “computer”  $\Rightarrow$  “comnputer”) and substitution (e.g., “computer”  $\Rightarrow$  “computer”) [Gao *et al.*, 2018]. Humans are robust to spelling errors maintaining certain morphological or phonological characteristics, e.g., “computer”. Hence, a few edits on a word would not affect the human perception of the word but may lead to completely opposite predictions by the models. In contrast, word substitutions require semantics-preserving constraints on substitute words or rely on a vocabulary of synonyms for substitutions [Zang *et al.*, 2020]. Paraphrasing is seldom considered in adversarial attacks due to the difficulty of generating high-precision paraphrases.

The last type of adversaries follows the work to generate UAPs for images, which perturb any images of a particular class [Moosavi-Dezfooli *et al.*, 2017]. Behjati *et al.*, [2019] generate the UAPs for NLP models in the form of n-grams (i.e., non-sensical phrases), which lead to misclassification when appended to text of a class. Wallace *et al.*, [2019a];

Knowledge	Perturbation Space	Perturbation Types
White-box	Continuous	Word substitution [Gong <i>et al.</i> , 2018], UAP [Song <i>et al.</i> , 2021], Character editing [Liu <i>et al.</i> , 2022]
	Discrete	Word substitution [Liang <i>et al.</i> , 2018; Ebrahimi <i>et al.</i> , 2018], UAP [Wallace <i>et al.</i> , 2019a; Behjati <i>et al.</i> , 2019]
Grey-box	Discrete	Word substitution [Alzantot <i>et al.</i> , 2018; Zang <i>et al.</i> , 2020; Jin <i>et al.</i> , 2020; Li <i>et al.</i> , 2020], Character editing [Gao <i>et al.</i> , 2018]
Black-box	Continuous	Paraphrasing [Zhao <i>et al.</i> , 2018]
	(Only generation model)	Paraphrasing [Iyyer <i>et al.</i> , 2018; Ribeiro <i>et al.</i> , 2018; Qi <i>et al.</i> , 2021]

Table 4: Classification of Attack Methods Based on Adversaries’ Knowledge, Perturbation Space, and Perturbation Types. UAP refers to Universal Adversarial Perturbation (UAP). Several studies employ generation models to produce paraphrases without adding any noise for perturbations.

2020] even extend the idea of UAPs for MRC, language modeling and machine translation. These studies assume that inserting non-sensical phrases does not change the semantics of text and ground-truth labels. Song *et al.*, [2021] generate natural phrases (e.g., “natural energy efficiency”) as UAPs via Adversarially Regularized Auto Encoder (ARAE). However, it can only maintain the coherence of UAPs rather than the whole perturbed text.

**Perturbation Space**

A text consists of a sequence of tokens (e.g., words or subwords), denoted as  $\mathbf{x} = w_1, w_2, \dots, w_T$ , where  $T$  is the number of tokens in the vocabulary  $\mathcal{V} = \{w_1, w_2, \dots, w_K\}$  ( $K$  is the size of the vocabulary). Note that the vocabulary can differ from the ones of the victim models, especially under black-box attacks.

All types of perturbations, except for paraphrasing, can be perturbed in a discrete space. When perturbing via word substitutions or UAPs, the objective is to identify significant tokens in  $\mathbf{x}$  and substitute them with tokens from the vocabulary  $\mathcal{V}$  in order to generate incorrect model predictions. This is a combinatorial optimization problem, where the size of the search space is  $K^T$ . The search space is intractable and is typically addressed using heuristic-based approaches [Alzantot *et al.*, 2018; Zang *et al.*, 2020] or approximation methods [Ebrahimi *et al.*, 2018; Wallace *et al.*, 2019a]. Character editing also requires identifying significant tokens, but it involves a subsequent manipulation of characters within the identified tokens [Gao *et al.*, 2018]. Assuming the average token length is  $l$ , the search space becomes  $l^T$ , leading to combinatorial explosion of modifications.

There are two types of methods to perturb text in the continuous space: 1) perturbing tokens (characters/subwords/words) in the embedding space. Gong *et al.*, [2018] add continuous perturbations directly to the continuous representation  $e_w$  of token  $w$  via element-wise addition between the perturbation  $\eta$  and  $e_w$ , similar to pixel-wise addition for images [Goodfellow *et al.*, 2015]. To transform  $\eta + e_w$  back to the text, they search for the nearest token  $\tilde{w}$  in the embedding space where  $e_{\tilde{w}} + \eta \in \mathcal{V}$ . However, it probably generates a semantically variant token, because the closest and most legible token can be far away from the original one in projection space. Note that they still require searching

for important tokens. 2) reparameterization for paraphrasing or UAPs. Zhao *et al.*, [2018] reparameterize the text  $\mathbf{x}$  into  $\mathbf{z}$  in a continuous space and then perturb  $\mathbf{z}$  into  $\tilde{\mathbf{z}}$ . They also train a text generator (e.g., LSTM) to decode  $\tilde{\mathbf{z}}$  back to text. Song *et al.*, [2020] apply the reparameterization trick for UAP generation.

Moreover, generation models are commonly used to create paraphrases such as machine translation models for back-translation [Ribeiro *et al.*, 2018], a syntactically controlled paraphrase network [Iyyer *et al.*, 2018] or a text style transfer model [Qi *et al.*, 2021]. The latter two specifically change task-irrelevant features.

**Adversary’s Knowledge**

We can categorize the attack methods into three types (white-box, grey-box and black-box) according to the three levels of model knowledge: model parameters, output logits (i.e., estimated probability distribution) and final predictions.

The white-box methods require model parameters to propagate gradients back to the input (e.g., the gradient of an adversarial loss with respect to the input word embedding  $\nabla_{e_w} \mathcal{L}$ ). They need  $\nabla_{e_w} \mathcal{L}$  to find important tokens and generate perturbations, e.g., the noise in the embedding space or substitute tokens. Liang *et al.*, [2018] adapt the gradient magnitude to find important words for perturbations and search for substitute words for each targeted label since the magnitude in each dimension of the word embedding indicates the sensitiveness of the prediction to the change. Gong *et al.*, [2018] directly use gradients to perform perturbations on the embedding space. Ebrahimi *et al.*, [2018] and Wallace *et al.*, [2019a; 2020] use the gradients and word embeddings to approximate the loss for word substitutions. Specifically, they approximate the loss change of substituting a word  $w$  with another word  $s$  in the vocabulary by the inner product of the word embedding  $e_s$  and  $\nabla_{e_w} \mathcal{L}$ , where  $\mathcal{L}$  is the adversarial loss. The selected word  $s$  is expected to minimize the adversarial loss on the perturbed text  $\tilde{\mathbf{x}}$ .

$$\begin{aligned} \arg \min_s \mathcal{L}(\tilde{\mathbf{x}}, y) \\ \approx e_s^T \nabla_{e_w} \mathcal{L}(\mathbf{x}, y) \end{aligned} \tag{2}$$

Under the grey-box setting, an adversary only has access to the probability distribution (or logits) over the output vari-

able  $Pr(y|\mathbf{x}) = f(\mathbf{x})$ , where  $Pr(y_i|\mathbf{x})$  is the probability of the outcome  $y_i$  given the text  $\mathbf{x}$ . Grey-box methods employ the probabilistic information to measure how important each token  $w_i \in \mathbf{x}$  is to the prediction. For  $w_i$ , its importance score  $I_i$  is commonly calculated as the difference of the probabilities of the correct class  $Pr(y_{\text{true}}|\mathbf{x}) - Pr(y_{\text{true}}|\mathbf{x}_{\setminus w_i})$  [Li *et al.*, 2020], where  $\mathbf{x}_{\setminus w_i}$  can either be  $\{w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_T\}$  (delete  $w_i$ ) [Li *et al.*, 2019; Jin *et al.*, 2020] or  $\{w_1, \dots, UNK, \dots, w_T\}$  (replace  $w_i$  with the unknown token *UNK*) [Gao *et al.*, 2018; Li *et al.*, 2020]. Jin *et al.*, [2020] also add the change to the misclassified class  $f_{\tilde{y}}(\tilde{x}) - f_{\tilde{y}}(x)$  if  $\arg \max_{y^*} f(\tilde{x}) \neq \arg \max_{y^*} f(x)$ , where  $\tilde{y}$  is

the misclassified class, and  $f_{\tilde{y}}(\tilde{x})$  is the probability of the misclassified class for the perturbed sample. Afterward, grey-box methods apply heuristics to search for substitute words or evaluate noisy words from character editing via  $Pr(y_{\text{true}}|\tilde{\mathbf{x}})$ .

The black-box attacks require no information about the victim model and commonly generate paraphrases. We have discussed the concrete methods in the previous paragraphs for perturbation types and space.

### 3.4 Evaluation Protocols

**Data-based evaluation.** There are different evaluation protocols for the three data types. NDS data and debiased data are evaluated on the same aggregate metrics for ID data (e.g., accuracy for text classification and F1 scores for MRC). The drops of these metrics from ID data to OOD data reveal the model’s ability of OOD generalization.

**Method-based evaluation.** It is meaningless to apply the ID metrics to adversarial examples unless used for adversarial transferability, since adversarial robustness only concerns the misclassified examples for the victim model. Instead, adversarial robustness is evaluated by the effectiveness of an attack method generating adversarial examples. The common evaluation metrics are *attack success rate* and *the number of queries*. Specifically, given a test sample  $\mathcal{D}_{\text{test}}$ , *attack success rate* measures the proportion of the data points in  $\mathcal{D}_{\text{test}}$  which can be perturbed to fool a victim model. It is calculated by dividing the number of adversarial examples that successfully achieve attack goals, e.g., misclassification, by the size of  $\mathcal{D}_{\text{test}}$ . *The number of model queries* during an attack measures how many perturbed examples have been evaluated before a valid adversarial example appears. Besides the query for predictions, white-box and grey-box methods also query victim models to get gradients or output logits. These two metrics can be used to compare the adversarial robustness between models (with fixed attack methods) [Gokhale *et al.*, 2022] and the performance of different attack methods [Li *et al.*, 2020] (with fixed victim models).

**Adversarial transferability.** Adversarial transferability is another evaluation type for adversarial robustness but uses the data-based evaluation method. It generates adversarial examples from a base model and then uses them to evaluate other models. The vulnerability can transfer across different model architectures and datasets [Wallace *et al.*, 2020]. Liang *et al.*, [2021] show that transferability can even occur across tasks as long as the source data distribution and the target one are

close. Adversarial transferability encourages the generation of costly human-made adversarial examples since they can be helpful in evaluating and improving the robustness of various models. For example, Nie *et al.*, [2020] and Wallace *et al.*, [2019b] show model predictions or interpretations (e.g., highlighting words that are important for model predictions) to crowdworkers to assist in the generation of adversarial examples.

## 4 Opportunities and Challenges

This section provides insights for future work based on the above OOD discussions. The opportunities and challenges related to adversarial robustness are also investigated.

### 4.1 General Challenges and Opportunities

Our study provides insights into OOD evaluation, OOD generalization and OOD detection.

**A benchmark for all the OOD types.** Developing a comprehensive benchmark for all the OOD types can help evaluate the model’s actual ability of language understanding and identify potential incapacity. Although there are open-source tools developed for OOD evaluation, most of them can only evaluate one perspective of the distribution shift. It is challenging to develop such a benchmark due to the idiomatic lock-in in each line of research, different data-generating processes and evaluation protocols. The disclosure of their connections and differences in this survey can provide insights into developing such a comprehensive benchmark for OOD evaluation. For example, adversarial robustness can be evaluated via adversarial transferability to unify evaluation metrics.

**Data-generating processes may break the assumptions of covariate shift.** Although all the three research lines aim to evaluate data with covariate shifts, some processes may break the assumptions of covariate shifts, i.e., fixed  $\mathbb{P}_0(\mathcal{Y}|\mathcal{X})$  and  $\mathbb{P}_0(\mathcal{Y})$ . These two assumptions also indicate  $\mathbb{P}_0(\mathcal{X}|\mathcal{Y})$  as a special case of covariate shift due to  $\mathbb{P}_0(\mathcal{X}|\mathcal{Y}) = \frac{\mathbb{P}_0(\mathcal{Y}|\mathcal{X})\mathbb{P}_0(\mathcal{X})}{\mathbb{P}_0(\mathcal{Y})}$ . For example, although  $\mathbb{P}_0(\mathcal{Y}|\mathcal{X})$  is always assumed the same for most NLP tasks since humans usually annotate each sentence with the same label, gradient-based methods may generate adversarial examples with labels mismatching with humans’ perceptions. In addition, some methods for generating debiased data may change  $\mathbb{P}_0(\mathcal{Y})$ . For example, Zhang *et al.*, [2019] balance examples with high lexical overlap for all kinds of labels in their debiased dataset to prevent the feature of lexical overlap from being a predictive indicator for paraphrase identification. Future work that evaluates covariate shifts (i.e., the changes of input domains) may better consider the two assumptions for data-generating processes and analyze their effects.

**Enhance general OOD performance.** The three types of out-of-distribution (OOD) data can be created by altering task-irrelevant features, as explained in Section 2. Our first argument suggests that OOD generalization techniques should aim to improve the acquisition of general linguistic or semantic knowledge to perform well on all three types of

OOD evaluation. This accounts for the observation that unsupervised pretraining and data augmentation strategies for learning general linguistic or semantic knowledge can enhance generalization on different domains and adversarial robustness [Gururangan *et al.*, 2020; Gokhale *et al.*, 2022]. Notably, such augmentation techniques [Wei and Zou, 2019] may solely modify the background features of training samples without any adversarial process. Our second argument is that even though dataset filtering for bias mitigation may improve model performance on biased datasets, it may lead to inferior performance on NDS data and decreased adversarial robustness, as biased data still contains knowledge of semantic and background features that can enhance general OOD performance. Supporting evidence for this argument is provided by Bras *et al.*, [2020] and Gokhale *et al.*, [2022] across various NLP tasks.

**Utilizing detection methods across different research lines.** The connections between OOD types regarding the shifted features also motivate future researchers to apply OOD detection methods across the three research lines. For example, since density estimation methods can effectively detect NDS data with background shift [Arora *et al.*, 2021], they may defend against black-box attacks, which always generate adversarial examples by shifting background features.

## 4.2 Challenges and Opportunities for Adversarial Robustness

**Beyond semantics preservation.** Adversarial examples can be generated beyond the assumption of semantics preservation. Chen *et al.*, [2022] define obstinate adversarial examples, which satisfy two conditions: 1) their ground-truth labels after the perturbation are changed, but 2) the victim model maintains the original predictions. Similarly, Kaushik *et al.*, [2020] minimally perturb test examples to change the labels of test examples. There are many reasons to explore obstinate adversarial examples. Gardner *et al.*, [2020] argue that such examples characterize the correct decision boundary for the task. Chen *et al.*, [2022] find that models with adversarial training on these examples reveal vulnerability to obstinate adversarial examples. Also, breaking the assumption of semantics preservation allows more adversarial behaviours for tasks beyond single-sentence classification. For example, Song *et al.*, [2020] generate nonsensical or natural sentences, leading to invariant model outputs for sentence-pair classification. Wallace *et al.*, [2020] change malicious content into nonsensical text which still makes machine translation models translate it into bad language.

**Developing realistically harmful adversarial behaviours.** There are different model behaviours designed for malicious intents during adversarial attacks. For classification tasks, we can either specify the expected prediction (targeted attack) or accept any prediction different from the correct label (non-targeted attack). The harm of attack depends on the specific task or labels. For example, it is harmful to generate adversarial examples to evade the detection of misinformation or toxic content. In contrast, there are many possibilities of adversarial behaviours for generation tasks due to various combinations of input and output text. Cheng *et al.*, [2020] define an

attack where all the words in the output text are different from the original output sequence while the input is similar. This kind of attack can lower the standard evaluation metrics based on n-grams, like BLEU scores. Wallace *et al.*, [2020] develop UAPs to make MT models hardly generate any translation or output random words. Targeted keyword attacks [Cheng *et al.*, 2020] can make models generate targeted words. Future researchers can define more realistically harmful behaviours, such as the generation of malicious nonsense or racial materials.

**Adversarial robustness may not be a good proxy for realistic scenarios.** Adversarial perturbations may make text deviate from their real-world distribution and generate rarely occurred examples. Particularly, pure gradient-based methods only pursue worst-case perturbations without considering the naturalness of adversarial examples. Hence, to alleviate this problem, adversarial candidates are commonly validated by some metrics such as language model perplexity [Alzantot *et al.*, 2018], and part-of-speech matching [Ebrahimi *et al.*, 2018]. However, attack processes become computationally expensive by including these separate modules and rejecting most of the perturbed examples. Another type of approach is to use limited search space for adversarial perturbations, e.g., using synonyms for word substitutions with the compromise of worst-case performance [Zang *et al.*, 2020]. It is worth exploring the attack methods that can inherently ensure the naturalness of perturbed text. There are some scenarios, as summarized below, where adversarial robustness against rare and unnatural text is undoubtedly critical. 1) When models' behaviours can cause devastating outcomes, e.g., astronautics or legal services. 2) When models are deployed in ubiquitous, unforeseeable scenarios. For example, Facebook translation once made the mistake of translating the simple phrase "good morning" in Arabic into "attack it" in English. Such a mysterious translation convinced the police that the user posting this message might launch a vehicle attack, resulting in the arrest of the user. 3) When attackers have sufficient malicious intent to explore rare cases, such as fact verification and security tasks where attackers would like to evade the model's detection to get unqualified access, e.g., propagating anti-social content or fake news on social media.

## 5 Conclusion

In this paper, we characterized and summarized the three mainstream lines of research on dataset biases, domain generalization and adversarial robustness. We encouraged future researchers to think comprehensively about OOD evaluation and improve the OOD generalization of NLP models on all the three types of OOD data. We also highlighted the gap between adversarial robustness and realistic OOD evaluation.

## References

[Alzantot *et al.*, 2018] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *EMNLP*, pages 2890–2896, 2018.

- [Arora *et al.*, 2021] Udit Arora, William Huang, and He He. Types of out-of-distribution texts and how to detect them. In *EMNLP*, pages 10687–10701, 2021.
- [Barrett *et al.*, 2019] Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. Adversarial removal of demographic attributes revisited. In *EMNLP-IJCNLP*, pages 6330–6335, 2019.
- [Behjati *et al.*, 2019] Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. Universal adversarial attacks on text classifiers. In *ICASSP*, pages 7345–7349, 2019.
- [Borkan *et al.*, 2019] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *World Wide Web Conference*, page 491–500, 2019.
- [Bras *et al.*, 2020] Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *ICML*, pages 1078–1088, 2020.
- [Chen *et al.*, 2022] Hannah Chen, Yangfeng Ji, and David Evans. Balanced adversarial training: Balancing trade-offs between fickleness and obstinacy in NLP models. In *EMNLP*, 2022.
- [Cheng *et al.*, 2020] Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *AAAI*, pages 3601–3608, 2020.
- [Ebrahimi *et al.*, 2018] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *ACL*, pages 31–36, 2018.
- [Emelin *et al.*, 2020] Denis Emelin, Ivan Titov, and Rico Sennrich. Detecting word sense disambiguation biases in machine translation for model-agnostic adversarial attacks. In *EMNLP*, pages 7635–7653, 2020.
- [Gao *et al.*, 2018] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56, 2018.
- [Gardner *et al.*, 2020] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models’ local decision boundaries via contrast sets. In *EMNLP*, pages 1307–1323, 2020.
- [Geva *et al.*, 2019] Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *EMNLP-IJCNLP*, pages 1161–1166, 2019.
- [Gokhale *et al.*, 2022] Tejas Gokhale, Swaroop Mishra, Man Luo, Bhavdeep Sachdeva, and Chitta Baral. *Generalized but not Robust?* comparing the effects of data modification methods on out-of-domain generalization and adversarial robustness. In *ACL*, pages 2705–2718, 2022.
- [Gong *et al.*, 2018] Zhitao Gong, Wenlu Wang, Bo Li, Dawn Song, and Wei-Shinn Ku. Adversarial texts with gradient methods. *arXiv preprint arXiv:1801.07175*, 2018.
- [Goodfellow *et al.*, 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.
- [Gururangan *et al.*, 2018] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *NAACL-HLT*, pages 107–112, 2018.
- [Gururangan *et al.*, 2020] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t stop pretraining: Adapt language models to domains and tasks. In *ACL*, pages 8342–8360, July 2020.
- [Hendrycks *et al.*, 2020] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. In *ACL*, pages 2744–2751, 2020.
- [Ilyas *et al.*, 2019] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in NIPS*, 2019.
- [Iyyer *et al.*, 2018] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL-HLT*, pages 1875–1885, 2018.
- [Jia and Liang, 2017] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*, pages 2021–2031, 2017.
- [Jin *et al.*, 2020] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI*, pages 8018–8025, 2020.
- [Kaushik *et al.*, 2020] Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*, 2020.
- [Kedzie *et al.*, 2018] Chris Kedzie, Kathleen McKeown, and Hal Daumé III. Content selection in deep learning models of summarization. In *EMNLP*, pages 1818–1828, 2018.
- [Lai *et al.*, 2021] Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. Why machine reading comprehension models learn shortcuts? In *ACL-IJCNLP*, pages 989–1002, 2021.
- [Li *et al.*, 2019] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. In *NDSS*, 2019.



- [Li *et al.*, 2020] Linyang Li, Ruotian Ma, Qipeng Guo, Xi-angyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In *EMNLP*, pages 6193–6202, 2020.
- [Liang *et al.*, 2018] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *IJCAI*, pages 4208–4215, 7 2018.
- [Liang *et al.*, 2021] Kaizhao Liang, Jacky Y Zhang, Boxin Wang, Zhuolin Yang, Sanmi Koyejo, and Bo Li. Uncovering the connections between adversarial transferability and knowledge transferability. In *ICML*, pages 6577–6587, 2021.
- [Liu *et al.*, 2022] Aiwei Liu, Honghai Yu, Xuming Hu, Shu’ang Li, Li Lin, Fukun Ma, Yawen Yang, and Lijie Wen. Character-level white-box adversarial attacks against transformers via attachable subwords substitution. In *EMNLP*, pages 7664–7676, December 2022.
- [McCoy *et al.*, 2019] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*, pages 3428–3448, 2019.
- [Moosavi-Dezfooli *et al.*, 2017] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, pages 1765–1773, 2017.
- [Naik *et al.*, 2018] Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *CoLing*, pages 2340–2353, 2018.
- [Nie *et al.*, 2020] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *ACL*, pages 4885–4901, 2020.
- [Oprea and Magdy, 2020] Silviu Vlad Oprea and Walid Magdy. The effect of sociocultural variables on sarcasm communication online. In *ACM*, page 22, 2020.
- [Qi *et al.*, 2021] Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *EMNLP*, pages 4569–4580, 2021.
- [Ribeiro *et al.*, 2018] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In *ACL*, pages 856–865, 2018.
- [Sakaguchi *et al.*, 2020] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial Winograd schema challenge at scale. In *AAAI*, pages 8732–8740, 2020.
- [Schuster *et al.*, 2019] Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. In *EMNLP-IJCNLP*, pages 3419–3425, 2019.
- [Song *et al.*, 2020] Congzheng Song, Alexander Rush, and Vitaly Shmatikov. Adversarial semantic collisions. In *EMNLP*, pages 4198–4210, 2020.
- [Song *et al.*, 2021] Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. Universal adversarial attacks with natural triggers for text classification. In *NAACL-HLT*, pages 3724–3733, 2021.
- [Sugawara *et al.*, 2018] Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. What makes reading comprehension questions easier? In *EMNLP*, pages 4208–4219, 2018.
- [Wallace *et al.*, 2019a] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In *EMNLP-IJCNLP*, pages 2153–2162, 2019.
- [Wallace *et al.*, 2019b] Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *ACL*, 7:387–401, 2019.
- [Wallace *et al.*, 2020] Eric Wallace, Mitchell Stern, and Dawn Song. Imitation attacks and defenses for black-box machine translation systems. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *EMNLP*, pages 5531–5546, 2020.
- [Wei and Zou, 2019] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP-IJCNLP*, pages 6382–6388, 2019.
- [Yi *et al.*, 2021] Mingyang Yi, Lu Hou, Jiacheng Sun, Lifeng Shang, Xin Jiang, Qun Liu, and Zhiming Ma. Improved ood generalization via adversarial training and pretraining. In Marina Meila and Tong Zhang, editors, *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 11987–11997. PMLR, 18–24 Jul 2021.
- [Zang *et al.*, 2020] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *ACL*, pages 6066–6080, 2020.
- [Zellers *et al.*, 2018] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*, pages 93–104, 2018.
- [Zhang *et al.*, 2019] Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In *NAACL*, pages 1298–1308, 2019.
- [Zhao *et al.*, 2018] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *ICLR*, 2018.