Sibylvariant Transformations for Robust Text Classification

Fabrice Harel-Canada[1], Muhammad Ali Gulzar[2], Nanyun Peng[1], Miryung Kim[1]
[1]Computer Science Department, University of California, Los Angeles
[2]Computer Science Department, Virginia Tech
{fabricehc, violetpeng, miryung}@cs.ucla.edu, gulzar@cs.vt.edu

Abstract

The vast majority of text transformation techniques in NLP are inherently limited in their ability to expand input space coverage due to an implicit constraint to preserve the original class label. In this work, we propose the notion of *sibylvariance (SIB)* to describe the broader set of transforms that relax the label-preserving constraint, knowably vary the expected class, and lead to significantly more diverse input distributions. We offer a unified framework to organize all data transformations, including two types of SIB: (1) *Transmutations* convert one discrete kind into another, (2) *Mixture Mutations* blend two or more classes together. To explore the role of sibylvariance within NLP, we implemented 41 text transformations, including several novel techniques like `Concept2Sentence` and `SentMix`. Sibylvariance also enables a unique form of adaptive training that generates new input mixtures for the most confused class pairs, challenging the learner to differentiate with greater nuance. Our experiments on six benchmark datasets strongly support the efficacy of sibylvariance for generalization performance, defect detection, and adversarial robustness.

1 Introduction

Automatically generating new data is a critical component within modern machine learning pipelines. During training, data augmentation can expose models to a larger portion of potential input space, consistently leading to better generalization and performance (Simard et al., 1998; Krizhevsky et al., 2012; Perez and Wang, 2017). After training, creating effective test instances from existing data can expose specific model failure modes and provide actionable corrective feedback (Zhang et al., 2019; Ribeiro et al., 2020).

While many techniques can artificially expand labeled training sets or test suites, nearly all of them are class-preserving. That is to say, the model outputs are invariant (INV) with respect to the transformations. This cautious constraint ensures the new data does not lie in an out-of-distribution null class which might impede the learning objective. However, it also requires more conservative transforms that inherently limit the degree of diversification.

In this work, we propose and extensively investigate the potential of *sibylvariant (SIB)* transformations that *knowably* vary the expected class. From the Greek *sibyls*, or oracles, the term parallels the oracle construction problem in software testing (Barr et al., 2015). In a nutshell, sibylvariants either fully transmute a datum from one class c_i to another c_j, or mix data from multiple classes together to derive a new input with a soft label that reflects the mixed membership. In this way, SIB can more strongly perturb and diversify the underlying distribution. Moreover, SIB makes possible a new type of adaptive training by synthesizing data from frequently confused class pairs, challenging the model to differentiate with greater refinement.

In the following sections, we position SIB within a broader conceptual framework for all data transforms (Section 2) and highlight several newly proposed techniques (Section 3). To support a comprehensive evaluation of how SIB may complement or even surpass its INV counterparts, we implemented 41 new and existing techniques into an open source tool called `Sibyl`. Equipped with the framework and tool, we evaluate 3 central research questions:

- **RQ1. Generalization Performance.** Does training on SIB-augmented data improve model accuracy on the original test set?
- **RQ2. Defect Detection.** For models trained on the original dataset, how effective are SIB tests at inducing misclassifications?
- **RQ3. Adversarial Robustness.** Are models trained on SIB-augmented data more robust to existing adversarial attack algorithms?

Findings of the Association for Computational Linguistics: ACL 2022, pages 1771 - 1788

Our comprehensive evaluation encompasses 6 text classification datasets, 11 transformation pipelines, and 3 different levels of data availability. In total, we trained 216 models and generated over 30 million new training inputs, 480,000 testing inputs, and 3,300 adversarial inputs. In the generalization study, SIB attained the highest accuracies in 89% (16 out of 18) of experimental configurations, with the adaptive mixture mutations being the most consistently effective. SIB also revealed the greatest number of model defects in 83% (5 out of 6) of the testing configurations. Lastly, of all the experimental configurations where adversarial robustness was improved over the no-transform baseline, 92% (11 out of 12) of them involved SIB. Overall, our findings strongly support the efficacy of sibylvariance for generalization performance, defect detection, and adversarial robustness.

Lastly, we describe how SIB may operate theoretically and discuss potential threats to validity (Section 5) before contrasting it with related work (Section 6). The source code for `Sibyl` and our experiments is available at: https://github.com/UCLA-SEAL/Sibyl.

2 Sibylvariance

All data transformations in the classification setting can be categorized into one of two types:

- ***Invariant (INV)*** preserves existing labels.

$$\{T_{INV}(X_i), y_i\} \rightarrow \{X_j, y_i\} \quad \text{where } X_i \neq X_j \tag{1}$$

  For example, contracting "What is the matter?" to "What's the matter?"should preserve a model behavior for sentiment analysis.

- ***Sibylvariant (SIB)*** changes an existing label in a knowable manner.

$$T_{SIB}(\{X_i, y_i\}) \rightarrow \{X_j, y_j\} \quad \text{where } X_i \neq X_j \text{ and } y_i \neq y_j. \tag{2}$$

  SIB transforms both the input X_i to X_j and the output label from y_i to y_j label, corresponding to the new X_j; such transformation is analogous to mutating an input and setting a corresponding oracle in metamorphic testing (Chen et al., 2020b). For example, performing a verb-targeted antonym substitution on "I love pizza." to generate "I hate pizza." has the effect of negating the original semantics and will knowably affect the outcome of binary sentiment analysis

It is important to note that transformation functions are not inherently INV nor SIB. The same exact transformation may have a different effect on expected model behavior depending on the particular classification task. For example, random word insertions generally have an INV effect on topic classification tasks, but would be SIB with respect to grammaticality tasks (Warstadt et al., 2018).

2.1 Sibylvariant Subtypes

SIB can be further refined based on the types and degree of semantic shift in newly generated data:

- ***Transmutation*** changes one discrete kind into another, excluding the existing label, $L \backslash \{y_i\}$,

$$T_{SIB}(\{X_i, y_i\}) \rightarrow \{X_j, y_j\} \quad \text{where } X_i \neq X_j \text{ and } y_j \in L \backslash \{y_i\}. \tag{3}$$

  Critically, the newly created data points retain stylistic and structural elements of the original that help boost diversity.

- ***Mixture Mutation*** mixes inputs from multiple classes and interpolates the expected behavior into a mixed label distribution (i.e. *soft label*). Equivalently, we have:

$$T_{SIB}(\{X_i, y_i\}) \rightarrow \{X_j, y_j\} \quad \text{where } X_i \neq X_j \text{ and } y_j \in \bigcap_{l}^{|L|} \lambda_l \tag{4}$$

  where the final term indicates a λ-degree of membership in each label l belonging to the expected input space and is normalized as a probability distribution (i.e. $\sum_l \lambda_l = 1$). For example, a document with topic 'surfing' can be combined with another document with topic 'machine learning' to yield a new label with probability mass placed on both topics. While mixture mutations may seem unnatural, the intuition is that humans can recognize mixed examples and adjust their predictions accordingly. Models ought to do the same.

2.2 Adaptive Sibylvariant Training

One unique and promising aspect of SIB is to target specific class pairings dynamically during training. In much the same way that a human teacher might periodically assess a students' understanding and alter their lesson plan accordingly, `Sybil` computes a confusion matrix and constructs more examples containing classes for which the model has the most difficulty differentiating. For example,

if a topic model most frequently misclassifies 'science' articles as 'business,' *adaptive* SIB (denoted as αSIB) will generate new blended examples of those classes in every mini-batch until the next evaluation cycle. At that point, if the model confuses 'science' for "health," αSIB will construct new mixtures of those classes and so on. `Sybil` supports built-in runtime monitoring for αSIB training.

3 Transformations

In `Sybil`, we defined 18 new transforms and adapt 23 existing techniques from prior work (Ribeiro et al., 2020; Morris et al., 2020; Wei and Zou, 2019) to expand the coverage of SIB and INV text transformations. At a high level, Table 1 shows these 41 transforms organized into 8 categories: `Mixture` (i.e., blending text), `Generative` (i.e. concept-based text generation), `Swap` (e.g., substituting antonyms, synonyms, hypernyms, etc.), `Negation` (e.g., adding or removing negation), `Punctuation` (e.g., adding or removing punctuation), `Text Insert` (e.g., adding negative, neutral, or positive phrases), `Typos` (e.g. adding various typos), and `Emojis` (e.g. adding or removing positive or negative emoji). We highlight several signature transforms here and provide a more detailed listing in Appendix A.

Category	Transformations
Mixture	TextMix†, SentMix†, WordMix†
Generative	Concept2Sentence†, ConceptMix†
Swap	ChangeNumber, ChangeSynonym, ChangeAntonym, ChangeHyponym, ChangeHypernym, ChangeLocation, ChangeName, RandomSwap
Negation	AddNegation, RemoveNegation
Punctuation	ExpandContractions, ContractContractions
Text Insert	RandomInsertion, AddPositiveLink†, AddNegativeLink†, ImportLinkText†, InsertPositivePhrase, InsertNegativePhrase
Typos	RandomCharDel, RandomCharInsert, RandomCharSubst, RandomCharSwap, RandomSwapQwerty, WordDeletion, HomoglyphSwap
Emojis	Emojify†, AddEmoji†, AddPositiveEmoji†, AddNegativeEmoji†, AddNeutralEmoji†, Demojify†, RemoveEmoji†, RemovePositiveEmoji†, RemoveNegativeEmoji†, RemoveNeutralEmoji†

Table 1: Transformations currently available in `Sybil`. New transforms that we defined are marked with †.

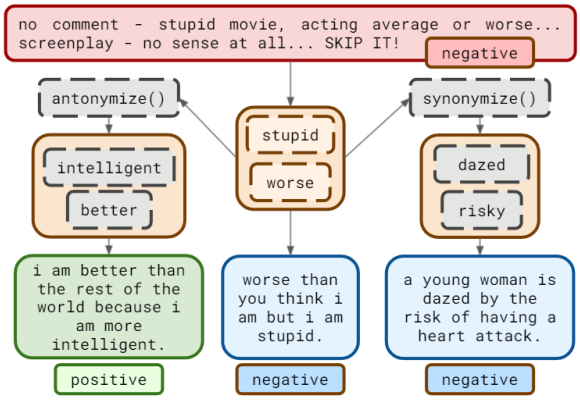


Figure 1: C2S intakes a text and its label (red) to extract keywords, ['stupid, worse']. These words are used to generate a new INV sentence shown in red. Alternatively, antonym (left) and synonym (right) substitution can produce new concepts that further boost diversity.

Concept2Sentence (C2S). C2s is a two step process: (1) extract a short list of key concepts from a document and (2) generate a new sentence that retains critical semantic content of the original while varying its surface form, style, and even subject matter. To accomplish this, we leveraged integrated gradients (Sundararajan et al., 2017; Pierse, 2021) to produce saliency attributions that identify the most relevant tokens for a given class label. We then generate a well-composed sentence from the extracted concepts using a pre-trained BART (Lewis et al., 2019) model fine-tuned on the CommonGen dataset (Lin et al., 2019).

Prior to generation, it is possible to apply other transformations to the extracted concepts to encourage diversity or knowably alter the label. For example, on the left hand side of Figure 1 an antonym substitution produces a SIB effect by changing the extracted concepts from ['stupid', 'worse'] to ['intelligent', 'better']. The new sentence exhibits a change in subject and style, but is correctly transmuted to have positive sentiment. C2S is thus an extremely promising transformation for diversifying text along both INV and SIB directions.

TextMix, SentMix, and WordMix. Mixture mutations, like mixup (Zhang et al., 2017) and cutmix (Yun et al., 2019) from the image domain, take a batch of inputs and blend them together to form new inputs with an interpolated loss and they have shown robustness to adversarial attacks. `TextMix` translates this idea to the text domain by merging two inputs and interpolating a *soft label* according to the proportion of tokens belonging to the constituent classes. While `TextMix` does

a straightforward concatenation, `SentMix` shuffles the sentences and thus encourages long-range comprehension. `WordMix` concatenates and shuffles all words, encouraging keyword-to-topic understanding when sentence structure is compromised.

4 Experiments

4.1 Transformation Pipelines & Datasets

To compare the potential of INV, SIB, and both (INVSIB) in aggregate, we construct a *transformation pipeline* (TP) (Cubuk et al., 2019; Xie et al., 2019), where we uniformly sample n transformations of the selected kind to generate new $\{X_i, y_i\}$ pairs. We also create TPs that apply a single transform, $T_{\texttt{SINGLE}}$, to highlight the efficacy of `C2S`, `TextMix`, `SentMix`, `WordMix` and their adaptive versions, prefixed with α. In total, we evaluate 11 TPs per dataset, shown in Table 2.

Due to space limitations, we report the top performing TP of each kind using an asterisk (*). INV* represents the best from T_{INV} and T_{C2S}, while SIB* represents the best from T_{SIB} and the mixture mutations. For **RQ1**, we also compare against TMix (Chen et al., 2020a), EDA (Wei and Zou, 2019), and AEDA (Karimi et al., 2021). TMix is a recent *hidden*-space mixture mutation for text, as opposed to `Sybil`'s direct mixture mutation on the input space with greater transparency and examinability. EDA and AEDA are examples of recent INV transformations. Full results are available in the appendices.

Shorthand	Description
T_{ORIG}	0 transformations as baseline
T_{INV}	sample 2 INVs
T_{SIB}	sample 2 SIBs
T_{INVSIB}	sample 1 INV and 1 SIB
T_{SINGLE}	apply one from C2S, TextMix, SentMix, WordMix, αTextMix, αSentMix, αWordMix

Table 2: TP descriptions. TPs with an α-prefix use targeted, adaptive training (Section 2.2).

We study six benchmarks for two kinds of NLP tasks: topic classification and sentiment analysis. Table 3 summarizes their relevant details. To simulate different levels of resource availability, we create three data subsets with by varying number of examples per class — 10, 200, and 2500. These subsets were expanded 30× via augmentation for each TP. In total, we generated 144 new datasets (144 = 6 benchmarks * 3 levels of data availability * 8 TPs which persist data. αSIB is runtime only.)

4.2 Model Setting

We used a `bert-base-uncased` model (Devlin et al., 2018) with average pooling of encoder output, followed by a dropout layer (Srivastava et al., 2014) with probability 0.1, and a single linear layer with hidden size 768 and GELU (Hendrycks and Gimpel, 2016) activation. Maximum sentence length was set to 250. We use a batch size 16, an Adam optimizer (Kingma and Ba, 2014) with a linear warmup, a 0.1 weight decay, and compute accuracy every 2,000 steps. All models were trained for 30 epochs on eight Nvidia RTX A6000 GPUs, with early stopping. In total, we constructed 198 different models.

For all TPs that produce a soft-label, we use a multi-class cross-entropy loss and computed performance via a weighted top-k accuracy,

$$\sum_{j}^{k} \lambda_l \cdot \mathbb{1}(y_l = \hat{y}_j), \tag{5}$$

where λ_j is the degree of class membership, $\mathbb{1}(\cdot)$ is the indicator function, and y_j and $\hat{y}_j$ are the indices of the j-th largest predicted score for the ground truth label and predicted label, respectively.

4.3 RQ1. Generalization Performance

For RQ1, we explore how model accuracy on the original test set is influenced by training data augmented with INV and SIB transformations. Table 4 shows the results on six benchmarks with three levels of data availability.

We observe the most significant performance gains when training 10 examples per class —accuracy is improved by 4.7% on average across all datasets and by a maximum of up to 15% for IMDB. Figure 2 shows that as the number of labeled training data increases, a dominant trend emerged —T_{SIB} always generalized better to unseen test data. In fact, the only kind of transformation to always outperform both T_{ORG} and TMix is SIB*. Figure 3 shows the performance delta between INV* and SIB* against the T_{ORG} baseline at 200 examples per class. For every dataset, either αSentMix or αTextMix is the best performing TP, while INV* actually leads to performance decreases for DBPedia, Yahoo! Answers, and IMDB.

One key reason that aided SIB in attaining strong performance is the use of adaptive training. On average, crafting new examples that target the

Dataset	Source	Task	Subject	Classes	Test	Avg Len
AG News	(Zhang et al., 2015)	Topic	News Articles	4	1,900	38
DBpedia	(Zhang et al., 2015)	Topic	Wikipedia Articles	14	5,000	46
Yahoo! Answers	(Zhang et al., 2015)	Topic	QA Posts	10	6,000	92
Amazon Polarity	(Zhang et al., 2015)	Sentiment	Product Reviews	2	200,000	74
Yelp Polarity	(Zhang et al., 2015)	Sentiment	Business Reviews	2	10,000	133
IMDB	(Maas et al., 2011)	Sentiment	Movies Reviews	2	12,500	234

Table 3: Dataset details. Test represents the number of examples per class in the test set.

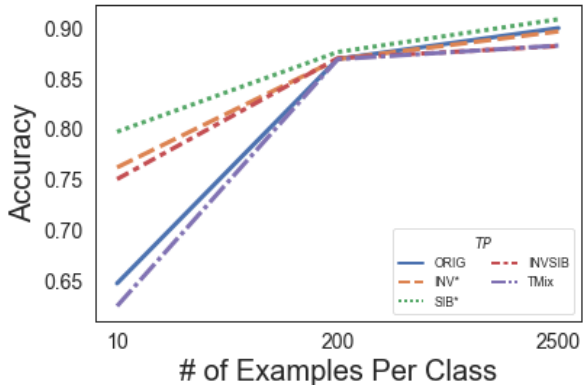


Figure 2: SIB* outperforms INV* most, when data availability is low, indicating the necessity of SIB to complement INV.

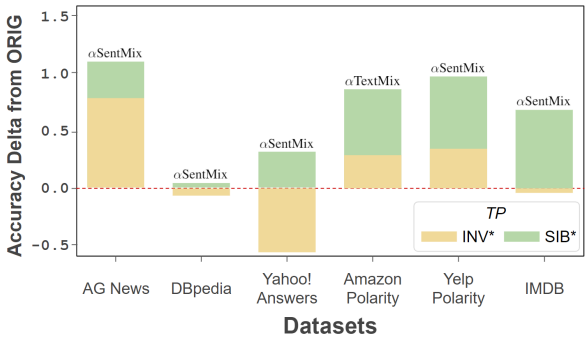


Figure 3: The best performing TP for each dataset trained on 200 examples per class. αSentMix or αTextMix leads to the highest performance gains. SIB* consistently outperforms INV*.

model's primary confusions during training added approximately 1% to accuracy relative to mixing classes uniformly at random. This shows another unique benefit of sibylvariance that is not transferable to its INV counterparts.

While our full scale experiments show a clear trend that SIB generally outperforms INV, we primarily evaluated TPs combining multiple transforms instead of assessing the efficacy of each in isolation. Initially, this was a logistical decision due to computational limitations. To investigate each transformation's effect individually, we conducted a small scale experiment training 756 models ((39 transformations + 3 αSIB) $\times$ 6 datasets $\times$ 3 runs) on 10 examples per class with a 3$\times$ augmentation multiplier. Based on this experiment, we then computed each transform's performance by averaging the accuracy change relative to a T_{ORIG} baseline across all datasets. Table 5 shows the top ten best performing transforms, six of which employ SIB techniques. These results expand support for the overall conclusion that sibylvariance represents an especially effective class of transformations for improving generalization performance.

Generalization Performance.
Models trained upon SIB-augmented data attained the highest test set accuracy in 89% (16 out of 18) of experimental configurations, with the adaptive mixture mutations being the most consistently effective.

4.4 RQ2. Defect Detection

For RQ2, we assess how generating new tests with INV and SIB can expose defective model behavior. A single test is simply an $\{X_i, y_i\}$ pair and a test suite is a set of such tests. Defective behavior is misclassification, which is measured via a test suite's accuracy. For each dataset D, we select a high-performing BERT model trained only on the original dataset without any augmentation. Then for each of eight TPs (excluding αSIB relevant to training only), we create 100 test suites, each containing 100 randomly sampled tests. This yields a total of 480,000 tests. We then report an average accuracy for each D and TP pair.

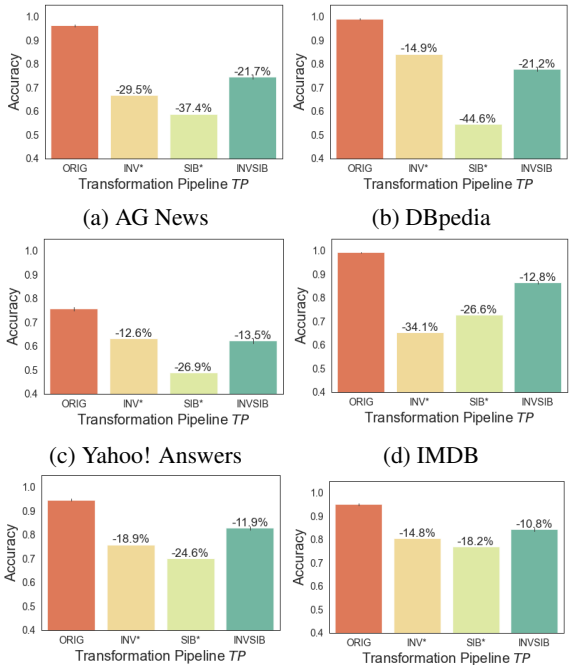
Figure 4 shows how defect detection is enabled by INV and SIB. With the exception of Yahoo! Answers, the models scored nearly perfect accuracy on T_{ORIG}; however, when the same models are tested using data generated with INV and SIB, they struggle to generalize. Test data synthesized with SIB can reveal most defects in these models, indicating the value of *sibylvariance* in constructing test oracles for ML models in the absence of

Dataset	TP	# Examples / Class			Dataset	TP	# Examples / Class			Dataset	TP	# Examples / Class		
		10	200	2500			10	200	2500			10	200	2500
AG News	ORIG	75.08	88.70	91.65	**DBpedia**	ORIG	95.71	98.87	98.96	**Yahoo! Answers**	ORIG	56.24	69.77	73.18
	INV*	**84.28**	89.46	91.95		INV*	97.29	98.81	99.00		INV*	61.39	69.21	72.53
	SIB*	83.52	**89.80**	**92.42**		SIB*	**97.96**	**98.90**	**99.06**		SIB*	**62.47**	**70.10**	**73.37**
	INVSIB	84.09	89.00	91.36		INVSIB	95.64	98.74	98.92		INVSIB	62.01	67.75	73.16
	TMix ‡	81.38	88.62	89.43		TMix ‡	97.51	98.66	98.89		TMix ‡	53.68	69.03	69.50
	EDA ‡	81.50	88.98	90.93		EDA ‡	97.42	98.63	98.89		EDA ‡	57.88	68.03	69.15
	AEDA ‡	81.03	88.74	92.09		AEDA ‡	97.30	98.88	98.89		AEDA ‡	59.51	67.37	69.91
Amazon Polarity	ORIG	67.30	89.22	92.08	**Yelp Polarity**	ORIG	74.62	91.66	93.70	**IMDB**	ORIG	64.70	86.96	90.02
	INV*	73.69	89.53	92.21		INV*	**83.91**	92.00	94.29		INV*	76.20	86.94	89.69
	SIB*	**74.90**	**90.03**	**92.26**		SIB*	80.46	**92.60**	**94.69**		SIB*	**79.74**	**87.65**	**90.90**
	INVSIB	73.50	89.06	91.26		INVSIB	78.90	91.85	93.03		INVSIB	75.04	87.04	88.24
	TMix ‡	62.14	87.98	91.00		TMix ‡	61.81	91.19	92.80		TMix ‡	62.45	86.94	88.29
	EDA ‡	59.40	87.68	92.20		EDA ‡	71.90	90.88	94.11		EDA ‡	67.37	86.45	89.07
	AEDA ‡	64.72	88.92	91.83		AEDA ‡	79.39	91.60	94.06		AEDA ‡	72.61	86.56	88.63

Table 4: RQ1 accuracy comparison for INV*, SIB*, and INVSIB against baselines ORIG, TMix (Chen et al., 2020a), EDA (Wei and Zou, 2019), AEDA (Karimi et al., 2021). An asterisk (*) indicates the best performance observed across underlying TPs of each kind, while a ‡ indicates related works for comparison.

Transform	Type	Avg Δ (%)
αSentMix	SIB	+4.26
αTextMix	SIB	+3.55
RandomCharInsert	INV	+3.55
TextMix	SIB	+3.22
Concept2Sentence	INV	+2.70
AddPositiveLink	INV / SIB	+2.48
AddNegativeEmoji	INV / SIB	+2.45
SentMix	SIB	+2.33
ExpandContractions	INV	+2.15
RandomCharSubst	INV	+2.06

Table 5: Top ten individual transforms over a no-transform baseline averaged across all datasets. The INV / SIB types were SIB for the sentiment analysis datasets and INV for the topic classification datasets. See Table 11 in the Appendix for more details.



(a) AG News (b) DBpedia (c) Yahoo! Answers (d) IMDB (e) Amazon Polarity (f) Yelp Polarity

Figure 4: RQ2 defect detection comparison. Percentages show change in accuracy relative to $T_{\texttt{ORIG}}$. Lower accuracy indicates greater efficacy at inducing error.

expensive human labeling and judgements.

Tests which lie outside the expected input distribution are not likely to be fair nor actionable. Since SIB transforms generally perturb data more aggressively than INV ones, they likewise possess more potential for creating unreasonable, out-of-domain tests of model quality. However, the positive results in **RQ1** may justify the use of SIB transformations as reasonable for testing. Had the newly transformed data truly belonged to a different distribution, model performance on the in-domain test set should have decreased as a result of dataset shift (Quiñonero-Candela et al., 2009; Hu et al., 2022). In fact, we observed the opposite as model performance was consistently improved. This suggests that SIB transforms yield data that is tenably in-domain and therefore may complement INV transforms in exposing defective model behavior.

We theorize that the effectiveness of SIB-generated tests comes from the expanded objectives it permits. For example, T_{TextMix} assess whether the model can recognize which classes are present and to what degree. T_{SentMix} does the same but further scrutinizes long-range comprehension by broadly distributing related topic sentences. Datasets with lengthy inputs are particularly vulnerable to transformations of this kind. Lastly, T_{WordMix} forces the model to forgo reliance on text structure and evaluates keyword comprehension amidst noisy contexts. In contrast, most INV transformations involve minor changes — e.g. expand contractions — and test the aspect of language already well modeled from extensive pre-training. The INV C2S transform is an exception that drastically alters input and thus reveals more defects than other T_{INV} pipelines.

Defect Detection. Models tested with SIB-transformed data exhibited the greatest number of defects in 83% (5 out of 6) of experimental configurations.

4.5 RQ3. Adversarial Robustness

For RQ3, we assess whether models trained on INV or SIB are more resilient to adversarial attacks than models trained on an original data. An adversarial text input is typically obtained via semantic preserving (i.e. *invariant*) perturbations to legitimate examples in order to deteriorate the model performance. The changes are typically generated by ascending the gradient of the loss surface with respect to the original example and improving robustness to adversarial attacks is a necessary precondition for real-world NLP deployment.

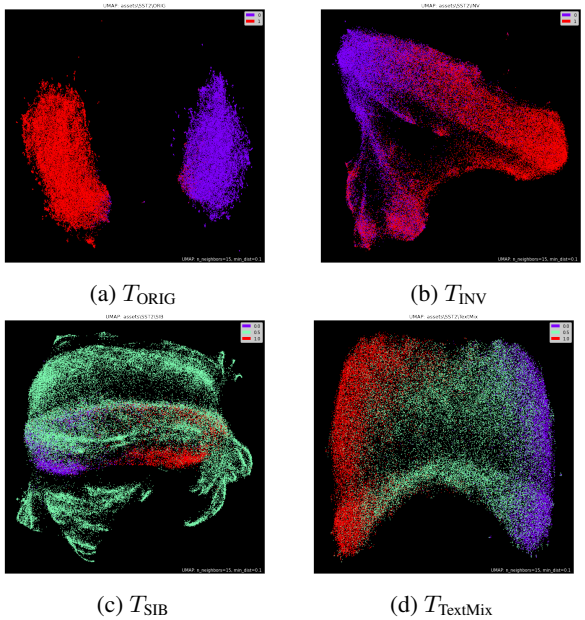
We select three attack algorithms based on their popularity and effectiveness: (1) TextFooler (Jin et al., 2019), (2) DeepWordBug (Gao et al., 2018), and (3) TextBugger (Li et al., 2018), all as implemented in TextAttack (Morris et al., 2020). We focus on models trained with 10 examples per class because the largest changes in generalization performance are more likely to exhibit the clearest trend for adversarial robustness. For each of 11 models and 3 attacks, we randomly sample 100 inputs from the original data and perturb them to create a total of 3,300 adversarial examples.

Table 6 shows that, of all the cases where adversarial robustness is improved over $T_{\texttt{ORIG}}$, 92% of them involve SIB. On average, SIB*-trained models improve robustness by 4%, while INV*-trained models sustain a 1% decrease. Topic classification is made more robust via training with augmented data. Consistently, $T_{\alpha\texttt{-SentMix}}$ produces the most resilient models. For sentiment analysis, improved generalization performance enabled by SIB does not necessarily lead to improved robustness to existing adversarial attacks. The underlying sentiment models trained with augmented data improves generalization over $T_{\texttt{ORIG}}$ by an average of 5%. However, counter-intuitively, the models are not more robust to the three attacks than $T_{\texttt{ORIG}}$ and that Pearson correlation is -0.28 between accuracy and adversarial robustness. This finding motivates future work to investigate why there is a negative correlation and how to design SIB such that accuracy improvement also translates to corresponding adversarial robustness.

Adversarial Robustness. Of all the experimental configurations where adversarial robustness was improved over the no-transform baseline, 92% (11 out of 12) of them involved models trained on SIB-augmented data.

5 Discussion

How does sibylvariance help? The primary purpose of data transformations in ML is to *diversify* datasets in the neighborhood of existing points, a principle formalized as Vicinal Risk Minimization (VRM) (Chapelle et al., 2001). Synthetic examples can be drawn from a vicinal distribution to find similar but different points that enlarge the original data distribution. For instance, within image classification, it is common to define the vicinity of an image as the set of its random crops, axal reflections, and other label-preserving INV transforms. While VRM can expose ML models to more diverse input space and consequently reduce generalization errors, the neighborhoods created by INV are relatively restricted. This is due to the label-preserving constraint limiting the degree of perturbation freedom on the original data.



(a) T_{ORIG} (b) T_{INV}

(c) T_{SIB} (d) T_{TextMix}

Figure 5: UMAP visualizations of BERT `[CLS]` tokens for SST-2. Blue, red, and green represent "Negative," "Positive," and "Mixed", respectively.

SIB effectively expands the vicinity relation via transmutations and mixture mutations. Newly created data can claim full or mixed membership in target classes. To support our intuition, we vi-

Dataset	TP	Attack Success Rate			Dataset	TP	Attack Success Rate			Dataset	TP	Attack Success Rate		
		TF	DWB	TB			TF	DWB	TB			TF	DWB	TB
AG News	ORIG	0.69	0.56	0.54	**DBpedia**	ORIG	0.92	0.55	0.64	**Yahoo! Answers**	ORIG	0.54	0.46	0.52
	INV*	0.66	0.56	0.48		INV*	**0.76**	0.47	0.48		INV*	0.57	0.49	0.49
	SIB*	**0.60**	**0.43**	**0.45**		SIB*	0.77	**0.40**	**0.41**		SIB*	**0.48**	**0.41**	0.49
	INVSIB	0.78	0.62	0.57		INVSIB	0.83	0.56	0.52		INVSIB	0.54	0.44	**0.46**
Amazon Polarity	ORIG	**0.48**	0.40	0.42	**Yelp Polarity**	ORIG	**0.48**	**0.20**	**0.28**	**IMDB**	ORIG	0.86	**0.25**	0.71
	INV*	0.49	0.42	**0.36**		INV	0.64	0.41	0.52		INV*	0.70	0.50	0.68
	SIB*	0.55	**0.39**	0.46		SIB	0.61	0.39	0.53		SIB*	**0.56**	0.32	**0.55**
	INVSIB	0.65	0.58	0.60		INVSIB	0.75	0.51	0.61		INVSIB	0.89	0.79	0.88

Table 6: RQ3 adversarial robustness comparison for INV*, SIB*, and INVSIB using TextFooler (TF), DeepWordBug (DWB), and TextBugger (TB). A lower attack success rate indicates a higher adversarial robustness.

sualize the effects of various transformations on SST-2 (Socher et al., 2013). Figure 5 presents the UMAP-reduced (McInnes et al., 2020) `[CLS]` tokens produced by a BERT transformer for sentiment classification. Figure 5a shows that the classes are initially well separated and high performance can be obtained by selecting any separating surface between the two clusters. However, a more reasonable choice for the best boundary is one that exhibits the largest margin between classes — the very intuition behind Support Vector Machines (Cortes and Vapnik, 1995). Figure 5d suggests that a model trained on mixture mutations is likely to arrive at a boundary with the lowest loss. For example, in 5d, the augmented examples in green provide additional loss feedback from uncovered portions of the input space to encourage a decision boundary that maximizes the margin between class clusters. A similar expectation may hold for SIB in Figure 5c. However, the effects of INV transforms shown in Figure 5b do not appear to support such margin maximization.

Threats to Validity. External threats to validity include the generalization of our results to model architectures dissimilar to BERT (i.e. `bert-base-uncased`). It is possible that larger autoencoder models like RoBERTa (Liu et al., 2019) and auto-regressive models like XLNet (Yang et al., 2019) may respond differently to SIB transformations. Secondly, while the framework of sibylvariance is applicable to all data types, we have only provided empirical results supporting their efficacy for text classification models. We leave the exploration of SIB applications to image, time series, and other domains to future work.

Internal threats include how we derived mixed labels for generated text. We assumed that the critical semantics can be approximated via the ratio of words contributed by source text. This assumption may not account for other linguistic interaction and thus could lead to suboptimal labels. However, SIB did significantly improve upon the INV and the ORIG baselines in the **RQ1** generalization study, suggesting that the constructed soft labels still reflected useful semantics. This indirectly supports the validity of SIB-transformed data for testing in **RQ2**, although we acknowledge that additional caution is required for using any aggressively modified, synthetic data as a substitute for real data for the purpose of exposing defective model behavior.

6 Related Work

In this section, we broadly cover data transformations within and outside of the text domain because our proposed framework for sibylvariance is applicable to all classification contexts.

Data Augmentation. Effective data augmentation is a key factor enabling superior model performance on a wide range of tasks (Krizhevsky et al., 2012; Jiang et al., 2018; Xie et al., 2019). In many cases, practitioners leverage domain knowledge to reinforce critical invariances in the underlying data. In computer vision, for example, translation invariance is the idea that no matter where the objects of interest reside within an image, the model will still classify them correctly. Image translations and random crops encourage this more generalized conceptualization within the model (Simard et al., 1998) and all other transforms have a similar goal: reinforce a particular invariance that helps the learner perform well on future unseen data.

Numerous techniques have been proposed to assist with this learning objective and thereby improve generalization. Random erasing (Zhong et al., 2017; Devries and Taylor, 2017) and noise injection (Wen et al., 2020; Xie et al., 2019) support invariance to occlusions and promote robust features. Interpolating (Bowyer et al., 2011) and extrapolating (DeVries and Taylor, 2017) nearest neighbors in the input / feature space reinforces a linear relationship between the newly created data and the supervision signal while reducing class imbalance. However, nearly all of these approaches, and many others (Shorten and Khoshgoftaar, 2019;

Feng et al., 2021), are label-preserving and therefore limited in their capacity to induce deeper learning of invariant concepts.

Sibylvariant transforms enjoy several desirable aspects of INV transformations while mitigating their drawbacks. Similar to feature space functions (DeVries and Taylor, 2017), mixture mutations do not require significant domain knowledge. Like approaches that reduce dataset imbalance (Bowyer et al., 2011), SIB transforms can increase class representation through mixed membership or targeted transmutations that inherit diverse characteristics of the source inputs. In all cases, relaxing the label-preserving constraint enables SIB functions to both complement and enhance the learning of critical invariances by further expanding the support of the dataset in new directions.

Adversarial Attacks & Robustness. Adversarial attacks are a special class of INV transformations that simultaneously minimize perturbations to the input while maximizing the perception of change to a learner. This task is more difficult within the NLP domain due to the discrete nature of text, but several works (Alzantot et al., 2018; Zhang et al., 2020) have proven successful at inducing model errors. Real-world use of NLP requires resilience to such attacks and our work complements robust training (Parvez et al., 2018) and robust certification (Ye et al., 2020; Pruksachatkun et al., 2021) to produce more reliable models.

Emerging Sibylvariant Transforms. Specific transformations designed to alter the expected class of an input have existed prior to this work (Zhang et al., 2017; Yun et al., 2019; Guo, 2020; Zhu et al., 2017), albeit primarily in the image domain and also in a more isolated, ad hoc fashion. Among our primary contributions is to propose a unifying name, framework, and taxonomy for this family of sibylvariant functions. Furthermore, most prior works introduce a single transformation and evaluate its efficacy on training alone. In contrast, we proposed several novel transformations, a new adaptive training routine, and evaluated the broader impacts of 41 INV and SIB transforms on training, defect detection, and robustness simultaneously.

Recently published examples of SIB mixture mutations for text (Guo et al., 2019; Chen et al., 2020a) differ from ours in several important ways. Prior work operates exclusively within the *hidden* space inside specific models, which limits transferability between different algorithm types. All of our transformations operate in the *input* space, which is both more general and more challenging because we have to contend with rules of grammar and style. However, this also provides greater transparency. Furthermore, because our overall approach samples from 41 different transformations, we are able to exercise a broader range of model behaviors. For example, `SentMix` is designed to encourage long-range understanding, while other transforms evoke their own specific objectives. Any individual transformation is inherently more limited, e.g. TMix can only encourage the model to behave linearly for borderline cases.

7 Conclusion

Inspired by metamorphic testing, we proposed the notion of *sibylvariance* to jointly transform both input and output class (X_i, y_i) pairs in a knowable way. To explore the potential of sibylvariance, we define 18 new text transformations and adapt 23 existing transformations into an open source tool called `Sybil`. In particular, we define several types of mixture mutations and design a novel concept-based text transformation technique utilizing salience attribution and neural sentence generation. Across six benchmarks from two different NLP classification tasks, we systematically assess the effectiveness of INV and SIB for generalization performance, defect detection, and adversarial robustness. Our extensive evaluation shows that many SIB transforms, and especially the *adaptive* mixture mutations, are extremely effective. SIB achieves the highest training accuracy in 89% of the experimental configurations. When used for testing, SIB test suites reveal the greatest number of model defects in 5 out of 6 benchmarks. Finally, models trained on SIB-augmented data improve adversarial robustness $11\times$ more often than those trained on INV-augmented data.

Acknowledgements

This work is supported in part by National Science Foundations via grants CCF-2106420, CCF-2106404, CNS-2106838, CCF-1764077, CHS-1956322, CCF-1723773, ONR grant N00014-18-1-2037, Intel CAPA grant, Samsung, and a CISCO research contract. We would also like to thank Atharv Sakhala for early contributions to the `Sybil` project as well as Jason Teoh, Sidi Lu, Aaron Hatrick, Sean Gildersleeve, Hannah Pierce, and all the anonymous reviewers for their many helpful suggestions.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *CoRR*, abs/1804.07998.

E. Barr, M. Harman, P. McMinn, M. Shahbaz, and Shin Yoo. 2015. The oracle problem in software testing: A survey. *IEEE Transactions on Software Engineering*, 41:507–525.

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. 2011. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. 2001. Vicinal risk minimization. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020a. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *CoRR*, abs/2004.12239.

T. Chen, S. Cheung, and S. Yiu. 2020b. Metamorphic testing: A new approach for generating next test cases. *ArXiv*, abs/2002.12543.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2019. Randaugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Terrance DeVries and Graham W. Taylor. 2017. Dataset augmentation in feature space.

Terrance Devries and Graham W. Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *CoRR*, abs/1801.04354.

Hongyu Guo. 2020. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *AAAI*.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *CoRR*, abs/1905.08941.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Qiang Hu, Yuejun Guo, Maxime Cordy, Xiaofei Xie, Lei Ma, Mike Papadakis, and Yves Le Traon. 2022. An empirical study on data distribution-aware test selection for deep learning enhancement.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0.1: the winning entry to the VQA challenge 2018. *CoRR*, abs/1807.09956.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: an easier data augmentation technique for text classification. *CoRR*, abs/2108.13230.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *CoRR*, abs/1812.05271.

Bill Yuchen Lin, Ming Shen, Yu Xing, Pei Zhou, and Xiang Ren. 2019. Commongen: A constrained text generation dataset towards generative commonsense reasoning. *CoRR*, abs/1911.03705.

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

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Md. Rizwan Parvez, Tolga Bolukbasi, Kai-Wei Chang, and Venkatesh Sarigrama. 2018. Building a robust text classifier on a test-time budget. *CoRR*, abs/1808.08270.

Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621.

Charles Pierse. 2021. Transformers Interpret.

Yada Pruksachatkun, Satyapriya Krishna, Jwala Dhamala, Rahul Gupta, and Kai-Wei Chang. 2021. Does robustness improve fairness? approaching fairness with word substitution robustness methods for text classification. *CoRR*, abs/2106.10826.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. 2009. When training and test sets are different: Characterizing learning transfer.

Marco Túlio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *ACL*.

Connor Shorten and T. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48.

Patrice Simard, Yann LeCun, John S. Denker, and Bernard Victorri. 1998. Transformation invariance in pattern recognition-tangent distance and tangent propagation. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, page 239–27, Berlin, Heidelberg. Springer-Verlag.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *ArXiv*, abs/1703.01365.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196.

Qingsong Wen, Liang Sun, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2020. Time series data augmentation for deep learning: A survey. *CoRR*, abs/2002.12478.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised data augmentation. *CoRR*, abs/1904.12848.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. *CoRR*, abs/2005.14424.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. *CoRR*, abs/1905.04899.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412.

Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. 2019. Machine learning testing: Survey, landscapes and horizons. *CoRR*, abs/1906.10742.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3).

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2017. Random erasing data augmentation. *CoRR*, abs/1708.04896.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593.

A Implemented `Sybil` Transformations

Category	Transformation	Sentiment	Topic
Mixture	TextMix	SIB	SIB
Mixture	SentMix	SIB	SIB
Mixture	WordMix	SIB	SIB
Generative	Concept2Sentence	INV	INV
Generative	ConceptMix	SIB	SIB
Word Swap	replace antonym	SIB	INV
Word Swap	replace cohyponym	INV	INV
Word Swap	replace hypernym	INV	INV
Word Swap	replace hyponym	INV	INV
Word Swap	replace synonym (wordnet)	INV	INV
Word Swap	change numbers (except 2 and 4)	INV*	INV
Word Swap	change locations based on dictionary	INV	INV
Word Swap	change names based on dictionary	INV	INV
Negation	add negation	INV*	INV
Negation	remove negation	INV*	INV
Punctuation	expand contractions	INV	INV
Punctuation	reduce contractions	INV	INV
Text Insertion	add URL to negative content	SIB	INV
Text Insertion	add URL to positive content	SIB	INV
Text Insertion	add negative phrase	SIB	INV
Text Insertion	add positive phrase	SIB	INV
Typos	char deletion	INV*	INV
Typos	char insertion	INV*	INV
Typos	char movement (n spaces)	INV*	INV
Typos	char repacement (homoglyph)	INV	INV
Typos	char replacement	INV*	INV
Typos	char swap (n spaces)	INV*	INV
Typos	char swap (QWERTY)	INV*	INV
Typos	word deletion	INV*	INV
Typos	word insertion	INV*	INV
Typos	word replacement	INV*	INV
Typos	word replacement (homophone)	INV	INV
Typos	word swap	INV*	INV
Emojis	replace words with emojis (Emojify)	INV	INV
Emojis	replace emojis with words (Demojify)	INV	INV
Emojis	add negative emoji	SIB	INV
Emojis	add neutral emoji	INV	INV
Emojis	add positive emoji	SIB	INV
Emojis	remove negative emoji	SIB	INV
Emojis	remove neutral emoji	INV	INV
Emojis	remove positive emoji	SIB	INV

Table 7: Transform descriptions currently implemented in `Sybil`, sampled from according to task (sentiment analysis or topic) and TP (INV, SIB, or INVSIB). Note that transformations are INV or SIB with respect to specific tasks. Asterisks (*) indicate that the variance type could be either INV or SIB, but the listed variance was judged to be more likely.

B Other Possible Text Transformations

Category	Transformation	Sentiment	Topic
Word Swap	replace synonym (embedding)	INV	INV
Word Swap	word swap (masked)	INV*	INV
Word Swap	change gendered pronoun	INV	INV*
Word Swap	change protected class	INV	INV*
Word Swap	change "for" to 4	INV	INV
Word Swap	change "to" to 2	INV	INV
Word Swap	swap phrase with acronym	INV	INV
Negation	negation of negative clause	SIB	INV
Negation	negation of neutral clause	INV	INV
Negation	negation of positive clause	SIB	INV
Paraphrase	backtranslation	INV	INV
Punctuation	add exclamation	INV*	INV
Punctuation	add period	INV	INV
Punctuation	add question mark	INV	INV
Punctuation	remove exclamation	SIB*	INV
Punctuation	remove period	INV	INV
Punctuation	remove question mark	INV	INV
Text Insertion	add random URL (404)	INV	INV
Text Insertion	add neutral phrase	INV	INV
Tense / Voice	make continuous future tense	INV*	INV
Tense / Voice	make continuous past tense	INV*	INV
Tense / Voice	make continuous present tense	INV*	INV
Tense / Voice	make perfect continuous future tense	INV*	INV
Tense / Voice	make perfect continuous past tense	INV*	INV
Tense / Voice	make perfect continuous present tense	INV*	INV
Tense / Voice	make perfect future tense	INV*	INV
Tense / Voice	make perfect past tense	INV*	INV
Tense / Voice	make perfect present tense	INV*	INV
Tense / Voice	make simple future tense	INV*	INV
Tense / Voice	make simple past tense	INV*	INV
Tense / Voice	make simple present tense	INV*	INV
Tense / Voice	change voice active	INV	INV
Tense / Voice	change voice passive	INV	INV
Emojis	replace emoji with word antonym	SIB	INV
Emojis	replace emoji with word synonym	INV	INV
Emojis	replace word with emoji antonym	SIB	INV
Emojis	replace word with emoji synonym	INV	INV

Table 8: Transform NOT currently implemented in `Sybil`, but represent potentially interesting directions for future work. Asterisks (*) indicate that the variance type could be either INV or SIB, but the listed variance was judged to be more likely.

C Sibylvariant Subtype Examples

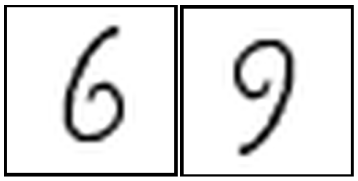
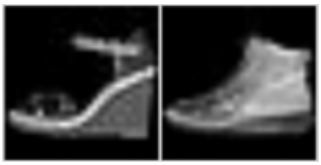
SIB Subtype	Image (Classification)	Text (Sentiment Analysis)
Transmutation $A \rightarrow B$ (Hard Label) Changes one class into another class, while retaining stylistic elements of the original.	***Rotation***  Digit 6 $\rightarrow$ Digit 9 ***GAN-based Object Transfiguration***  Sandal $\rightarrow$ Sneaker	***Antonym Replacement*** I love NY $\downarrow$ I hate NY ***Clause Negation*** You are a good person. $\downarrow$ You are not a good person. ***Stock Phrase Insertion*** It was a clever movie. $\downarrow$ It was a clever movie. That said, I absolutely hated it.
Mixture Mutation $A + B \rightarrow AB$ (Soft Label) Mixes two or more class labels into a single data point and then interpolates the expected behavior.	***Mixup*** (Zhang et al., 2017) ***Cutmix*** (Yun et al., 2019)  $[1, 0] + [0, 1] \rightarrow [0.35, 0.65]$ ***Tile***  $[1, 0, 0, 0] + [0, 0, 1, 0] +$ $[0, 1, 0, 0] + [0, 0, 0, 1] \rightarrow$ $[0.25, 0.25, 0.25, 0.25]$	***TextMix*** virutally unwatchable... + a vivid, thoughtful, unapologetically raw coming-of-age tale full of sex, drugs and rock 'n' roll. = virutally unwatchable... a vivid, thoughtful, unapologetically raw coming-of-age tale full of sex, drugs and rock 'n' roll. $[1, 0] + [0, 1] \rightarrow [0.17, 0.83]$ ***WordMix*** it is essentially empty + this is a visually stunning rumination on love = love visually is is essentially rumination on it stunning this a empty $[1, 0] + [0, 1] \rightarrow [0.33, 0.67]$

Table 9: Examples of SIB transformations for the image and text domains. For mixture mutations, we show a soft label proportional to the pixel and word counts of their constituent parts.

D RQ1. Detailed Training Results

Dataset	TP	10	200	2500	Dataset	TP	10	200	2500
AG News	ORIG	75.08	88.70	91.65	Amazon Polarity	ORIG	67.30	89.22	92.08
	INV	**84.28**	89.46	91.95		INV	71.09	89.53	92.21
	C2S	82.82	87.84	91.43		C2S	73.69	86.76	90.20
	SIB	83.52	89.20	91.55		SIB	69.23	87.00	91.45
	TextMix	83.53	89.17	91.58		TextMix	68.20	88.63	91.46
	SentMix	83.56	89.28	91.49		SentMix	71.22	88.85	91.28
	WordMix	82.61	88.59	90.42		WordMix	60.27	85.40	87.68
	αTextMix	81.53	89.51	92.20		αTextMix	**74.90**	**90.03**	**92.26**
	αSentMix	77.28	**89.80**	**92.42**		αSentMix	64.19	90.01	92.16
	αWordMix	83.13	89.46	91.91		αWordMix	64.21	89.09	91.98
	INVSIB	84.09	89.00	91.36		INVSIB	73.50	89.06	91.26
	TMix ‡	81.38	88.62	89.43		TMix ‡	62.14	87.98	91.00
	EDA ‡	81.50	88.98	90.93		EDA ‡	59.40	87.68	92.20
	AEDA ‡	81.03	88.74	92.09		AEDA ‡	64.72	88.92	91.83
DBpedia	ORIG	95.71	98.87	98.96	Yelp Polarity	ORIG	74.62	91.66	93.70
	INV	97.29	98.81	99.00		INV	77.92	92.00	94.29
	C2S	96.23	98.36	96.41		C2S	**83.91**	89.59	92.80
	SIB	95.26	98.73	97.60		SIB	78.67	91.89	93.69
	TextMix	**97.96**	98.88	97.86		TextMix	79.27	91.07	93.36
	SentMix	97.95	98.86	99.01		SentMix	80.46	91.96	93.62
	WordMix	97.03	97.89	98.59		WordMix	74.47	88.39	92.12
	αTextMix	97.72	98.87	99.04		αTextMix	77.72	91.73	94.50
	αSentMix	96.38	**98.90**	**99.06**		αSentMix	76.63	**92.60**	**94.69**
	αWordMix	97.01	**98.90**	98.90		αWordMix	78.30	91.50	93.67
	INVSIB	95.64	98.74	98.92		INVSIB	78.90	91.85	93.03
	TMix ‡	95.76	98.53	98.55		TMix ‡	61.81	91.19	92.80
	EDA ‡	97.42	98.63	98.89		EDA ‡	71.90	90.88	94.11
	AEDA ‡	97.30	98.88	98.89		AEDA ‡	79.39	91.60	94.06
Yahoo! Answers	ORIG	56.24	69.77	73.18	IMDB	ORIG	64.70	86.96	90.02
	INV	60.24	69.21	72.53		INV	76.20	86.94	89.69
	C2S	61.39	67.31	70.60		C2S	70.18	85.67	86.98
	SIB	61.30	68.45	73.18		SIB	73.51	86.38	88.71
	TextMix	**62.47**	68.72	72.08		TextMix	73.23	85.24	89.45
	SentMix	60.95	68.72	72.07		SentMix	76.75	85.55	89.10
	WordMix	59.98	67.66	72.96		WordMix	67.15	84.19	88.23
	αTextMix	60.26	69.89	73.15		αTextMix	74.09	87.52	90.60
	αSentMix	59.10	**70.10**	73.00		αSentMix	**79.74**	**87.65**	**90.90**
	αWordMix	60.74	69.99	**73.37**		αWordMix	73.01	86.92	87.85
	INVSIB	62.01	67.75	73.16		INVSIB	75.04	87.04	88.24
	TMix ‡	53.68	69.03	69.50		TMix ‡	62.45	86.94	88.29
	EDA ‡	57.88	68.03	69.15		EDA ‡	67.37	86.45	89.07
	AEDA ‡	59.51	67.37	69.91		AEDA ‡	72.61	86.56	88.63

Table 10: Performance (test set accuracy (%)) for all TPs. The results are averaged across three runs. Models are trained with either 10, 200, or 2500 examples per class. TPs are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively. White with a ‡ indicates related works for comparison. For TMix, EDA, and AEDA, we used the author's open source code with their default / recommended configurations to transform the training datasets. However, we maintained the same model training hyperparameters as our other TPs to facilitate fair comparisons with our work.

Transform	Type	Avg Δ (%)
αSentMix	SIB	+4.26
αTextMix	SIB	+3.55
RandomCharInsert	INV	+3.55
TextMix	SIB	+3.22
Concept2Sentence	INV	+2.70
AddPositiveLink	INV / SIB	+2.48
AddNegativeEmoji	INV / SIB	+2.45
SentMix	SIB	+2.33
ExpandContractions	INV	+2.15
RandomCharSubst	INV	+2.06
AddNeutralEmoji	INV	+1.90
RandomInsertion	INV	+1.72
AddNegativeLink	INV / SIB	+1.64
αWordMix	SIB	+1.62
ChangeNumber	INV	+1.44
AddPositiveEmoji	INV / SIB	+1.25
InsertNegativePhrase	INV / SIB	+1.15
RemoveNegation	INV	+1.00
WordDeletion	INV	+0.86
RandomSwapQwerty	INV	+0.83
RandomCharSwap	INV	+0.77
ContractContractions	INV	+0.69
Emojify	INV	+0.59
ChangeLocation	INV	+0.37
Demojify	INV	+0.34
AddNegation	INV	+0.13
WordMix	SIB	+0.08
ConceptMix	SIB	-0.11
RandomCharDel	INV	-0.16
RemovePositiveEmoji	INV	-0.24
RandomSwap	INV	-0.28
ImportLinkText	INV	-0.56
ChangeHyponym	INV	-0.63
RemoveNeutralEmoji	INV	-0.72
RemoveNegativeEmoji	INV / SIB	-0.80
ChangeName	INV	-0.84
InsertPositivePhrase	INV / SIB	-0.95
ChangeSynonym	INV	-1.26
ChangeHypernym	INV	-1.78
ChangeAntonym	INV / SIB	-2.82
HomoglyphSwap	INV	-3.78

Table 11: Performance (test set accuracy (%)) for individual transforms over a no-transform baseline averaged across all datasets. The INV / SIB types were SIB for the sentiment analysis datasets and INV for the topic classification datasets.

E RQ2. Detailed Defect Detection Results

Dataset	TP	Test Suite Accuracy	Dataset	TP	Test Suite Accuracy
AG News	ORIG	96.22	**Amazon Polarity**	ORIG	94.68
	INV	89.77		INV	86.91
	C2S	66.67		C2S	75.78
	SIB	74.77		SIB	80.99
	TextMix	59.97		TextMix	79.83
	SentMix	60.48		SentMix	79.83
	WordMix	**58.82**		WordMix	**70.08**
	INVSIB	74.50		INVSIB	82.78
DBpedia	ORIG	99.04	**Yelp Polarity**	ORIG	95.15
	INV	93.27		INV	89.76
	C2S	84.17		C2S	80.39
	SIB	71.67		SIB	82.76
	TextMix	**54.42**		TextMix	80.67
	SentMix	57.09		SentMix	81.09
	WordMix	57.48		WordMix	**76.91**
	INVSIB	77.79		INVSIB	84.32
Yahoo! Answers	ORIG	75.64	**IMDB**	ORIG	99.25
	INV	69.71		INV	90.01
	C2S	63.08		C2S	**65.15**
	SIB	58.87		SIB	84.48
	TextMix	**48.77**		TextMix	78.42
	SentMix	51.82		SentMix	79.45
	WordMix	53.58		WordMix	72.64
	INVSIB	62.17		INVSIB	86.42

Table 12: Test suite accuracy (%) by dataset and TP. Lower accuracy indicates higher defect detection potential. TPs are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively.

F RQ3. Detailed Robustness Results

Dataset	TP	TF	DWB	TB	Dataset	TP	TF	DWB	TB
AG News	ORIG	0.69	0.56	0.54	Amazon Polarity	ORIG	**0.48**	0.40	0.42
	INV	0.73	0.59	0.48		INV	0.49	0.42	**0.36**
	C2S	0.66	0.56	0.48		C2S	0.51	0.49	0.50
	SIB	0.80	0.69	0.49		SIB	0.68	0.55	0.63
	TextMix	0.78	0.60	**0.45**		TextMix	0.56	0.41	0.46
	SentMix	0.70	0.57	0.61		SentMix	0.58	0.47	0.46
	WordMix	0.84	0.71	0.60		WordMix	0.74	0.69	0.73
	αTextMix	0.77	0.60	0.57		αTextMix	0.55	**0.39**	0.48
	αSentMix	**0.60**	**0.43**	0.46		αSentMix	0.56	0.49	0.53
	αWordMix	0.79	0.64	0.55		αWordMix	0.74	0.69	0.69
	INVSIB	0.78	0.62	0.57		INVSIB	0.65	0.58	0.60
DBpedia	ORIG	0.92	0.55	0.64	Yelp Polarity	ORIG	**0.48**	**0.20**	**0.28**
	INV	**0.76**	0.47	0.48		INV	0.64	0.41	0.52
	C2S	0.85	0.59	0.56		C2S	0.76	0.58	0.66
	SIB	0.80	0.58	0.64		SIB	0.68	0.53	0.65
	TextMix	0.85	0.48	**0.41**		TextMix	0.76	0.61	0.67
	SentMix	0.96	0.69	0.69		SentMix	0.70	0.52	0.60
	WordMix	0.91	0.64	0.76		WordMix	0.78	0.72	0.76
	αTextMix	0.82	0.51	0.53		αTextMix	0.61	0.39	0.53
	αSentMix	0.87	**0.40**	0.51		αSentMix	0.94	0.77	0.87
	αWordMix	0.83	0.55	0.49		αWordMix	0.62	0.49	0.56
	INVSIB	0.83	0.56	0.52		INVSIB	0.75	0.51	0.61
Yahoo! Answers	ORIG	0.54	0.46	0.52	IMDB	ORIG	0.86	**0.25**	0.71
	INV	0.57	0.49	0.49		INV	0.70	0.50	0.68
	C2S	0.58	0.53	0.54		C2S	0.93	0.59	0.89
	SIB	0.56	0.50	0.53		SIB	0.71	0.47	0.71
	TextMix	0.58	0.47	0.50		TextMix	0.85	0.32	0.73
	SentMix	0.72	0.64	0.72		SentMix	0.80	0.46	0.78
	WordMix	0.65	0.52	0.63		WordMix	0.84	0.74	0.84
	αTextMix	0.54	0.47	0.49		αTextMix	**0.56**	0.32	**0.55**
	αSentMix	**0.48**	**0.41**	0.48		αSentMix	0.95	0.91	0.96
	αWordMix	0.66	0.59	0.61		αWordMix	0.73	0.52	0.68
	INVSIB	0.54	0.44	**0.46**		INVSIB	0.89	0.79	0.88

Table 13: Attack success by dataset and TP for three adversarial algorithms: TextFooler (TF), DeepWordBug (DWB), and TextBugger (TB). Lower attack success indicates higher adversarial robustness. TPs are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively.