Reevaluating Adversarial Examples in Natural Language

John X. Morris,* **Eli Lifland**,* **Jack Lanchantin, Yangfeng Ji, Yanjun Qi**
Department of Computer Science, University of Virginia
{jm8wx, edl9cy, jjl5sw, yj3fs, yq2h}@virginia.edu

Abstract

State-of-the-art attacks on NLP models lack a shared definition of what constitutes a successful attack. These differences make the attacks difficult to compare and hindered the use of adversarial examples to understand and improve NLP models. We distill ideas from past work into a unified framework: a successful natural language adversarial example is a perturbation that fools the model and follows four proposed linguistic constraints. We categorize previous attacks based on these constraints. For each constraint, we suggest options for human and automatic evaluation methods. We use these methods to evaluate two state-of-the-art synonym substitution attacks. We find that perturbations often do not preserve semantics, and 38% introduce grammatical errors. Next, we conduct human studies to find a threshold for each evaluation method that aligns with human judgment. Human surveys reveal that to successfully preserve semantics, we need to significantly increase the minimum cosine similarities between the embeddings of swapped words and between the sentence encodings of original and perturbed sentences. With constraints adjusted to better preserve semantics and grammaticality, the attack success rate drops by over 70 percentage points. [1]

1 Introduction

One way to evaluate the robustness of a machine learning model is to search for inputs that produce incorrect outputs. Inputs intentionally designed to fool deep learning models are referred to as adversarial examples (Goodfellow et al., 2017). Adversarial examples have successfully tricked deep neural networks for image classification: two images that look exactly the same to a human receive completely different predictions from the classifier (Goodfellow et al., 2014).

Original: Skip the **film** and buy the philip glass soundtrack cd.	Prediction: **Negative ✗**
Adversarial: Skip the **films** and buy the philip glass soundtrack cd.	Prediction: **Positive ✓**

Figure 1: An adversarial example generated by TFADJUSTED for BERT fine-tuned on the Rotten Tomatoes sentiment analysis dataset. Swapping a single word causes the prediction to change from positive to negative.

While applicable in the image case, the idea of an indistinguishable change lacks a clear analog in text. Unlike images, two different sequences of text are never entirely indistinguishable. This raises the question: if indistinguishable perturbations are not possible, what are adversarial examples in text?

The literature contains many potential answers to this question, proposing varying definitions for successful adversarial examples (Zhang et al., 2019). Even attacks with similar definitions of success often measure it in different ways. The lack of a consistent definition and standardized evaluation has hindered the use of adversarial examples to understand and improve NLP models. [2]

Therefore, we propose a unified definition for successful adversarial examples in natural language: perturbations that both fool the model and fulfill a set of linguistic constraints. In Section 2, we present four categories of constraints NLP adversarial examples may follow, depending on the context: semantics, grammaticality, overlap, and non-suspicion to human readers.

* Equal contribution
[1] Our code and datasets are available here.
[2] We use 'adversarial example generation methods' and 'adversarial attacks' interchangeably in this paper.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3829–3839

By explicitly laying out categories of constraints adversarial examples may follow, we introduce a shared vocabulary for discussing constraints on adversarial attacks. In Section 4, we suggest options for human and automatic evaluation methods for each category. We use these methods to evaluate two SOTA synonym substitution attacks: GENETICATTACK by Alzantot et al. (2018) and TEXTFOOLER by Jin et al. (2019). Human surveys show that the perturbed examples often fail to fulfill semantics and non-suspicion constraints. Additionally, a grammar checker detects 39% more errors in the perturbed examples than in the original inputs, including many types of errors humans almost never make.

In Section 5, we produce TFADJUSTED, an attack with the same search process as TEXTFOOLER, but with constraint enforcement tuned to generate higher quality adversarial examples. To enforce semantic preservation, we tighten the thresholds on the cosine similarity between embeddings of swapped words and between the sentence encodings of original and perturbed sentences. To enforce grammaticality, we validate perturbations with a grammar checker. As in TEXTFOOLER, these constraints are applied at each step of the search. Human evaluation shows that TFADJUSTED generates perturbations that better preserve semantics and are less noticeable to human judges. However, with stricter constraints, the attack success rate decreases from over 80% to under 20%. When used for adversarial training, TEXTFOOLER's examples decreased model accuracy, but TFADJUSTED's examples did not.

Without a shared vocabulary for discussing constraints, past work has compared the success rate of search methods with differing constraint application techniques. Jin et al. (2019) reported a higher attack success rate for TEXTFOOLER than Alzantot et al. (2018) did for GENETICATTACK, but it was not clear whether the improvement was due to a better search method[3] or more lenient constraint application[4]. In Section 6 we compare the search methods with constraint application held constant. We find that GENETICATTACK's search method is more successful than TEXTFOOLER's, contrary to the implications of Jin et al. (2019).

The five main contributions of this paper are:

- A definition for constraints on adversarial perturbations in natural language and suggest evaluation methods for each constraint.
- Constraint evaluations of two SOTA synonym-substitution attacks, revealing that their perturbations often do not preserve semantics, grammaticality, or non-suspicion.
- Evidence that by aligning automatic constraint application with human judgment, it is possible for attacks to produce successful, valid adversarial examples.
- Demonstration that reported differences in attack success between TEXTFOOLER and GENETICATTACK are the result of more lenient constraint enforcement.
- Our framework enables fair comparison between attacks, by separating effects of search methods from effects of loosened constraints.

2 Constraints on Adversarial Examples in Natural Language

We define $F : \mathcal{X} \rightarrow \mathcal{Y}$ as a predictive model, for example, a deep neural network classifier. $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. We focus on adversarial perturbations which perturb a correctly predicted input, $\mathbf{x} \in \mathcal{X}$, into an input $\mathbf{x}_{adv}$. The boolean goal function $G(F, \mathbf{x}_{adv})$ represents whether the goal of the attack has been met. We define $C_1 ... C_n$ as a set of boolean functions indicating whether the perturbation satisfies a certain constraint.

Adversarial attacks search for a perturbation from $\mathbf{x}$ to $\mathbf{x}_{adv}$ which fools F by both achieving some goal, as represented by $G(F, \mathbf{x}_{adv})$, and fulfilling each constraint $C_i(\mathbf{x}, \mathbf{x}_{adv})$.

The definition of the goal function G depends on the purpose of the attack. Attacks on classification frequently aim to either induce any incorrect classification (untargeted) or induce a particular classification (targeted). Attacks on other types of models may have more sophisticated goals. For example, attacks on translation may attempt to change every word of a translation, or introduce targeted keywords into the translation (Cheng et al., 2018).

In addition to defining the goal of the attack, the attacker must decide the constraints perturbations must meet. Different use cases require different

[3] TEXTFOOLER uses a greedy search method with word importance ranking. GENETICATTACK uses a genetic algorithm.

[4] For example, TEXTFOOLER applies a minimum cosine distance of .5 between embeddings of swapped words. GENETICATTACK uses a threshold of .75.

Input, x: "Shall I compare thee to a summer's day?" – William Shakespeare, Sonnet XVIII		
Constraint	**Perturbation, $\mathbf{x}_{adv}$**	**Explanation**
Semantics	Shall I compare thee to a winter's day?	$\mathbf{x}_{adv}$ has a different meaning than $\mathbf{x}$.
Grammaticality	Shall I compares thee to a summer's day?	$\mathbf{x}_{adv}$ is less grammatically correct than $\mathbf{x}$.
Edit Distance	Sha1l i conpp$haaare thee to a 5umm3r's day?	$\mathbf{x}$ and $\mathbf{x}_{adv}$ have a large edit distance.
Non-suspicion	Am I gonna compare thee to a summer's day?	A human reader may suspect this sentence to have been modified. [1]

[1] Shakespeare never used the word "gonna". Its first recorded usage wasn't until 1806, and it didn't become popular until the 20th century.

Table 1: **Adversarial Constraints and Violations.** For each of the four proposed constraints, we show an example for which violates the specified constraint.

constraints. We build on the categorization of attack spaces introduced by Gilmer et al. (2018) to introduce a categorization of constraints for adversarial examples in natural language.

In the following, we define four categories of constraints on adversarial perturbations in natural language: semantics, grammatically, overlap, and non-suspicion. Table 1 provides examples of adversarial perturbations that violate each constraint.

2.1 Semantics

Semantics constraints require the semantics of the input to be preserved between $\mathbf{x}$ and $\mathbf{x}_{adv}$. Many attacks include constraints on semantics as a way to ensure the correct output is preserved (Zhang et al., 2019). As long as the semantics of an input do not change, the correct output will stay the same. There are exceptions: one could imagine tasks for which preserving semantics does not necessarily preserve the correct output. For example, consider the task of classifying passages as written in either Modern or Early Modern English. Perturbing "why" to "wherefore" may retain the semantics of the passage, but change the correct label from Modern to Early Modern English[5]

2.2 Grammaticality

Grammaticality constraints place restrictions on the grammaticality of $\mathbf{x}_{adv}$. For example, an adversary attempting to generate a plagiarised paper which fools a plagiarism checker would need to ensure that the paper remains grammatically correct. Grammatical errors don't necessarily change semantics, as illustrated in Table 1.

2.3 Overlap

Overlap constraints restrict the similarity between $\mathbf{x}$ and $\mathbf{x}_{adv}$ at the character level. This includes constraints like Levenshtein distance as well as n-gram based measures such as BLEU, METEOR and chRF (Papineni et al., 2002; Denkowski and Lavie, 2014; Popović, 2015).

Setting a maximum edit distance is useful when the attacker is willing to introduce misspellings. Additionally, the edit distance constraint is sometimes used when improving the robustness of models. For example, Huang et al. (2019) uses Interval Bound Propagation to ensure model robustness to perturbations within some edit distance of the input.

2.4 Non-suspicion

Non-suspicion constraints specify that $\mathbf{x}_{adv}$ must appear to be unmodified. Consider the example in Table 1. While the perturbation preserves semantics and grammar, it switches between Modern and Early Modern English and thus may seem suspicious to readers.

Note that the definition of the non-suspicious constraint is context-dependent. A sentence that is non-suspicious in the context of a kindergartner's homework assignment might be suspicious in the context of an academic paper. An attack scenario where non-suspicion constraints do not apply is illegal PDF distribution, similar to a case discussed by Gilmer et al. (2018). Consumers of an illegal PDF may tacitly collude with the person uploading it. They know the document has been altered, but do not care as long as semantics are preserved.

3 Review and Categorization of SOTA:

Attacks by Paraphrase: Some studies have generated adversarial examples through paraphrase. Iyyer et al. (2018) used neural machine translation systems to generate paraphrases. Ribeiro et al. (2018) proposed semantically-equivalent adversarial rules. By definition, paraphrases preserve semantics. Since the systems aim to generate perfect

[5] Wherefore is a synonym for why, but was used much more often centuries ago.

paraphrases, they implicitly follow constraints of grammaticality and non-suspicion.

Attacks by Synonym Substitution: Some works focus on an easier way to generate a subset of paraphrases: replacing words from the input with synonyms (Alzantot et al., 2018; Jin et al., 2019; Kuleshov et al., 2018; Papernot et al., 2016; Ren et al., 2019). Each attack applies a search algorithm to determine which words to replace with which synonyms. Like the general paraphrase case, they aim to create examples that preserve semantics, grammaticality, and non-suspicion. While not all have an explicit edit distance constraint, some limit the number of words perturbed.

Attacks by Character Substitution: Some studies have proposed to attack natural language classification models by deliberately misspelling words (Ebrahimi et al., 2017; Gao et al., 2018; Li et al., 2018). These attacks use character replacements to change a word into one that the model doesn't recognize. The replacements are designed to create character sequences that a human reader would easily correct into the original words. If there aren't many misspellings, non-suspicion may be preserved. Semantics are preserved as long as human readers can correct the misspellings.

Attacks by Word Insertion or Removal: Liang et al. (2017) and Samanta and Mehta (2017) devised a way to determine the most important words in the input and then used heuristics to generate perturbed inputs by adding or removing important words. In some cases, these strategies are combined with synonym substitution. These attacks aim to follow all constraints.

Using constraints defined in Section 2 we categorize a sample of current attacks in Table 2.

4 Constraint Evaluation Methods and Case Study

For each category of constraints introduced in Section 2, we discuss best practices for both human and automatic evaluation. We leave out overlap due to ease of automatic evaluation.

Additionally, we perform a case study, evaluating how well black-box synonym substitution attacks GENETICATTACK and TEXTFOOLER fulfill constraints. Both attacks find adversarial examples by swapping out words for their synonyms until the classifier is fooled. GENETICATTACK uses a genetic algorithm to attack an LSTM trained on the IMDB[6] document-level sentiment classification dataset. TEXTFOOLER uses a greedy approach to attack an LSTM, CNN, and BERT trained on five classification datasets. We chose these attacks because:

- They claim to create perturbations that preserve semantics, maintain grammaticality, and are not suspicious to readers. However, our inspection of the perturbations revealed that many violated these constraints.
- They report high attack success rates.[7]
- They successfully attack two of the most effective models for text classification: LSTM and BERT.

To generate examples for evaluation, we attacked BERT using TEXTFOOLER and attacked an LSTM using GENETICATTACK. We evaluate both methods on the IMDB dataset. In addition, we evaluate TEXTFOOLER on the Yelp polarity document-level sentiment classification dataset and the Movie Review (MR) sentence-level sentiment classification dataset (Pang and Lee, 2005; Zhang et al., 2015). We use $1,000$ examples from each dataset. Table 3 shows example violations of each constraint.

4.1 Evaluation of Semantics

4.1.1 Human Evaluation

A few past studies of attacks have included human evaluation of semantic preservation (Ribeiro et al., 2018; Iyyer et al., 2018; Alzantot et al., 2018; Jin et al., 2019). However, studies often simply ask users to simply rate the "similarity" of $\mathbf{x}$ and $\mathbf{x}_{adv}$. We believe this phrasing does not generate an accurate measure of semantic preservation, as users may consider two sentences with different semantics "similar" if they only differ by a few words. Instead, users should be explicitly asked whether changes between $\mathbf{x}$ and $\mathbf{x}_{adv}$ preserve the meaning of the original passage.

We propose to ask human judges to rate if meaning is preserved on a Likert scale of 1-5, where 1 is "Strongly Disagree" and 5 is "Strongly Agree" (Likert, 1932). A perturbation is semantics-preserving if the average score is at least ϵ_{sem}. We propose

[6] https://datasets.imdbws.com/

[7] We use "attack success rate" to mean the percentage of the time that an attack can find a successful adversarial example by perturbing a given input. "After-attack accuracy" or "accuracy after attack" is the accuracy the model achieves after all successful perturbations have been applied.

Selected Attacks Generating Adversarial Examples in Natural Language	Semantics	Grammaticality	Edit Distance	Non-Suspicion
Synonym Substitution. (Alzantot et al., 2018; Kuleshov et al., 2018; Jin et al., 2019; Ren et al., 2019)	✓	✓	✓	✓
Character Substitution. (Ebrahimi et al., 2017; Gao et al., 2018; Li et al., 2018)	✓	×	✓	✓
Word Insertion or Removal. (Liang et al., 2017; Samanta and Mehta, 2017)	✓	✓	✓	✓
General Paraphrase. (Zhao et al., 2017; Ribeiro et al., 2018; Iyyer et al., 2018)	✓	✓	×	✓

Table 2: **Summary of Constraints and Attacks.** This table shows a selection of prior work (rows) categorized by constraints (columns). A "✓" indicates that the respective attack is supposed to meet the constraint, and a "×" means the attack is not supposed to meet the constraint.

Constraint Violated	Input, $\mathbf{x}$	Perturbation, $\mathbf{x}_{adv}$
Semantics	Jagger, Stoppard and director Michael Apted deliver a riveting and surprisingly romantic ride.	Jagger, Stoppard and director Michael Apted deliver a baffling and surprisingly sappy motorbike.
Grammaticality	A grating, emaciated flick.	A grates, lanky flick.
Non-suspicion	Great character interaction.	Gargantuan character interaction.

Table 3: **Real World Constraint Violation Examples.** Perturbations by TEXTFOOLER against BERT fine-tuned on the MR dataset. Each $\mathbf{x}$ is classified as positive, and each $\mathbf{x}_{adv}$ is classified as negative.

$\epsilon_{sem} = 4$ as a general rule: on average, humans should at least "Agree" that $\mathbf{x}$ and $\mathbf{x}_{adv}$ have the same meaning.

4.1.2 Automatic Evaluation

Automatic evaluation of semantic similarity is a well-studied NLP task. The STS Benchmark is used as a common measurement (Cer et al., 2017).

Michel et al. (2019) explored the use of common evaluation metrics for machine translation as a proxy for semantic similarity in the attack setting. While n-gram overlap based approaches are computationally cheap and work well in the machine translation setting, they do not correlate with human judgment as well as sentence encoders (Wieting and Gimpel, 2018).

Some attacks have used sentence encoders to encode two sentences into a pair of fixed-length vectors, then used the cosine distance between the vectors as a proxy for semantic similarity. TEXTFOOLER uses the Universal Sentence Encoder (USE), which achieved a Pearson correlation score of 0.782 on the STS benchmark (Cer et al., 2018). Another option is BERT fine-tuned for semantic similarity, which achieved a score of 0.865 (Devlin et al., 2018).

Additionally, synonym substitution methods, including TEXTFOOLER and GENETICATTACK, often require that words be substituted only with neighbors in the counter-fitted embedding space, which is designed to push synonyms together and antonyms apart (Mrksic et al., 2016). These automatic metrics of similarity produce a score that represents the similarity between $\mathbf{x}$ and $\mathbf{x}_{adv}$. Attacks depend on a minimum threshold value for each metric to determine whether the changes between $\mathbf{x}$ and $\mathbf{x}_{adv}$ preserve semantics. Human evaluation is needed to find threshold values such that people generally "agree" that semantics is preserved.

4.1.3 Case Study

To quantify semantic similarity of $\mathbf{x}$ and x_{adv}, we asked users whether they agreed that the changes between the two passages preserved meaning on a scale of 1 (Strongly Disagree) to 5 (Strongly Agree). We averaged scores for each attack method to determine if the method generally preserves semantics.

Perturbations generated by TEXTFOOLER were rated an average of **3.28**, while perturbations generated by GENETICATTACK were rated on average **2.70**.[8] The average rating given for both methods was significantly less than our proposed ϵ_{sem} of 4. Using a clear survey question illustrates that humans, on average, don't assess these perturbations as semantics-preserving.

[8]We hypothesize that TEXTFOOLER achieved higher scores due to its use of USE.

4.2 Evaluation of Grammaticality

4.2.1 Human Evaluation

Both Jin et al. (2019) and Iyyer et al. (2018) reported a human evaluation of grammaticality, but neither study clearly asked if any errors were introduced by a perturbation. For human evaluation of the grammaticality constraint, we propose presenting $\mathbf{x}$ and $\mathbf{x}_{adv}$ together and asking judges if grammatical errors were introduced by the changes made. However, due to the rule-based nature of grammar, automatic evaluation is preferred.

4.2.2 Automatic Evaluation

The simplest way to automatically evaluate grammatical correctness is with a rule-based grammar checker. Free grammar checkers are available online in many languages. One popular checker is LanguageTool, an open-source proofreading tool (Naber, 2003). LanguageTool ships with thousands of human-curated rules for the English language and provides an interface for identifying grammatical errors in sentences. LanguageTool uses rules to detect grammatical errors, statistics to detect uncommon sequences of words, and language model perplexity to detect commonly confused words.

4.2.3 Case Study

We ran each of the generated $(\mathbf{x}, \mathbf{x}_{adv})$ pairs through LanguageTool to count grammatical errors. LanguageTool detected more grammatical errors in $\mathbf{x}_{adv}$ than $\mathbf{x}$ for **50%** of perturbations generated by TEXTFOOLER, and **32%** of perturbations generated by GENETICATTACK.

Additionally, perturbations often contain errors that humans rarely make. LanguageTool detected 6 categories for which errors in the perturbed samples appear at least 10 times more frequently than in the original content. Details regarding these error categories and examples of violations are shown in Table 4.

4.3 Evaluation of Non-suspicion

4.3.1 Human Evaluation

We propose evaluation of non-suspicion by having judges view a shuffled mix of real and adversarial inputs and guess whether each is real or computer-altered. This is similar to the human evaluation done by Ren et al. (2019), but we formulate it as a binary classification task rather than on a 1-5 scale. A perturbed example $\mathbf{x}_{adv}$ is not suspicious if the percentage of judges who identify $\mathbf{x}_{adv}$ as computer-altered is at most ϵ_{ns}, where $0 \leq \epsilon_{ns} \leq 1$.

4.3.2 Automatic Evaluation

Automatic evaluation may be used to guess whether or not an adversarial example is suspicious. Models can be trained to classify passages as real or perturbed, just as human judges do. For example, Warstadt et al. (2018) trained sentence encoders on a real/fake task as a proxy for evaluation of linguistic acceptability. Recently, Zellers et al. (2019) demonstrated that GROVER, a transformer-based text generation model, could classify its own generated news articles as human or machine-written with high accuracy.

4.3.3 Case Study

We presented a shuffled mix of real and perturbed examples to human judges and asked if they were real or computer-altered. As this is a time-consuming task for long documents, we only evaluated adversarial examples generated by TEXTFOOLER on the sentence-level MR dataset.

If all generated examples were non-suspicious, judges would average 50% accuracy, as they would not be able to distinguish between real and perturbed examples. In this case, judges achieved **69.2%** accuracy.

5 Producing Higher Quality Adversarial Examples

In Section 4, we evaluated how well generated examples met constraints. We found that although attacks in NLP aspire to meet linguistic constraints, in practice, they frequently violate them. Now, we adjust automatic constraints applied during the course of the attack to produce better quality adversarial examples.

We set out to find if a set of constraint application methods with appropriate thresholds could produce adversarial examples that are semantics-preserving, grammatical and non-suspicious. We modified TEXTFOOLER to produce TFADJUSTED, a new attack with stricter constraint application. To enforce grammaticality, we added LanguageTool. To enforce semantic preservation, we tuned two thresholds which filter out invalid word substitutions: (a) minimum cosine similarity between counter-fitted word embeddings and (b) minimum

Grammar Rule ID	x	x_{adv}	Explanation	Context
TO_NON_BASE	2	123	Did you mean "know"? —— Replace with one of [know]	...ees at person they don't really want to **knew**
PRP_VBG	3	112	Did you mean "we're wanting", "we are wanting", or "we were wanting"? —— Replace with one of [we're wanting,we are wanting,we were wanting]	while **we wanting** macdowell's character to retrieve her h...
A_PLURAL	20	294	Don't use indefinite articles with plural words. Did you mean "a grate", "a gratis" or simply "grates"? —— Replace with one of [a grate,a gratis,grates]	a **grates**, lanky flick
DID_BASEFORM	25	328	The verb 'can't' requires base form of this verb: "compare" —— Replace with one of [compare]	...first two cinema in the series, i can't **compares** friday after next to them, but nothing ...
PRP_VB	6	73	Do not use a noun immediately after the pronoun 'it'. Use a verb or an adverb, or possibly some other part of speech. —— Replace game with one of []	...ble of being gravest, so thick with wry it **game** like a readings from bartlett's familia...
PRP_MD_NN	4	46	It seems that a verb or adverb has been misspelled or is missing here. —— Replace with one of [can be appreciative,can have appreciative]	...y bit as awful as borchardt's coven, we can **appreciative** it anyway
NON3PRS_VERB	7	78	The pronoun 'they' must be used with a non-third-person form of a verb: "do" —— Replace with one of [do]	they **does** a ok operating of painting this family ...

Table 4: **Adversarial Examples Contain Uncommon Grammatical Errors.** This table shows grammatical errors detected by LanguageTool that appeared far more often in the perturbed samples. $\mathbf{x}$ and $\mathbf{x}_{adv}$ denote the numbers of errors detected in $\mathbf{x}$ and $\mathbf{x}_{adv}$ across 3,115 examples generated by TEXTFOOLER and GENETICATTACK.

cosine similarity between sentence embeddings. Through human studies, we found threshold values of **0.9** for (a) and **0.98** for (b)[9]. We implemented TFADJUSTED using TextAttack, a Python framework for implementing adversarial attacks in NLP (Morris et al., 2020).

5.1 With Adjusted Constraint Application

We tested TFADJUSTED to determine the effect of tightening constraint application. We used the IMDB, Yelp, and MR datasets for classifcation as in Section 4. We added the SNLI and MNLI entailment datasets (Bowman et al., 2015; Williams et al., 2018) for the portions not requring human evaluation. Table 5 shows the results.

Semantics. TEXTFOOLER generates perturbations for which human judges are on average "Not sure" if semantics are preserved. With perturbations generated by TFADJUSTED, human judges on average "Agree" that semantics are preserved.

Grammaticality. Since all examples produced by TFADJUSTED are checked with LanguageTool, no perturbation can introduce grammatical errors. [10]

Non-suspicion. We repeated the non-suspicion study from Section 4.3 with the examples generated by TFADJUSTED. Participants were able to guess with 58.8% accuracy whether inputs were computer-altered. The accuracy is over 10% lower than the accuracy on the examples generated by TEXTFOOLER.

Attack success. For each of the three datasets, the attack success rate decreased by at least 71 percentage points (see last row of Table 5).

5.2 Adversarial Training With Higher Quality Examples

Using the 9,595 samples in the MR training set as seed inputs, TEXTFOOLER generated **7,382** adversarial examples, while TFADJUSTED generated just **825**. We append each set of adversarial examples to a copy of the original MR training set and fine-tuned a pre-trained BERT model for 10 epochs. Figure 2 plots the test accuracy over 10 training epochs, averaged over 5 random seeds per dataset. While neither training method strongly impacts accuracy, the augmentation using TFADJUSTED has a better impact than that of TEXTFOOLER.

We then re-ran the two attacks using 1000 examples from the MR test set as seeds. Again averaging over 5 random seeds, we found no significant change in robustness. That is, models trained on the original MR dataset were approximately as robust as those trained on the datasets augmented with TEXTFOOLER and TFADJUSTED examples. This corroborates the findings of Alzantot et al. (2018) and contradicts those of Jin et al. (2019). We include further analysis along with some hypotheses for the discrepancies in adversarial training results in A.4.

[9] Details in the appendix, Section A.2.2.

[10] Since the MR dataset is already lowercased and tokenized, it is difficult for a rule-based grammar checker like LanguageTool to parse some inputs.

Datasets ⟶	IMDB	Yelp	MR	SNLI	MNLI	Note
Semantic Preservation (before)	3.41	3.05	3.37	–	–	
Semantic Preservation (after)	4.06	3.94	4.18	–	–	Higher value: more preserved
Grammatical Error % (before)	52.8	61.2	28.3	26.7	20.1	
Grammatical Error % (after)	0	0	0	0	0	Lower value: less mistakes
Non-suspicion % (before)	–	–	69.2	–	–	
Non-suspicion % (after)	–	–	58.8	–	–	Lower value: less suspicious
Attack Success % (before)	85.0	93.2	86.6	94.5	95.1	
Attack Success % (after)	13.9	5.3	10.6	7.2	14.8	
Difference (before - after)	**71.1**	**87.9**	**76.0**	**87.3**	**80.3**	

Table 5: Results from running TEXTFOOLER (before) and TFADJUSTED (after). Attacks are on BERT classification models fine-tuned for five respective NLP datasets.

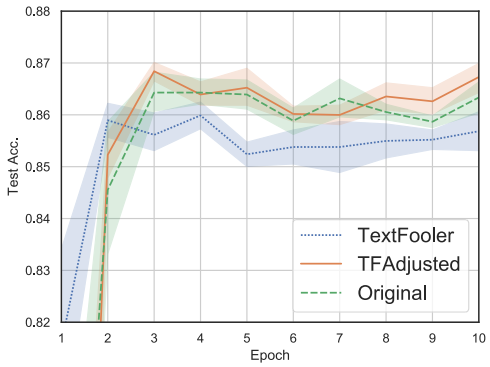

Figure 2: Accuracy of adversarially trained models on the MR test set. Augmentation with adversarial examples generated by TEXTFOOLER (blue), although higher in quantity, decreases the overall test accuracy while examples generated by TFADJUSTED (orange) have a small positive effect.

Constraint Removed	Yelp	IMDB	MR	MNLI	SNLI
(Original - all used)	5.3	13.9	10.6	14.3	7.2
Sentence Encoding	22.9	45.0	28.7	44.4	31.2
Word Embedding	**74.6**	**87.1**	**52.9**	**82.7**	**69.8**
Grammar Checking	5.8	15.0	11.6	15.4	9.0

Table 6: Ablation study: effect of removal of a single constraint on TFADJUSTED attack success rate. Attacks against BERT fine-tuned on each dataset.

5.3 Ablation of TFADJUSTED Constraints

TFADJUSTED generated better quality adversarial examples by constraining its search to exclude examples that fail to meet three constraints: word embedding distance, sentence encoder similarity, and grammaticality. We performed an ablation study to understand the relative impact of each on attack success rate.

We reran three TFADJUSTED attacks (one for each constraint removed) on each dataset. Table 6 shows attack success rate after individually removing each constraint. The word embedding distance constraint was the greatest inhibitor of attack success rate, followed by the sentence encoder.

6 Comparing Search Methods

When an attack's success rate improves, it may be the result of either (a) improvement of the search method for finding adversarial perturbations or (b) more lenient constraint definitions or constraint application. TEXTFOOLER achieves a higher success rate than GENETICATTACK, but Jin et al. (2019) did not identify whether the improvement was due to (a) or (b). Since TEXTFOOLER uses both a different search method and different constraint application methods than GENETICATTACK, the source of the difference in attack success rates is unclear.

To determine which search method is more effective, we used TextAttack to compose attacks from the search method of GENETICATTACK and the constraint application methods of each of TEXTFOOLER and TFADJUSTED (Morris et al., 2020). With the constraint application held constant, we can identify the source of the difference in attack success rate. Table 7 reveals that the genetic algorithm of GENETICATTACK is more successful than the greedy search of TEXTFOOLER at both constraint application levels. This reveals the source of improvement in attack success rate between GENETICATTACK and TEXTFOOLER to be more lenient constraint application. However, GENETICATTACK's genetic algorithm is far more computationally expensive, requiring over 40x more model queries.

7 Discussion

Tradeoff between attack success and example quality. TFADJUSTED made semantic constraints more selective, which helped attacks generate examples that scored above 4 on the Likert scale for preservation of semantics. However, this led to a steep drop in attack success rate. This indicates that, when only allowing adversarial perturbations

Constraints	TFADJUSTED		TEXTFOOLER	
Search Method	TEXTFOOLER	GENETICATTACK	TEXTFOOLER	GENETICATTACK
Semantic Preservation	4.06	4.11	-	-
Grammatical Error %	0	0	-	-
Non-suspicion Score	58.8	56.9	-	-
Attack Success %	10.6	**12.0**	91.1	**95.0**
Perturbed Word %	11.1	11.0	18.9	17.2
Num Queries	**27.1**	4431.6	**77.0**	3225.7

Table 7: Comparison of the search methods from GENETICATTACK and TEXTFOOLER with two sets of constraints (TEXTFOOLER and TFADJUSTED). Attacks were run on 1000 samples against BERT fine-tuned on the MR dataset. GENETICATTACK's genetic algorithm is more successful than TEXTFOOLER's greedy strategy, albeit much less efficient.

that preserve semantics and grammaticality, NLP models are relatively robust to current synonym substitution attacks. Note that our set of constraints isn't necessarily optimal for every attack scenario. Some contexts may require fewer constraints or less strict constraint application.

Decoupling search methods and constraints. It is critical that researchers decouple new search methods from new constraint evaluation and constraint application methods. Demonstrating the performance of a new attack that simultaneously introduces a new search method and new constraints makes it unclear whether empirical gains indicate a more effective attack or a more relaxed set of constraints. This mirrors a broader trend in machine learning where researchers report differences that come from changing multiple independent variables, making the sources of empirical gains unclear (Lipton and Steinhardt, 2018). This is especially relevant in adversarial NLP, where each experiment depends on many parameters.

Towards improved methods for generating textual adversarial examples. As models improve at paraphrasing inputs, we will be able to explore the space of adversarial examples beyond synonym substitutions. As models improve at measuring semantic similarity, we will be able to more rigorously ensure that adversarial perturbations preserve semantics. It remains to be seen how robust BERT is when subject to paraphrase attacks that rigorously preserve semantics and grammaticality.

8 Related Work

The goal of creating adversarial examples that preserve semantics and grammaticality is common in the NLP attack literature (Zhang et al., 2019). However, previous works use different definitions of adversarial examples, making it difficult to compare methods. We provide a unified definition of an adversarial example based on a goal function and a set of linguistic constraints.

Gilmer et al. (2018) laid out a set of potential constraints for the attack space when generating adversarial examples, which are each useful in different real-world scenarios. However, they did not discuss NLP attacks in particular. Michel et al. (2019) defined a framework for evaluating attacks on machine translation models, focusing on meaning preservation constraints, but restricted their definitions to sequence-to-sequence models. Other research on NLP attacks has suggested various constraints but has not introduced a shared vocabulary and categorization that allows for effective comparisons between attacks.

9 Conclusion

We showed that two state-of-the-art synonym substitution attacks, TEXTFOOLER and GENETICATTACK, frequently violate the constraints they claim to follow. We created TFADJUSTED, which applies constraints that produce adversarial examples judged to preserve semantics and grammaticality.

Due to the lack of a shared vocabulary for discussing NLP attacks, the source of improvement in attack success rate between TEXTFOOLER and GENETICATTACK was unclear. Holding constraint application constant revealed that the source of TEXTFOOLER's improvement was lenient constraint application (rather than a better search method). With a shared framework for defining and applying constraints, future research can focus on developing better search methods and better constraint application techniques for preserving semantics and grammaticality.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

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

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *ArXiv*, abs/1803.11175.

Minhao Cheng, Jinfeng Yi, Huan Zhang, Pin-Yu Chen, and Cho-Jui Hsieh. 2018. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

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

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *IEEE Security and Privacy Workshops (SPW)*.

Justin Gilmer, Ryan P. Adams, Ian J. Goodfellow, David Andersen, and George E. Dahl. 2018. Motivating the rules of the game for adversarial example research. *CoRR*, abs/1807.06732.

Ian Goodfellow, Nicolas Papernot, Sandy Huang, Rocky Duan, Pieter Abbeel, and Jack Clark. Attacking machine learning with adversarial examples [online]. 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

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. *ArXiv*, abs/1909.01492.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *CoRR*, abs/1804.06059.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. *arXiv e-prints*, page arXiv:1907.11932.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems.

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

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

R. Likert. 1932. *A Technique for the Measurement of Attitudes*. Number nos. 136-165 in A Technique for the Measurement of Attitudes. publisher not identified.

Zachary Chase Lipton and Jacob Steinhardt. 2018. Troubling trends in machine learning scholarship. *ArXiv*, abs/1807.03341.

Paul Michel, Xian Li, Graham Neubig, and Juan Miguel Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. *CoRR*, abs/1903.06620.

John X. Morris, Eli Lifland, Jin Yong Yoo, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks in natural language processing.

Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Maria Rojas-Barahona, Pei hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. 2016. Counter-fitting word vectors to linguistic constraints. In *HLT-NAACL*.

Daniel Naber. 2003. A rule-based style and grammar checker.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association*

for Computational Linguistics (ACL'05), pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 49–54. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2018. Neural network acceptability judgments. *CoRR*, abs/1805.12471.

John Wieting and Kevin Gimpel. 2018. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

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

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *CoRR*, abs/1905.12616.

Wei Emma Zhang, Quan Z. Sheng, and Ahoud Abdulrahmn F. Alhazmi. 2019. Generating textual adversarial examples for deep learning models: A survey. *CoRR*, abs/1901.06796.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.