Identifying Adversarial Attacks on Text Classifiers

Zhouhang Xie[*1] **Jonathan Brophy**[*2] **Adam Noack**[2] **Wencong You**[2]
Kalyani Asthana[1] **Carter Perkins**[2] **Sabrina Reis**[2] **Sameer Singh**[1] **Daniel Lowd**[2]
[1] University of California, Irvine CA
[2] University of Oregon, Eugene OR
zhx022@ucsd.edu {jbrophy, anoack2, wyou, carterp, lowd}@cs.uoregon.edu
sreis@uoregon.edu {kasthana, sameer}@uci.edu

Abstract

The landscape of adversarial attacks against text classifiers continues to grow, with new attacks developed every year and many of them available in standard toolkits, such as TextAttack and OpenAttack. In response, there is a growing body of work on robust learning, which reduces vulnerability to these attacks, though sometimes at a high cost in compute time or accuracy. In this paper, we take an alternate approach — we attempt to understand the attacker by analyzing adversarial text to determine which methods were used to create it. Our first contribution is an extensive dataset for attack detection and labeling: 1.5 million attack instances, generated by twelve adversarial attacks targeting three classifiers trained on six source datasets for sentiment analysis and abuse detection in English. As our second contribution, we use this dataset to develop and benchmark a number of classifiers for attack identification — determining if a given text has been adversarially manipulated and by which attack. As a third contribution, we demonstrate the effectiveness of three classes of features for these tasks: text properties, capturing content and presentation of text; language model properties, determining which tokens are more or less probable throughout the input; and target model properties, representing how the text classifier is influenced by the attack, including internal node activations. Overall, this represents a first step towards forensics for adversarial attacks against text classifiers.

1 Introduction

Text classifiers have been under attack ever since spammers started evading spam filters, nearly 20 years ago (Hulten et al., 2004). In recent years, however, attacking classifiers has become much easier to carry out. Many general-purpose attacks have been developed and are now available in standard, plug-and-play frameworks, such as TextAttack (Morris et al., 2020) and OpenAttack (Zeng et al., 2021). The wide use of standard architectures and shared pretrained representations have further increased the risk of attack by decreasing the diversity of text classifiers.

Attack	Text	Label
Original	the acting is amateurish	Negative
Pruthi	the acting is amateirish	Positive
DeepWordBug	the acting is aateurish	Positive
IGA	the acting is enthusiastic	Positive

Table 1: Attack Samples on SST-2

Our focus is on evasion attacks (Barreno et al., 2006), in which an attacker attempts to change a classifier's prediction by making minor, semantics-preserving perturbations to the original input. To accomplish this, different adversarial attack algorithms employ different types of perturbations, search methods, and constraints. See Table 1 for some brief examples of how different attacks make different word or character substitutions to change a classifier's prediction(more examples in §B.6).

A common defense strategy is to make classifiers more robust, using algorithms with heuristic or provable guarantees on their performance (Madry et al., 2018; Cohen et al., 2019). However, these defenses are often computationally expensive or result in reduced accuracy. Therefore, as a complement to making classifiers more robust, we introduce the task of *attack identification* — automatically determining the adversarial attacks (if any) used to generate a given piece of text. The idea behind attack identification is that many attackers will use whatever attacks are most convenient, such as public implementations of attack algorithms, instead of developing new ones or implementing ones on their own. Thus, we can identify specific attacks instead of detecting or preventing *all* possible attacks. Our primary focus is on attack *labeling*, determining which specific attack was used (or none).

[*]Equal contribution

This gives us information about how the attacks are being conducted, which can be used to develop defense strategies for the overall system, such as uncovering malicious actors behind misinformation or abuse campaign on social media.

To address the problem of attack identification, we introduce the **Text Classification Attack Benchmark (TCAB)**, an extensive dataset of attacks on English text classifiers, which can be used for training and evaluating attack-identification models. TCAB uses six domain datasets from sentiment analysis and abuse detection. For each domain dataset, we train three target classifiers for the adversary to attack; we choose classifiers with transformer architectures (Wolf et al., 2020) as these models achieve state-of-the-art results *and* previous work has shown that adversarial examples curated against transformers have the highest transferability to other architectures such as CNNs and LSTMs, while the reverse is not true (Li et al., 2021b). We then run twelve attacks from the TextAttack and OpenAttack toolkits against each classifier for all datasets, this amounts to a total of 216 domain dataset/target model/attack combinations. The final TCAB dataset consists of: (1) all attacks that successfully flipped the label of the target classifier, a total of 1,539,881 adversarial instances, and (2) the unperturbed "clean" instances from the original domain test sets. The data and code used to generate TCAB will be released publicly, so it can be expanded and updated with additional datasets and new attacks as they are developed.

To characterize the text properties that are useful in identifying attacks, we investigate three classes of features. *Text properties (T)* capture the content and presentation of the text, such as the number of non-ASCII characters and average word length. This can differentiate attacks that perform different transformations on the input, from replacing words with synonyms to inserting characters or punctuation. *Language model properties (L)* indicate how natural each token is in the context of the input, helping to identify attacks that, for example, generate very improbable tokens. *Target classifier properties (C)* represent how a text classifier responds to the input, including its internal node activations and gradients. We refer to these three sets of features collectively as *TLC features*. We combine these with a standard BERT (Devlin et al., 2019) representation of the input.

We evaluate linear and tree-ensemble classifiers on TCAB using this rich feature set. We find that attack detection (determining if any attack is present) can be done with 84–97% accuracy when the target model used to generate the attack matches the target model properties, and 83–91% accuracy when they differ. We also find that our models generalize to detecting attacks that were unseen at training time, which is promising for detecting new attacks in the wild. For labeling, accuracies range from 45–71% accuracy[1]. Together, the TCAB dataset, our TLC features, and the evaluation represent a substantial first step towards automated identification of adversarial attacks against text classifiers.

2 Attack Identification

Existing work on defending against adversarial textual attacks mainly focuses on building robust models via adversarial training (Miyato et al., 2017), and much less attention has been put on detecting the presence of an attack (Mozes et al., 2021) (for a detailed list of existing defenses, see §6). In computer vision, recent work has shown that attacks on image classifiers can be detected and the attack method responsible can be identified (Moayeri and Feizi, 2021); they argue that knowing which specific attack was used allows for more specific defenses. With an increasing number of published textual attack methods (Gao et al., 2018; Garg and Ramakrishnan, 2020; Zang et al., 2020) and attack frameworks (Morris et al., 2020; Zeng et al., 2021), being able to not only detect an attack but also *label* the attack method used to perturb the input may better help NLP practitioners understand the attack.

Strictly speaking, labeling is not necessary for defending classifiers — if we determine with some confidence that an attack is present, we can simply label a piece of text (or its source) as malicious. The benefit of labeling is to learn more about the attacker. For example, if we find that the same attack is being applied by many different accounts on a social network, then we may reasonably suspect that all are part of a coordinated bot attack. Alternatively, if we find that an attack is novel and does not match any we have seen before, then we might suspect the attacker to be more sophisticated and have more resources than an attacker using a standard implementation. Thus, the information

[1]Test sets are balanced, so baseline accuracy is 50% for binary detection and 8.33% on the multiclass labeling task.

gained from attack labeling could be used when developing defense strategies for the overall system.

Problem Setup In this work, we focus on text classifiers and attacks on them. Given an input sequence $\mathbf{x} = (x_1, x_2, ..., x_N) \in \mathcal{X}$ (the instance space), a text classifier f maps $\mathbf{x}$ to a label $y \in \mathcal{Y}$, the set of output labels. For sentiment analysis, $\mathcal{Y}$ may be positive or negative sentiment; or for toxic comment detection, $\mathcal{Y}$ may be toxic or non-toxic.

A text-classification adversary aims to generate an adversarial example $\mathbf{x}'$ such that $f(\mathbf{x}') \neq f(\mathbf{x})$. Ideally, the changes made on $\mathbf{x}$ to obtain $\mathbf{x}'$ are minimal such that a human would label them the same way. Perturbations may occur on the character-, token-, phrase-, or sentence-level, or a combination of levels; perturbations may also be structured such that certain input properties are preserved, such as the semantics, perplexity, fluency, or grammar.

Given a (possibly) perturbed input sequence $\mathbf{x}^* \in \mathcal{X}$, we aim to develop a detector $f^D : \mathcal{X} \rightarrow \{-1, +1\}$ that detects the presence of any perturbations on $\mathbf{x}^*$; we call this task *attack detection*. In addition, we aim to identify the method used to perturb the input. Given $\mathbf{x}^* \in M(\mathbf{x})$, in which $M(\mathbf{x})$ is a function that perturbs $\mathbf{x}$ using any one attack method from a set of attacks S (including a "clean" attack in which the input is not perturbed), we develop an attack labeler $f^L : \mathcal{X} \rightarrow S$; we introduce this more difficult task as *attack labeling*.

In pursuit of these aims, we develop and curate a large collection of adversarial attacks on a number of classifiers trained on various domain datasets. In the following sections, we describe our process for generating this benchmark, and then detail the features we use to build f^D and f^L. Finally, we evaluate the effectiveness of our models in detecting and labeling different attack methods.

3 Creating an Identification Benchmark

We now present the Text Classification Attack Benchmark (TCAB), a dataset for developing and evaluating methods for identifying adversarial attacks against text classifiers.

3.1 Tasks and Domain Datasets

For sentiment analysis, we attack models trained on three domains: (1) **Climate Change**[2], 62,356 tweets on climate change; (2) **IMDB** (Maas et al., 2011), 50,000 movie reviews, and (3) **SST-2** (Socher et al., 2013), 68,221 movie reviews. For abuse detection, we attack models trained on three toxic-comment datasets: (1) **Wikipedia** (Talk Pages) (Wulczyn et al., 2017; Dixon et al., 2018), which contains 159,686 comments from Wikipedia administration webpages, (2) **Hatebase** (Davidson et al., 2017), which contains 24,783 comments, and (3) **Civil Comments**[3], which contains 1,804,874 comments from independent news sites. All datasets are binary (positive vs. negative or abusive vs. non-abusive) except for Climate Change, which includes neutral sentiment.

	BERT		RoBERTa		XLNet	
Dataset	Acc.	AUC	Acc.	AUC	Acc.	AUC
Climate Change*	79.8	0.899	**81.2**	**0.917**	80.1	0.910
IMDB	87.0	0.949	**90.7**	**0.968**	90.1	0.965
SST-2	91.8	0.972	**92.7**	**0.978**	92.3	0.974
Wikipedia	96.5	0.982	**96.6**	**0.985**	96.4	0.983
Hatebase	**95.8**	0.983	**95.8**	**0.987**	93.9	0.979
Civil Comments	**95.2**	**0.968**	95.1	0.967	95.0	0.965

Table 2: Predictive performance of the target models on test set for each domain dataset; *: multiclass-macro-averaged AUC; the rest are binary-classification tasks.

To create TCAB, we perturb examples from the test sets of these six domain datasets. SST-2, Wikipedia (Talk Pages), and IMDB have predefined train/test splits. For the other three datasets, we use an 80/10/10 split for training, validation, and testing. For each model/domain dataset combination, we only attack test set examples in which the model's prediction is correct. For the abuse datasets, we further constrain our focus to examples in the test set that are both predicted correctly *and* toxic; perturbing non-toxic text to be classified as toxic is a less likely adversarial task.

3.2 Target Models

We finetune BERT, RoBERTa, and XLNet models — all from HuggingFace's transformers library (Wolf et al., 2020) — on the six domain datasets. Table 2 shows the performance of these models on the test set of each domain dataset. On most datasets, RoBERTa slightly outperforms the other two models both in accuracy and AUROC. We will make all target models publicly available.

[2] https://www.kaggle.com/edqian/twitter-climate-change-sentiment-dataset

[3] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

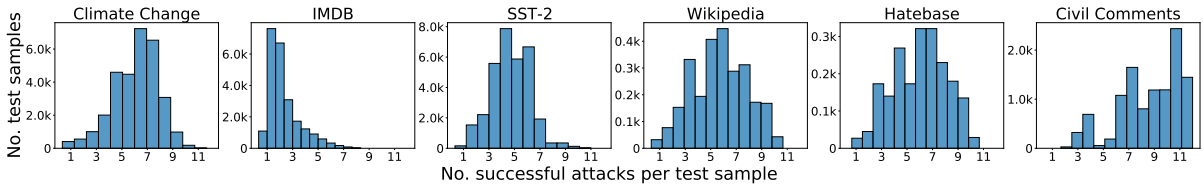


Figure 1: Histogram for number of successful attacks (out of 12), averaged across all three target models.

3.3 Attack Methods

We attempt to attack all target models using all of the attack methods implemented in two publicly available and easy-to-use toolchains: TextAttack (Morris et al., 2020) and OpenAttack (Zeng et al., 2021). Of the 16 attack methods offered by TextAttack, we find eight consistently fool the target models without crashing: (1) BAE (Garg and Ramakrishnan, 2020), DeepWordBug (Gao et al., 2018), FasterGenetic (a modified version of the attack proposed in Jia et al. (2019)), IGA (Wang et al., 2019), Pruthi (Pruthi et al., 2019), PSO (Zang et al., 2020), TextBugger (Li et al., 2019), and TextFooler (Jin et al., 2020). BAE, FasterGenetic, IGA, PSO, and TextFooler perturb text at the word level, whereas DeepWordBug, Pruthi, and TextBugger all perturb text at the character level. Of the 13 attacks implemented by OpenAttack, we find only four consistently fool the target models without crashing: Genetic (Alzantot et al., 2018), HotFlip (Ebrahimi et al., 2018), TextBugger (Li et al., 2019), and VIPER (Eger et al., 2019). Genetic perturbs text at the word level, HotFlip and TextBugger perturb text at the word level and at the character level, and VIPER perturbs text at the character level. See Table 9 for a taxonomy of all 12 attack methods used to create TCAB.

3.4 TCAB Statistics

TCAB consists of 1,539,881 successful attacks, and Table 3 shows a breakdown of attack success rates and number of successful attacks for each method.

Interestingly, the degree of input perturbation and attack-success frequency are only somewhat correlated (Figure 2). For example, DeepWordBug perturbs just 20% of the input words on average, but generates more successful attacks than any other method. The four attack methods from OpenAttack perturb more words in the input than any of the TextAttack methods.

For Civil Comments, many instances were very easy to manipulate successfully (Figure 1: far right), and it was not uncommon for all 12 attackers to successfully perturb the same instance. In contrast, it was quite rare for more than three of the attackers to be successful on any IMDB instance.

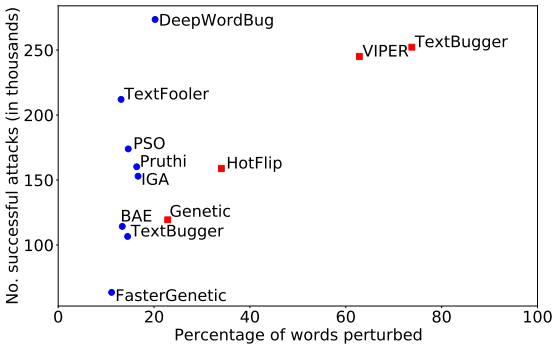


Figure 2: Average percentage of words perturbed per successful attack vs. the number of successful attacks across all domains, with TextAttack as blue circles and OpenAttack as red squares.

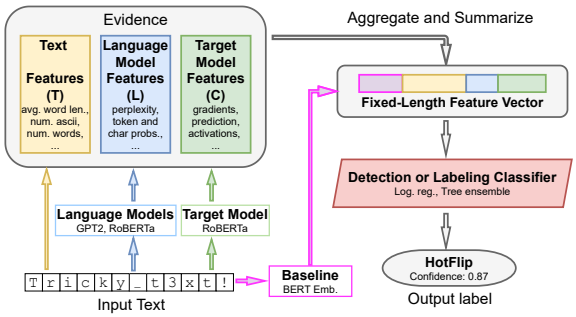


Figure 3: Pipeline for attack detection and labeling.

Of the three target-model architectures, XLNet was the most robust — it was successfully attacked 57% of the time (this percentage is an average across all attack attempts made against all XLNet models). BERT and RoBERTa were similar in robustness, both being fooled 60% of the time.

4 Feature Sets for Attack Identification

At a high level, our pipeline (Figure 3) extracts three types of features from the input (T, L, C), aggregates them into a fixed-length vector, and uses a trained classifier to identify determine the most probable attack method (*labeling*) or the probability that the instance is adversarial (*detection*).

Attack Method	Clim. Cha.	IMDB	SST-2	Wikipedia	Hatebase	Civ. Com.
BAE (Garg and Ramakrishnan, 2020)	52 (21.4k)	36 (5.1k)	68 (4.0k)	61 (4.5k)	61 (3.6k)	71 (21.6k)
DeepWordBug (**DWB**) (Gao et al., 2018)	86 (80.9k)	74 (16.6k)	79 (74.1k)	79 (5.8k)	76 (4.6k)	99 (30.0k)
FasterGenetic (**FG**) (Jia et al., 2019)	38 (14.1k)	11 (2.2k)	30 (1.8k)	32 (2.4k)	33 (2.0k)	66 (19.9k)
Genetic* (**Gn.***) (Alzantot et al., 2018)	67 (24.2k)	46 (11.7k)	34 (29.5k)	45 (3.4k)	13 (0.8k)	80 (24.3k)
HotFlip* (**HF***) (Ebrahimi et al., 2018)	52 (49.5k)	36 (12.3k)	42 (39.5k)	37 (2.8k)	35 (2.1k)	75 (22.6k)
IGA (Wang et al., 2019)	52 (49.1k)	0 (0)	59 (54.3k)	0 (0)	54 (3.7k)	64 (19.3k)
Pruthi (**Pr.**) (Pruthi et al., 2019)	43 (40.7k)	19 (5.2k)	59 (55.9k)	35 (2.6k)	40 (2.4k)	67 (20.3k)
PSO (Zang et al., 2020)	59 (55.7k)	27 (9.3k)	72 (66.7k)	31 (2.3k)	35 (2.1k)	62 (18.7k)
TextBugger* (**TB***) (Li et al., 2019)	81 (75.9k)	79 (33.1k)	65 (61.4k)	74 (5.5k)	54 (3.2k)	97 (29.3k)
TextBugger (**TB**) (Li et al., 2019)	74 (6.3k)	57 (8.0k)	68 (4.0k)	65 (4.8k)	56 (3.4k)	95 (28.8k)
TextFooler (**TF**) (Jin et al., 2020)	92 (86.5k)	51 (9.9k)	94 (5.5k)	82 (6.0k)	83 (5.0k)	98 (29.5k)
VIPER* (**VIP***) (Eger et al., 2019)	62 (58.7k)	88 (38.8k)	63 (59.6k)	66 (4.9k)	67 (4.0k)	75 (22.8k)

Table 3: Percentage (and number) of successful attacks across all three target models. Attack methods with an "*" are from the OpenAttack toolchain, those without are from the TextAttack toolchain.

Text Properties, T We use BERT (Devlin et al., 2019) to generate contextualized embeddings of the input as a baseline set of features (c.f. (Zhou et al., 2019)). Second, we extract features from the input text such as length, non-ascii character count, token casing/shape, punctuation marks, and other surface-level characteristics that may have been modified by the attacks.

Language Model Properties, L We compute the probability and rank of each token using RoBERTa, and the perplexity of the input sequence using GPT-2 (Radford et al., 2019). These features identify structures in the language of the input text, such as ungrammatical, awkward, or generic phrasing.

Target Model Properties, C We use the target model's output posteriors, node activations, gradients, and saliency (gradients w.r.t. input tokens) to capture any changes in the target model due to deceptive input. This measures the effect of deceptive text on the target classifier.

Aggregation For token-level properties, such as language model probability and rank, we compute mean, variance, and quantiles, both across the entire input sequence and within different input regions (first 25%, middle 50%, last 75%). This gives us a fixed-length feature vector. For a complete and detailed list of all properties, see §A.1.

5 Experimental Evaluation

We conduct extensive evaluations on attack detection and labeling to determine the effectiveness of our proposed TLC features and the relative difficulty of identifying attacks in the TCAB dataset.

5.1 Setup

We apply weight balancing and oversampling so that each class in the TCAB dataset is equally represented. Since the TextBugger attack was implemented by both TextAttack and OpenAttack, we merge the instances to consider them a single attack, resulting in 11 attacks.

As baselines, we train a standard logistic regression classifier and a gradient-boosting tree ensemble (Ke et al., 2017, LightGBM) applied to BERT features. We also train these classifiers with different combinations of our features, that is:

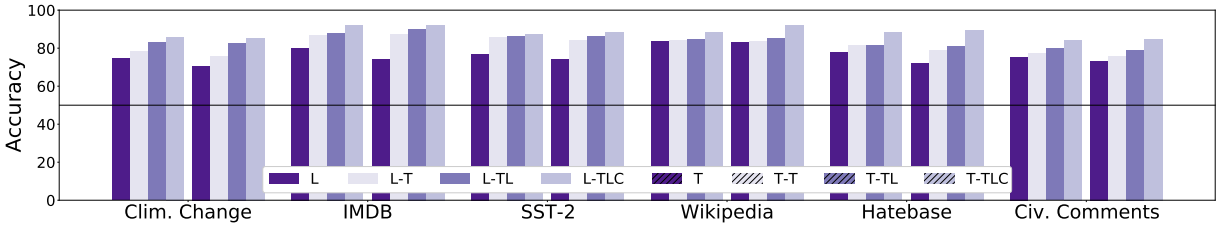
- L/T: baseline **L**inear or **T**ree ensemble model with BERT representation.
- L/T-T: includes hand-crafted **T**ext properties.
- L/T-TL: includes **L**anguage model properties.
- L/T-TLC: includes target **C**lassifier properties.

5.2 Detection and Labeling Methods

Considering the scenario where the attack methods, domain tasks, and target models are all known ahead of time, we test the ability of our methods to identify the presence of any attack in two categories: binary classification of separating perturbed texts versus the clean ones; and multiclass labeling in which each label corresponds to either a different attack or the original clean text.

When separating all attacks from clean data (Figure 4a, complete results in the Appendix, Table 7), we see strong results across all target classifiers. T shows a small improvement over L in most cases; additionally, L and T with TLC features outperform other feature ablations in nearly all cases, and outperform the BERT baseline (L and T) by 8% and 13% on average, respectively. When labeling clean data and every attack, we see substantial improvement over the baseline (Figure 4b, complete results

(a) Attack Detection (2 classes)



(b) Attack Labeling (12 classes)

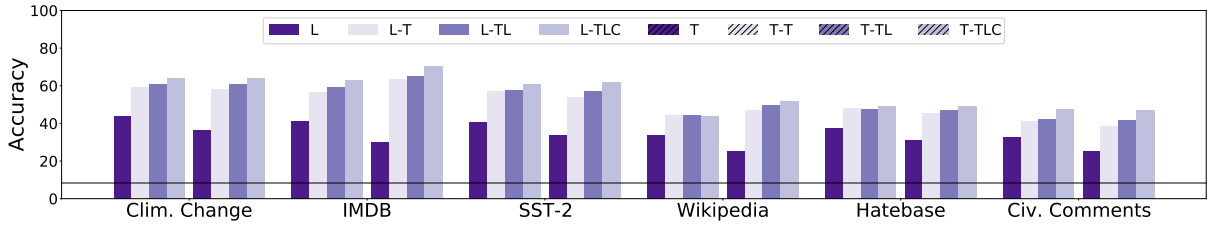


Figure 4: **Attack detection and labeling** results averaged across the three target classifiers: BERT, RoBERTa, and XLNet (see Tables 7 and 8 in the Appendix for unaggregated accuracy numbers). Horizontal lines represent random baseline performance.

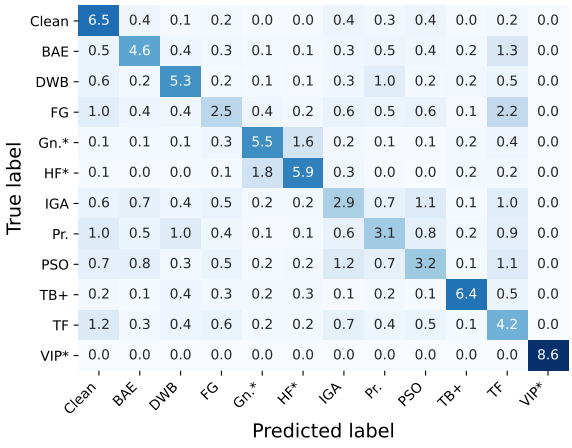


Figure 5: Confusion matrix results using T-TLC to label attacks; each entry represents the percentage of predictions made, averaged over all target models and datasets; confusion matrices for each dataset are in §B.5.

are in the Appendix, Table 8). T continues to show a small improvement over L due to its greater flexibility. L and T with language model and target model features outperform the BERT baseline (L and T) by 20% and 23% on average, respectively.

Figure 5 shows the confusion matrix of predictions for the attack labeling task. We observe word-level attacks such as HF*, PSO, and BAE are easier to distinguish than most character-level attacks, due to the effectiveness of the language-model features. This finding aligns with our results in Figure 4b where text features yield the least accurate predictions among all ablations. The one exception is VIP*, a character-level attack that uses a unique visual character embedding method, which is easy to detect and label. We also observe the model tends to confuse GN* and HF* attacks. On average, FG, Pr., and PSO tend to be the most difficult attacks to detect across all datasets. Confusion matrices for each individual dataset are in the Appendix, §B.5.

5.3 Generalizing to New Target Model

We test the ability of our methods to detect and label attacks when the target model is *not* known ahead of time by training our detection/labeling models on attacks aimed at two out of the three possible target classifiers (BERT, RoBERTA, XLNet) and evaluating them on attacks aimed at the remaining held-out target model. We compare the performance of L and T models with TLC features. Table 4 shows our methods generalize well to unseen target models for both attack detection (85% and 87% acc. for L and T, on average) and attack labeling (54% and 57% acc. for L and T).

5.4 Generalizing to New Attack Method

We also test the ability of our attack detection methods to identify new attacks by training them to detect all attacks except one, and then measuring the accuracy of the "any attack" detector on separating a new, previously unseen attack from clean data. We chose three attacks from TextAttack and OpenAttack: Pr., FG and VIP*. We isolate them and detect one at a time against RoBERTa. We compare the performance of L and T models with TLC features; average accuracy for each dataset is

	Attack Detection (2 classes)						Attack Labeling (12 classes)					
Held-Out Target:	**BERT**		**RoBERTA**		**XLNet**		**BERT**		**RoBERTA**		**XLNet**	
Dataset	L-TLC	T-TLC	L-TLC	T-TLC	L-TLC	T-TLC	L-TLC	T-TLC	L-TLC	T-TLC	L-TLC	T-TLC
Climate Change	**83.5**	83.1	**86.4**	86.0	**82.8**	81.7	**56.1**	55.5	63.5	**64.6**	59.2	**59.5**
IMDB	88.1	**89.4**	80.1	**82.3**	**87.6**	85.2	59.6	**63.8**	60.5	**63.1**	52.3	**52.9**
SST-2	88.5	**88.9**	88.5	**88.7**	85.6	**87.4**	60.4	**60.9**	**61.1**	60.8	60.3	**61.7**
Wikipedia	89.0	**92.4**	87.6	**89.4**	87.9	**90.8**	49.2	**54.6**	50.5	**56.5**	49.7	**56.3**
Hatebase	82.3	**86.1**	84.1	**87.9**	82.5	**83.6**	46.6	**51.5**	49.1	**53.7**	47.7	**51.9**
Civil Comments	83.8	**85.4**	82.8	**84.7**	83.7	**85.4**	**50.8**	50.5	**51.0**	**51.0**	50.9	**51.3**

Table 4: **Held-out target model**. Accuracy for the detection and labeling tasks evaluated on a held-out target model; f^D and f^L are trained on attacks aimed at two out of the three possible target models (BERT, RoBERTA, XLNet), and tested on attacks aimed at the remaining held-out target model. For example, the BERT columns represent accuracy for models trained on attacks aimed at RoBERTA and XLNet, and evaluated on attacks aimed at BERT.

	Pr.		FG		VIP*	
Dataset	L-TLC	T-TLC	L-TLC	T-TLC	L-TLC	T-TLC
Clim. Change	79.1	**80.7**	80.7	**84.6**	**85.7**	82.9
IMDB	**75.2**	69.3	**89.5**	82.6	66.4	**87.4**
SST-2	75.3	**86.7**	79.7	**92.7**	74.2	**88.2**
Wikipedia	84.9	**89.0**	84.0	**91.8**	**88.9**	87.9
Hatebase	93.3	**98.3**	92.7	**98.0**	**97.0**	95.2
Civ. Comm.	75.6	**80.6**	75.0	**81.5**	**75.2**	66.1

Table 5: **Held-out attacks**. Balanced detection accuracy of clean vs. each held-out attack. Each model is trained to separate clean instances from all attacks (except held-out ones) and tested on separating clean instances from each held-out attack.

presented in Table 5. Our detection methods show good detection accuracy on heldout attacks (82% acc. for L and 86% acc. for T), with T holding a slight advantage over L in the majority of cases.

5.5 Analyzing Feature Importances

We use the following as the contribution of each feature (or feature set) to predictions of each class:

$$I(A, c) = \frac{1}{m} \sum_{j=1}^{m} |x_A^{(j)} \cdot w_A^{(c)}|, \qquad (1)$$

in which c is a class (an attack or *clean*), m is the number of test instances, and x_A and w_A denote the segment of the instance and weight vectors associated with feature or feature set A, respectively.

Figure 6 shows the contribution of the top six features from the L-TLC (RoBERTa) attack labeling model for Hatebase (see Table 14 in the Appendix for all feature contributions). Target **C**lassifier properties (activations and gradients) and BERT representations show consistent heavy impact across all attacks and clean data. This is consistent with our findings described in §5.2 where L and T models with TLC features outperform other feature ablations in nearly all cases.

Figure 6: Top 5 Features (and their contributions, see Eq. 1) for each attack method for the L-TLC (RoBERTa) labeling model on the Hatebase dataset. Gray indicates feature is not in the top 5.

6 Related Work

We now review prior work on making models robust to adversarial attacks, detecting adversarial examples, and frameworks and datasets for training and evaluating the robustness of NLP models.

Building Robust Models Much of the work on defending against textual adversarial attacks consists of building more robust predictive models through some form of adversarial training. Liu et al. (2020b) and Tan et al. (2020) augment their datasets with adversarial examples, Dong et al. (2021); Wang et al. (2021a); Ivgi and Berant (2021); Yoo and Qi (2021); Wang et al. (2021d); Miyato et al. (2017); Zhu et al. (2020) train models on adversarial examples generated in an online fashion, and Dinan et al. (2019) uses humans-in-the-loop to generate high-quality adversarial examples. Malykh (2019); Jones et al. (2020); Liu et al. (2020a) encode the input to leverage the fact that perturbations will be close to the original unperturbed input.

Other methods of creating robust models have been proposed as well. Jiang et al. (2020) and Li et al. (2016) modify the training objecive to control

the complexity of the model. Jones et al. (2020) introduce a robust encoding layer and Ye et al. (2020) develop a randomized smoothing method; both provide some degree of guaranteed robustness even for large models like BERT. Huang et al. (2019) and Jia et al. (2019) both use interval bound propagation to train models that have guaranteed robustness to word substitutions. Shi et al. (2020) develop a new robustness verification algorithm that applies to transformers and produces tighter bounds than naive interval bound propagation.

Perturbation Detection In contrast to building more robust models, Zhou et al. (2019) train a model using contextualized BERT features to detect token/word perturbations and attempt to fix perturbed instances by replacing any perturbed word/token with a suitable replacement from a learned input space. Similarly, Xie et al. (2021) propose categorizing adversarial attacks using BERT sentence embeddings and target model activations. Pruthi et al. (2019) use a word-recognition model to combat word-mispelling attacks, and Mozes et al. (2021) compare inputs before and after replacing infrequent words with more frequent words to detect word-substitution attacks. Le et al. (2021) inject multiple trapdoors into textual classifiers, baiting attackers with local optima to detect universal triggers (Wallace et al., 2019). Hovy (2016) train a logistic regression model to detect fake reviews using word n-grams plus review meta-information for improved performance. Li et al. (2021a) show that adversarial examples often have high perplexity, as measured by a language model.

While prior work on labeling text attacks is limited, there is analogous work on attribution of GAN-generated images. Yu et al. (2019) find that GANs generate stable "fingerprints" which can be used for fine-grained attribution. Albright and McCloskey (2019) determine if a given image was generated by a specific GAN by reversing the generation process. Since text perturbations are typically much sparser than pixel perturbations, attack attribution is much harder with short text.

Robustness Evaluation Frameworks/Datasets Evaluation frameworks such as Robustness Gym (Goel et al., 2021) and TextFLINT (Wang et al., 2021c) allow users to measure the performance of their own models on a variety of text transformations and adversaries. TextFLINT also makes available a dataset of 67,000 transformed text samples for training. Adversarial GLUE (Wang et al., 2021b) is a multi-task robustness benchmark that was created by applying 14 textual adversarial attack methods to GLUE tasks. Dynabench (Kiela et al., 2021) is a related framework for evaluating and training NLP models on adversarial examples created by human adversaries.

Of these, TCAB is most similar to Adversarial GLUE. However, TCAB was designed for a different task — attack identification rather than robustness evaluation. TCAB is also much larger than Adversarial GLUE (1.5 million fully automated attacks vs. 5,000 human-verified attacks), focuses only on classification domains, and includes multiple classification domain datasets.

7 Conclusions and Future Work

Unlike general robust training methods, attack identification attempts to understand existing "low-effort" attacks. Using our new TCAB attack identification benchmark, we find that attack detection works well and generalizes to unseen attacks. We achieve positive results for attack labeling as well, though similarities among different attacks make it difficult to determine the exact attack used from a single example. Real-world settings may actually be easier — given a group of related adversarial examples, such as multiple abusive messages from a single account, there can be more clues to determine which attack was used to create the group.

However, this is just a first step. As new attacks are developed, we hope to expand TCAB with more attacks, as well as more domain datasets. We also expect that better detection and labeling accuracies are possible with more complex classifiers. Furthermore, while the confusion matrix reveals some relationships among the attacks, we expect that more structure could be discovered, especially as the number of attacks grows. Unknown attacks could be classified within this taxonomy based on their relationship to previously seen attacks. Another challenge is finding better features that generalize across different domains. The attacks themselves are not domain-specific, but we currently need to train on attacks for a specific domain dataset (e.g., IMDB) in order to label attacks in that domain. Finally, the true test of these methods is their effectiveness "in the wild" on attacks by actual adversaries. If attack identification leads to attacker understanding, then it is a critical piece of a comprehensive defense of NLP systems.

8 Broader Impacts

Attack identification can be used to fight spam, abuse, fake news, and more; however, adversarial attacks can also be used for evading surveillance and maintaining privacy. The same tools used to learn more about spammers could also reveal information about dissidents. Since adversarial attacks have dual use, defenses against them have dual use as well. See Albert et al. (2020) for a more extensive discussion of the political dimensions of adversarial machine learning, much of which applies to attack labeling as well.

Acknowledgements

This work was supported by DARPA under agreement number HR00112090135, and utilized computational resources from the University of Oregon high performance computer, Talapas.

References

Kendra Albert, Jonathon Penney, Bruce Schneier, and Ram Kumar. 2020. Politics of adversarial machine learning. *SSRN Electronic Journal*.

Michael Albright and Scott McCloskey. 2019. Source generator attribution via inversion. In *The Computer Vision and Pattern Recognition Conference Workshops*, volume 8.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. 2006. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, page 16–25, New York, NY, USA. Association for Computing Machinery.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1310–1320. PMLR.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, November 3-7, 2019*, pages 4536–4545. Association for Computational Linguistics.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, page 67–73, New York, NY, USA. Association for Computing Machinery.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. In *Proceedings of the 9th International Conference on Learning Representations*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *Proceedings of the 2018 IEEE Security and Privacy Workshops*, pages 50–56.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6174–6181, Online. Association for Computational Linguistics.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021.

Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 42–55, Online. Association for Computational Linguistics.

Dirk Hovy. 2016. The enemy in your own camp: How well can we detect statistically-generated fake reviews – an adversarial study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 351–356, Berlin, Germany. Association for Computational Linguistics.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4083–4093, Hong Kong, China. Association for Computational Linguistics.

Geoff Hulten, Anthony Penta, Gopalakrishnan Seshadrinathan, and Manav Mishra. 2004. Trends in spam products and methods. In *First Conference on Email and Anti-Spam, July 30-31, 2004, Mountain View, California, USA*.

Maor Ivgi and Jonathan Berant. 2021. Achieving model robustness through discrete adversarial training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4120–4133.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, 32nd Innovative Applications of Artificial Intelligence Conference, and 10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2752–2765, Online. Association for Computational Linguistics.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 3149–3157, Red Hook, NY, USA. Curran Associates Inc.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Thai Le, Noseong Park, and Dongwon Lee. 2021. A sweet rabbit hole by DARCY: Using honeypots to detect universal trigger's adversarial attacks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3831–3844, Online. Association for Computational Linguistics.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021a. Contextualized perturbation for textual adversarial attack. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating adversarial text against real-world applications. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2016. Learning robust representations of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1979–1985, Austin, Texas. Association for Computational Linguistics.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021b. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3137–3147, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hui Liu, Yongzheng Zhang, Yipeng Wang, Zheng Lin, and Yige Chen. 2020a. Joint character-level word embedding and adversarial stability training to defend adversarial text. *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 34:8384–8391.

Kai Liu, Xin Liu, An Yang, Jing Liu, Jinsong Su, Sujian Li, and Qiaoqiao She. 2020b. A robust adversarial training approach to machine reading comprehension. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, 32nd Innovative Applications of Artificial Intelligence Conference, and 10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, February 7-12, 2020*, pages 8392–8400. AAAI Press.

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.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Valentin Malykh. 2019. Robust to noise models in natural language processing tasks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 10–16, Florence, Italy. Association for Computational Linguistics.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *Proceedings of the Sixth International Conference on Learning Representations*.

Mazda Moayeri and Soheil Feizi. 2021. Sample efficient detection and classification of adversarial attacks via self-supervised embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7677–7686.

John 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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186, Online. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

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

Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2020. Robustness verification for transformers. In *Proceedings of the 6th International Conference on Learning Representations*, volume abs/2002.06622.

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.

Samson Tan, Shafiq Joty, Lav R Varshney, and Min-Yen Kan. 2020. Mind your inflections! Improving NLP for non-standard English with base-inflection encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, November 3-7, 2019*, pages 2153–2162. Association for Computational Linguistics.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021a. InfoBERT: Improving robustness of language models from an information theoretic perspective. In *Proceedings of the 9th International Conference on Learning Representations*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021b. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021c. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *CoRR*, abs/1909.06723.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021d. Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

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

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex Machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, page 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Zhouhang Xie, Jonathan Brophy, Adam Noack, Wencong You, Kalyani Asthana, Carter Perkins, Sabrina Reis, Zayd Hammoudeh, Daniel Lowd, and Sameer Singh. 2021. What models know about their attackers: Deriving attacker information from latent representations. In *Proceedings of the 4th BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 69–78, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3465–3475, Online. Association for Computational Linguistics.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ning Yu, Larry Davis, and Mario Fritz. 2019. Attributing fake images to GANs: Learning and analyzing GAN fingerprints. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, pages 7555–7565.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371, Online. Association for Computational Linguistics.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4906–4915.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. FreeLB: Enhanced adversarial training for natural language understanding. In *Proceedings of the 8th International Conference on Learning Representations*.

A Algorithmic Details

A.1 Samplewise Properties

This section provides a detailed description of each feature for the text, language model, and target model properties.

Text Properties

- **BERT features**: BERT embedding representation of the input sequence.
- **No. chars.**: Number of characters.
- **No. alpha chars.**: Number of alphabet characters (a-z).
- **No. digit chars.**: Number of digit characters (0-9).
- **No. punctuation.**: Number of punctuation mark characters (“?”, “!”, etc.)
- **No. multi. spaces.**: Number of times multiple spaces appear between words.
- **No. words.**: Number of words.
- **Avg. word len.**: Mean, variance, and quantiles (25%, 50%, and 75%) of the number of characters per word for different regions of the input (first 25%, middle 50%, last 25%, entire input).
- **No. non-ascii.**: Number of non-ascii characters.
- **Cased letters.**: Number of uppercase letters, number of lowercase letters, fraction of uppercase letters, and fraction of lowercase letters.
- **Is first word lowercase.**: True if the first character of the first word is lowercase, otherwise False.
- **No. mixed-case words.**: Number of words that contain both upper and lowercase letters (not including the first letter of each word).
- **No. single lowercase letters.**: Number of single-letter-lowercase words (excluding “a” and “i”).
- **No. lowercase after punctuation.**: Number of words after a punctuation that begin with a lowercase letter.
- **No. cased word switches.**: Number of times words switch from all uppercase to all lowercase and vice versa (e.g. THIS IS a sentence THAT contains 3 switches).

Language Model Properties

- **Probability and rank**: Mean, variance, and quantiles (25%, 50% and 75%) of the token probabilities and ranks for different regions of the input (first 25%, middle 50%, last 25%, entire input) using RoBERTa, a masked language model.
- **Perplexity**: Perplexity for different regions of the input (first 25%, middle 50%, last 25%, entire input) using GPT-2, a causal language model.

Target Model Properties

For the following properties, we assume the target model to be a RoBERTa text classifier.

- **Posterior.**: Output posteriors of the target model (softmax applied to logits).
- **Gradient.** Mean, variance, and quantiles (25%, 50%, and 75%) of the gradients for each layer of the target model given different regions of the input (first 25%, middle 50%, last 25%, entire input).
- **Activation.**: Mean, variance, and quantiles (25%, 50%, and 75%) of the node activations for each layer of the target model given different regions of the input (first 25%, middle 50%, last 25%, entire input).
- **Saliency.**: Mean, variance, and quantiles (25%, 50%, and 75%) of the saliency values (gradients of the target model with respect to the input tokens) given the input.

B Experiment Details

Experiments are run on a TITAN RTX GPU with 24GB of memory and an Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.6GHz with 60GB of memory. Experiments are run using Python 3.8. Source code for generating the attack dataset and all experiments will be made public upon publication.

B.1 Domain Datasets

- Climate Change[4] consists of 62,356 tweets from Twitter pertaining to climate change. The collection of this data was funded by a Canada Foundation for Innovation JELF Grant to Chris Bauch, University of Waterloo. The goal is to determine if the text has a negative, neutral, or positive sentiment.
- IMDB consists of 50,000 highly polar movie reviews collected from `imdb.com` curated by Maas et al. (2011). The goal is to determine if the text has a negative or positive sentiment.
- SST-2 (Socher et al., 2013) contains 68,221 phrases with fine-grained sentiment labels in the parse trees of 11,855 sentences from movie reviews. The goal is to determine if the text has a negative or positive sentiment.
- Wikipedia (Talk Pages) consists of 159,686 comments (9.6% toxic) from Wikipedia editorial talk pages. The data was curated by Wulczyn et al. (2017) and made readily available by Dixon et al. (2018). The goal is to distinguish between non-toxic and toxic comments.
- Hatebase (Davidson et al., 2017) consists of 24,783 (83.2% toxic) tweets from Twitter collected via searching for tweets containing words from the lexicon provided by `hatebase.org`. The goal is to distinguish between non-toxic and toxic comments.
- Civil Comments[5] consists of 1,804,874 (8% toxic) messages collected from the platform Civil Comments. The goal is to distinguish between non-toxic and toxic comments.

For datasets without a predefined split, we use an 80/10/10 train/validation/test split.

[4] `https://www.kaggle.com/edqian/twitter-climate-change-sentiment-dataset`
[5] `https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification`

B.2 Target Models

We use the popular HuggingFace transformers library[6] to fine-tune three transformer-based models designed for text classification (BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019)) on each domain dataset. Table 6 shows the training parameters used to fine-tune/train each model. We use cross entropy as the loss function and Adam as the optimizer to train all models.

Dataset	Model	Max. len.	Learning rate	Batch size	No. epochs	Decay	Max. norm
Climate Change	BERT	250	$1e^{-5}$	64	15	0	1.0
	RoBERTa	250	$1e^{-5}$	64	15	0	1.0
	XLNet	250	$4e^{-5}$	64	15	0	1.0
IMDB	BERT	128	$4e^{-5}$	64	5	0	1.0
	RoBERTa	128	$1e^{-6}$	64	10	0	1.0
	XLNet	128	$4e^{-5}$	64	5	0	1.0
SST-2	BERT	128	$1e^{-5}$	32	5	0	1.0
	RoBERTa	128	$1e^{-5}$	32	5	0	1.0
	XLNet	128	$1e^{-5}$	32	5	0	1.0
Wikipedia	BERT	250	$1e^{-6}$	32	10	0	1.0
	RoBERTa	250	$1e^{-6}$	32	10	0	1.0
	XLNet	250	$1e^{-6}$	16	10	0	1.0
Hatebase	BERT	250	$1e^{-6}$	32	50	0	1.0
	RoBERTa	250	$1e^{-6}$	32	50	0	1.0
	XLNet	128	$1e^{-6}$	16	50	0	1.0
Civil Comments	BERT	250	$1e^{-6}$	32	10	0	1.0
	RoBERTa	250	$1e^{-6}$	32	10	0	1.0
	XLNet	128	$1e^{-6}$	16	10	0	1.0

Table 6: Training parameters used to fine-tune/train each target model. Max. len. is the maximum number of tokens fed into each model; Decay denotes the weight decay of the model.

[6] https://huggingface.co/

B.3 Detection and Attack Identification

Table 7 shows accuracy scores for the detection task. Note that the best accuracy score always occurs when all of three feature categories (TLC) are used. The boosted trees outperform logistic regression in most cases. Also note that in the cases when the target model does not match the model that was used to create the "classifier" features (i.e. the BRT and XLN columns), accuracy is only slightly worse than when the target model *does* match the model used to create them (i.e. the RBT column).

Model	Climate Change			IMDB			SST-2			Wikipedia			Hatebase			Civil Comments		
	BRT	RBT	XLN	BRT	RBT	XLN	BRT	RBT	XLN	BRT	RBT	XLN	BRT	RBT	XLN	BRT	RBT	XLN
L	73.8	76.6	73.1	76.6	83.2	80.5	75.6	75.5	79.7	82.6	84.1	84.5	77.4	76.5	79.6	75.6	75.6	75.1
L-T	78.0	80.8	76.0	84.3	90.4	85.6	86.0	86.1	84.8	83.4	85.1	84.5	82.9	82.1	79.8	76.9	77.2	77.1
L-TL	82.2	85.3	81.2	85.5	91.0	87.3	86.5	87.0	86.0	83.2	85.3	85.3	82.6	82.0	79.9	79.4	79.7	80.3
L-TLC	**85.1**	**88.6**	**84.1**	**89.5**	**95.2**	**90.6**	**86.9**	**87.8**	**87.7**	**87.9**	**90.1**	**87.8**	**87.3**	**91.2**	**86.2**	**84.3**	**84.2**	**84.4**
T	69.6	71.6	69.3	70.6	76.6	74.5	72.8	73.2	76.7	82.5	82.3	83.6	69.5	69.7	75.9	74.0	73.3	72.3
T-T	75.4	78.0	73.5	86.2	91.5	83.9	86.1	84.4	81.9	82.6	84.0	84.1	79.5	79.1	77.3	77.3	75.4	74.0
T-TL	82.0	84.5	80.9	88.4	92.9	88.2	87.0	86.5	85.2	84.4	86.0	85.4	80.3	82.4	79.3	79.7	79.0	77.6
T-TLC	**84.2**	**88.2**	**83.1**	**90.6**	**97.2**	**91.0**	**88.2**	**89.5**	**87.5**	**90.9**	**93.3**	**91.1**	**89.5**	**93.0**	**85.0**	**84.5**	**84.4**	**84.6**

Table 7: Accuracy for clean vs. all attacks (binary). Attacked samples and clean samples are balanced 50/50. The best number in each column for each group (L vs. T) is bolded.

Table 8 shows accuracy scores for the multiclass attack labeling task. Note that the best accuracy score almost always occurs when all of three feature categories (TLC) are used. As with detection, boosted trees usually outperforms logistic regression. And also as with the detection figures, note that in the cases when the target model does not match the model that was used to create the "classifier" features (i.e. the BRT and XLN columns), accuracy is only slightly worse than when the target model *does* match the model used to create them (i.e. the RBT column).

Model	Climate Change			IMDB			SST-2			Wikipedia			Hatebase			Civil Comments		
	BRT	RBT	XLN	BRT	RBT	XLN	BRT	RBT	XLN	BRT	RBT	XLN	BRT	RBT	XLN	BRT	RBT	XLN
L	37.9	48.5	45.8	38.7	39.6	45.5	38.8	39.0	44.5	32.7	34.5	34.0	38.0	38.7	35.4	29.6	30.4	38.3
L-T	53.9	64.1	59.8	55.0	55.5	59.9	56.2	56.9	57.8	43.4	**44.2**	45.1	50.5	49.8	**43.4**	38.0	38.8	46.1
L-TL	55.6	65.5	61.5	57.2	57.8	62.8	56.6	57.3	58.7	43.3	44.1	**46.3**	49.6	50.6	43.0	39.2	40.0	47.4
L-TLC	**58.9**	**68.5**	**64.8**	**60.8**	**62.0**	**65.6**	**58.9**	**61.8**	**61.4**	**44.3**	43.3	44.8	**51.5**	**52.5**	43.1	**45.2**	**45.3**	**51.9**
T	32.4	38.9	37.2	27.4	28.5	33.5	31.7	32.0	37.1	26.1	24.8	24.8	28.5	32.1	32.1	28.0	22.8	23.8
T-T	52.6	63.0	57.9	63.8	61.2	64.3	53.4	53.8	53.9	45.8	47.5	47.5	48.4	48.1	38.6	41.3	37.1	37.4
T-TL	55.4	65.6	61.5	65.8	62.5	66.4	56.6	56.6	57.1	48.0	50.0	50.0	**49.4**	49.1	41.9	44.6	40.1	40.3
T-TLC	**58.4**	**69.4**	**64.5**	**69.7**	**69.3**	**70.9**	**60.9**	**62.8**	**61.7**	**50.5**	**51.5**	**52.2**	47.9	**55.1**	**44.7**	**44.9**	**45.4**	**50.7**

Table 8: Accuracy for the attack labeling task. Baseline accuracy is approximately 1/12 = 8.33%. The best number in each column for each group (L vs. T) is bolded.

B.4 Attack Methods

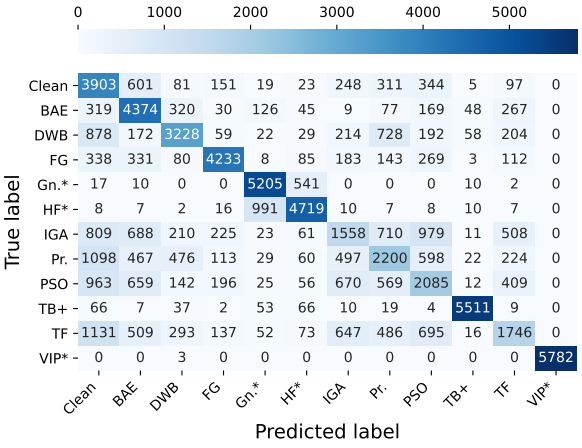
In Table 9, all of the attack methods used to create TCAB are listed along with the toolchain each attack method belongs to and the access level, linguistic constraints, and perturbation level each attack method has.

Attack method	Toolchain	Access level	Linguistic constraints	Perturbation level
BAE	TextAttack	Black box	Yes	Word
DeepWordBug	TextAttack	Gray box	No	Char
FasterGenetic	TextAttack	Gray box	Yes	Word
Genetic	OpenAttack	Gray box	Yes	Word
HotFlip	OpenAttack	White box	Yes	Char
IGA	TextAttack	Gray box	Yes	Word
Pruthi	TextAttack	Gray box	No	Word
PSO	TextAttack	Black box	Yes	Word
TextBugger	TextAttack	Black box	Yes	Char
TextBugger	OpenAttack	White box	Yes	Word + Char
TextFooler	TextAttack	Black box	Yes	Word
VIPER	OpenAttack	Black box	No	Char

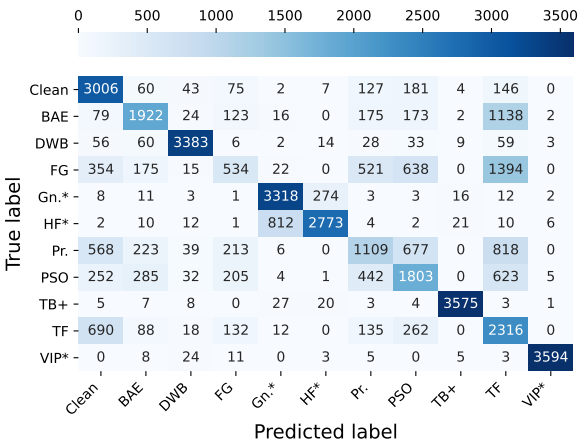
Table 9: The 12 attack methods used to create TCAB.

B.5 Attack Labeling: Confusion Matrix

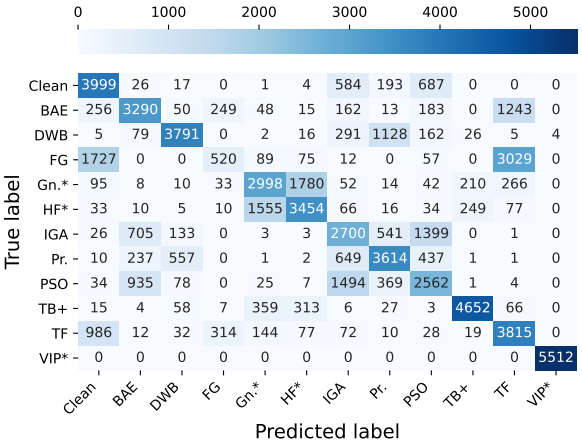
Figure 7 shows the confusion matrix results for each domain dataset using T-TLC to label attacks, averaged over all target models.



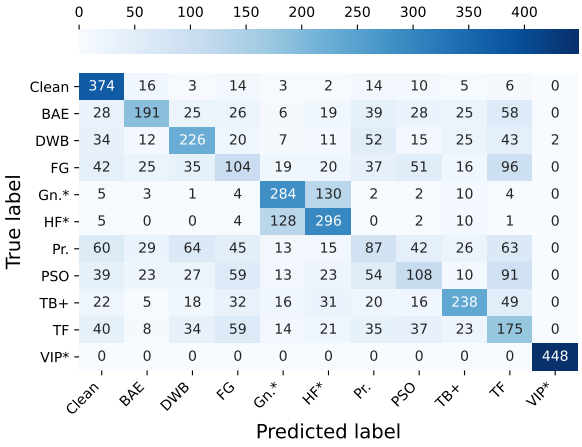
(a) Climate Change



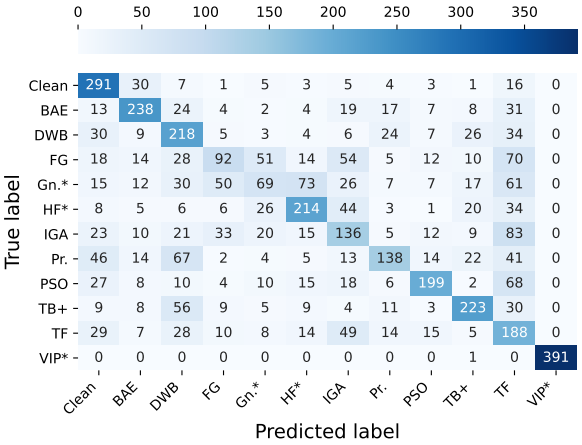
(b) IMDB: IGAWang is not included since it has no successful attacks on the Wikipedia dataset.



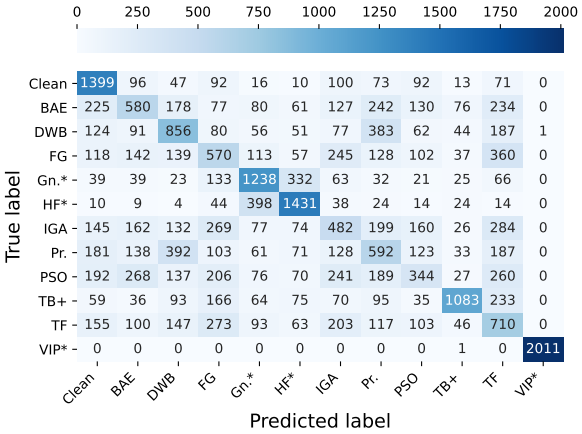
(c) SST-2



(d) Wikipedia: IGAWang is not included since it has no successful attacks on the Wikipedia dataset.



(e) Hatebase



(f) Civil Comments

Figure 7: Confusion matrix results using the T-TLC model to label attacks, averaged over all target models.

B.6 Attack Samples

We present Tables 10, 11 and 12 with a few successful attack samples on the same pieces of text from the SST and the Wikipedia datasets predicted by XLNet model attacked by a variety of attackers respectively; and Table 13 with a few successful attack samples on different pieces of text from SST predicted by RoBERTa model and attacked by BAE.

Attack	Text	Label	Confidence
Original	Part comedy, part drama, the movie winds up accomplishing neither in full, and leaves us feeling touched and amused by several moments and ideas, but nevertheless dissatisfied with the movie as a whole.	Negative	89.2%
BAE	Part comedy, part drama, the movie winds up accomplishing neither in full, and leaves us feeling touched and amused by several moments and ideas, but nevertheless satisfied with the movie as a whole.	Positive	84.4%
DeepWordBug	Part comedy, part drama, the movie wirnds up accomplishing enither in full, and leaves us feeling touched and amused by several moments and ideas, but nevertheless dissatAsfied with the movie as a whole.	Positive	60.1%
FasterGenetic	Part comedy, part drama, the movie winds up accomplishing or in full, and leaves us feeling touched and amused by several moments and ideas, but notwithstanding displeased with the movie as a whole.	Positive	96.2%
PSO	Part comedy, part drama, the movie winds up accomplishing neither in full, and leaves us feeling touched and amused by several moments and ideas, but nevertheless dazzling with the movie as a whole.	Positive	94.8%
Pruthi	Part comedy, part drama, the movie winds up accomplishing neither in full, and leaves us feeling touched and amused by several moments and ideas, but neverthelwss dissatisfied with the movie as a whole.	Positive	95.5%
TextFooler	Part comedy, part drama, the movie gusty up accomplishing neither in full, and leaves us feeling touched and amused by several moments and ideas, but nevertheless incensed with the movie as a whole.	Positive	74.1%
Genetic	Portions comedy, part theatrical, the movie winds up achieving nor in full, and leaves us feeling touched and tickled by diverse moments and thought, but however dissatisfied with the movie as a whole.	Positive	58.3%
TextBugger(TA)	Part comedy, part drama, the movie winds up accomplishing nor in full, and leaves us feeling touched and amused by several moments and ideas, but nevertheless ‘Åissatisfied with the movie as a whole.	Positive	53.3%

Table 10: Attack Samples of SST against XLNet.

Attack	Text	Label	Confidence
Original	YOu moron, do not accuse me to go and study. The fools and used like condom people with small or with no brains are YOU! . . .	Toxic	98.3 %
BAE	YOu suke, do not accuse me to go and study. The fools and used like condom people with small or with no brains are YOU! . . .	Non-toxic	78.5%
DeepWordBug	Yu mBron, do not accuse me to go and study. The fools and used like condom people with small or with no brains are YOU! . . .	Non-toxic	76.0%
TextBugger(TA)	YOu moon, do not accuse me to go and study. The fools and used like condom people with small or with no brains are YOU! . . .	Non-toxic	68.4%
TextFooler	tu cretin, do not accuse me to go and study. The fools and used like condom people with small or with no brains are YOU! . . .	Non-toxic	67.2%
TextBugger(OA)	You moron , do not accuse me to go and studying. The fols and utilized loves condom people with little or with no brains zre YOU! . . .	Non-toxic	91.1%
Genetic	You cretin , do not accuse me to go and study. The fools and usage like condom people with small or with no brains are you! . . .	Non-toxic	54.4%

Table 11: Attack Samples of Wikipedia against XLNet Example 1.

Attack	Text	Label	Confidence
Original	Why do not the two of you, the mentally r***ded Jennygirl and the omnipotent Richie Rich, go together to Harrods and . . . when you come back after they would be kicking off your dipshit spinning incapable ass from the central perk. What a pity. Receive all my sympathy at one time. . . .	Toxic	63.7 %
Pruthi	Why do not the two of you, the mentally regarded Jennygirl and the omnipotent Richie Rich, go together to Harrods and . . . when you come back after they would be kicking off your dipshit spinning incapable ass from the central perk. What a pity. Receive all my sympathy at one time.. . .	Non-toxic	82.1%
PSO	Why do not the two of you, the mentally checked Jennygirl and the omnipotent Richie Rich, go together to Harrods and . . . when you come back after they would be kicking off your dipshit spinning incapable ass from the central perk. What a pity. Receive all my sympathy at one time. . . .	Non-toxic	85.7%
BAE	Why do not the two of you, the mentally r***ded Jennygirl and the omnipotent Richie Rich, go together to Harrods and . . . when you come back after they would be kicking off your rella spinning incapable ass from the central perk. What a pity. Receive all my sympathy at one time. . . .	Non-toxic	92.2%
TextFooler	Why do not the two of you, the mentally r***ded Jennygirl and the omnipotent Richie Rich, go together to Harrods and . . . when you come back after they would be kicking off your cretin spinning incapable ass from the central perk. What a pity. Receive all my sympathy at one time. . . .	Non-toxic	91.2%
Genetic	Why do not the two of you, the mentally backward Jennygirl and the omnipotent Richie Rich, go together to Harrods and . . . when you come back after they would be kicking off your dipshit spinning incapable ass from the central perk. What a pity. Receive all my sympathy at one time. . . .	Non-toxic	62.3%
DeepWordBug	Why do not the two of you, the mentally retarded Jennygirl and the omnipotent Richie Rich, go together to Harrods and . . . when you come back after they would be kicking off your dipsgit spinning incapable ass from the central perk. What a pity. Receive all my sympathy at one time. . . .	Non-toxic	93.7%

Table 12: Attack Samples of Wikipedia against XLNet Example 2.

Original Text	Original	Perturbed Text	Perturbed
Watching Trouble Every Day , at least if you do n't know what 's coming , is like biting into what looks like a juicy , delicious plum on a hot summer day and coming away with your mouth full of rotten pulp and living worms .	Negative, 92.6%	Watching Trouble Every Day , at least if you do n't know what 's coming , is like biting into what looks like a juicy , delicious plum on a hot summer day and coming away with your mouth full of fresh pulp and living worms .	Positive, 55.3%
With a story inspired by the tumultuous surroundings of Los Angeles , where feelings of marginalization loom for every dreamer with a burst bubble , The Dogwalker has a few characters and ideas , but it never manages to put them on the same path .	Negative, 91.1%	With a story inspired by the tumultuous surroundings of Los Angeles , where feelings of marginalization loom for every dreamer with a burst bubble , The Dogwalker has a few characters and ideas , but it still manages to put them on the same path .	Positive, 98.2%
To imagine the life of Harry Potter as a martial arts adventure told by a lobotomized Woody Allen is to have some idea of the fate that lies in store for moviegoers lured to the mediocrity that is Kung Pow : Enter the Fist .	Negative, 94.0%	To experience the life of Harry Potter as a martial arts adventure told by a mad Woody Allen is to have some idea of the fate that lies in store for moviegoers lured to the mediocrity that is Kung Pow : Enter the Fist .	Positive, 51.1%
This is n't a narrative film – I do n't know if it 's possible to make a narrative film about September 11th , though I 'm sure some will try – but it 's as close as anyone has dared to come .	Positive, 63.8%	This is n't a narrative film – I do t know if it 's possible to make a narrative film about September 11th , though I 'm sure some will try – but it 's as close as anyone has dare to come .	Negative, 66.6%
Though Mama takes a bit too long to find its rhythm and a third-act plot development is somewhat melodramatic , its ribald humor and touching nostalgia are sure to please anyone in search of a Jules and Jim for the new millennium .	Positive, 98.2%	as Mama takes a bit too long to find its rhythm and a third-act plot development is somewhat questionable , its ribald humor and touching nostalgia are sure to deter anyone in search of a Jules and Jim for the new millennium .	Negative, 54.7%

Table 13: Attack Samples of SST Attacked by BAE against RoBERTa. The "Original" column contains the original label and the prediction confidence. The "Perturbed" column contains the perturbed label and the perturbation confidence.

Feature	Attack											
	Clean	BAE	DWB	FG	Gn.*	HF*	IGA	Pr.	PSO	TB	TF	VIP*
TP AVG WORD LENGTH	1.1	1.0	2.1	2.4	2.9	3.6	4.2	4.7	5.4	5.8	6.4	6.5
TP BERT	6.3	3.7	9.6	12.8	15.6	18.0	20.9	24.6	28.1	30.8	33.9	34.6
TP FIRST WORD LOWERCASE	0.2	0.2	0.3	0.3	0.5	0.7	0.7	0.7	0.7	0.7	0.7	0.7
TP NUM ALPHA CHARS	0.0	0.0	0.0	0.2	0.2	0.2	0.2	0.2	0.7	0.7	1.3	1.4
TP NUM CASED LETTERS	0.5	0.2	0.9	1.6	1.7	4.2	4.5	4.6	4.9	5.9	7.0	7.0
TP NUM CASED WORD SWITCHES	0.0	0.0	0.0	0.2	0.2	1.3	1.7	1.7	1.7	1.9	2.1	2.1
TP NUM CHARS	0.0	0.0	0.0	0.0	0.0	0.4	0.4	0.4	0.4	0.4	0.4	0.5
TP NUM DIGITS	0.0	0.0	0.5	0.5	0.5	0.5	0.5	0.8	0.8	0.8	1.0	1.0
TP NUM LOWERCASE AFTER PUNCTUATION	0.2	0.0	0.2	0.2	0.3	0.3	0.3	0.4	0.4	0.5	0.5	0.5
TP NUM SINGLE LOWERCASE LETTERS	0.6	0.4	1.2	1.2	1.2	1.2	1.4	1.6	1.6	1.8	1.9	1.9
TP NUM MIXED CASE WORDS	0.4	0.1	2.6	2.9	3.2	4.7	5.0	5.9	6.0	7.3	8.2	8.2
TP NUM MULTI SPACES	0.0	0.0	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2	0.2
TP NUM NON ASCII	0.2	0.0	4.2	4.2	4.2	4.2	4.5	5.2	5.2	5.2	6.7	8.9
TP NUM PUNCTUATION	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TP NUM WORDS	1.1	1.1	2.7	3.0	3.0	4.7	5.3	6.8	6.8	6.8	10.0	10.0
LM PERPLEXITY	0.2	0.2	0.2	0.3	0.3	0.3	0.4	0.5	0.5	0.6	0.8	0.8
LM PROBA AND RANK	1.5	1.1	2.2	3.2	3.9	4.3	5.0	5.8	6.2	6.5	7.0	7.0
TM ACTIVATION	9.7	6.2	15.4	19.5	23.5	28.6	33.4	39.4	44.1	49.8	54.6	57.3
TM GRADIENT	9.3	5.7	13.1	16.9	20.9	23.0	27.7	33.0	36.0	39.9	44.9	45.1
TM POSTERIOR	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TM SALIENCY	0.2	0.2	0.2	0.3	0.4	0.8	1.2	2.1	2.3	2.9	3.0	3.0

Table 14: All Features (and their contributions, see Eq. 1) for each attack method for the L-TLC (RoBERTa) labeling model on the Hatebase dataset.