BERT is Robust! A Case Against Synonym-Based Adversarial Examples in Text Classification

Anonymous ACL submission

Abstract

In this work, we investigate the robustness of BERT using four word substitution-based attacks. We combine a human evaluation of individual word substitutions and a probabilistic analysis to show that between 96% and 99% of the analyzed attacks do not preserve semantics, indicating that their success is mainly based on feeding poor data to the model. To further confirm that, we introduce an efficient data augmentation procedure and show that many successful attacks can be prevented by including data similar to adversarial examples during training. Compared to traditional adversarial training, our data augmentation procedure requires 30× less computation time per epoch, while achieving better performance on two out of three datasets. We introduce an additional post-processing step that reduces the success rates of state-of-the-art attacks below 4%, 5%, and 8% on the three considered datasets. Finally, by looking at constraints for word substitutions that better preserve the semantics, we conclude that BERT is considerably more robust than previous research suggests.

1 Introduction

Recent research in computer vision (Szegedy et al., 2014; Goodfellow et al., 2015) and speech recognition (Carlini and Wagner, 2018) has shown that neural networks are vulnerable to changes that are invisible to humans. This means that it is possible to imperceptibly modify a certain sample, e.g., an image, such that the neural network changes its prediction. These modified examples are called *adversarial examples*, and the process of generating them is often referred to as *attacking* a neural network. Following the outstanding success of adversarial examples in computer vision (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018; Carlini and Wagner, 2017), a considerable research effort has been dedicated to studying adversarial attacks in Natural Language Processing (NLP) (Papernot et al., 2016; Alzantot et al., 2018; Zhang et al., 2019; Ren et al., 2019; Jin et al., 2020; Li et al., 2020; Garg and Ramakrishnan, 2020).

However, since natural language tokens are non-differentiable, finding adversarial examples that are truly imperceptible to humans is extremely challenging in NLP. For research on textual adversarial attacks to be reliable, the generated examples must preserve the semantic meaning of the original examples, which is often neglected in current research. Therefore, we observe that as the effectiveness of the existing attacks increases, the line between adversarial examples and nonsensical text becomes blurry.

In this work, we show that despite the general consensus that textual adversarial attacks should preserve semantics (Morris et al., 2020a; Ren et al., 2019; Jin et al., 2020; Li et al., 2020; Garg and Ramakrishnan, 2020), current attacks are designed to optimize certain metrics, such as success rate, and neglect the importance of semantic preservation. We combine a human evaluation with a simple probabilistic analysis to show that between 96% and 99% of the adversarial examples on BERT (Devlin et al., 2019) created by four different state-of-the-art attack methods do not preserve semantics. Additionally, we propose a two-step procedure consisting of data augmentation and post-processing for defending against adversarial examples.[1] Our results show that we can eliminate up to two-thirds of the successful attacks by simply including data similar to the adversarial examples. Further, we can revert between 70% and 92% of the remaining adversarial examples using a post-processing step that consists of deciding by majority voting from several noisy versions of the input example. Compared to adversarial training strategies, our method results in a speedup of almost 30× per training epoch while achieving better robustness on two of the three considered datasets without losing classi-

[1] We will release the code with the publication of this work.

fication performance.

2 Related Work

Papernot et al. (2016) were the first to introduce adversarial examples in text. In the following years, a number of different attacks were proposed. Alzantot et al. (2018) use a population-based optimization algorithm for creating adversarial examples, while Zhang et al. (2019) use Metropolis-Hastings (Metropolis et al., 1953; Hastings, 1970). Further word substitution based attacks were proposed by Ren et al. (2019); Jin et al. (2020); Li et al. (2020) and Garg and Ramakrishnan (2020), which we discuss in more detail in Section 3.1.

Regarding adversarial defense, some studies that introduced attacks also incorporated the created adversarial examples during training (Alzantot et al., 2018; Ren et al., 2019). However, due to the high cost of running these attacks, they cannot create sufficiently many adversarial examples, achieving only minor improvements in robustness. Wang et al. (2021a) present the Synonym Encoding Method (SEM), a method that uses an encoder that maps clusters of synonyms to the same embedding. Although this method works well, it also limits the expressive capacity of the network. Wang et al. (2021b) propose a method for fast adversarial training called Fast Gradient Projection Method (FGPM) that is limited to models with non-contextual word vectors as input. On BERT, Meng et al. (2021) use a geometric attack that allows for creating adversarial examples in parallel and therefore leads to faster adversarial training. Another line of work is around certified robustness through Interval Bound Propagation (Jia et al., 2019; Huang et al., 2019); unfortunately, these approaches currently do not scale to large models and datasets.

There is little work analyzing in-depth or questioning current synonym-based adversarial attacks in NLP. Among those, Morris et al. (2020a) find that adversarial attacks often do not preserve semantics using a human evaluation. We extend this line of work by providing a probabilistic analysis that shows that adversarial examples do not preserve semantics according to human judgment.

3 Background

For a classifier $f : \mathcal{S} \to \mathcal{Y}$ and some correctly classified input $s \in \mathcal{S}$, an adversarial example is an input $s' \in \mathcal{S}$, such that $f(s) \neq f(s')$, and $sim(s, s') \geq t_{sim}$, where $sim(s, s') \geq t_{sim}$ is a constraint on the similarity of s and s'. For text classification, $s = \{w_1, w_2, ..., w_n\}$ is a sequence of words. Common notions of similarity are the cosine similarity of counter-fitted[2] word vectors (Mrkšić et al., 2016), which we will denote as $cos_{cv}(w_i, w_i')$, or the cosine similarity of sentence embeddings from the Universal Sentence Encoder (USE) (Cer et al., 2018), which we will denote as $cos_{use}(s, s')$. Note that this is a slight abuse of notation since s and s' are just sequences of words. This notation should be interpreted as follows: we first apply USE to s and s' to get two sentence vectors and then calculate the cosine similarity. The same holds for $cos_{cv}(w_i, w_i')$, where we first obtain the counter-fitted word vectors of w_i and w_i'. Also, note that whenever we talk about the *cosine similarity of words*, it refers to the cosine similarity of words in the counter-fitted embedding. Similarly, *USE score* refers to the cosine similarity of sentence embeddings from the USE.

3.1 Attacks

We consider four different attacks in our experiments, which exchange words from the input sequence with other words of similar meaning from a *candidate set*.

TextFooler Jin et al. (2020) propose TextFooler, which builds its candidate set from the 50 nearest neighbors in a vector space of counter-fitted word embeddings. The constraints are $cos_{cv}(w_i, w_i') \geq 0.5 \ \forall i$ and $cos_{use}(s, s') \geq 0.878$.[3]

Probability Weighted Word Saliency (PWWS) Ren et al. (2019) use WordNet[4] synonyms to construct a candidate set. This method uses no additional constraints.

BERT-Attack Li et al. (2020) present an attack based on BERT itself. BERT-Attack uses a BERT Masked-Language Model (MLM) that proposes 48 possible replacements to form the candidate set. The constraints are: $cos_{use}(s, s') \geq 0.2$, and a maximum of 40% of all words can be replaced.

BAE Garg and Ramakrishnan (2020) propose another attack based on a BERT MLM. BAE uses the

[2]Counter-fitting is a procedure that injects antonym and synonym constraints into static word embeddings.

[3]The original value is 0.841 on the angular similarity between sentence embeddings, which corresponds to a cosine similarity of 0.878.

[4]https://wordnet.princeton.edu/

Dataset	Attack Success Rate (%)			
	TextFooler	PWWS	BERT-Attack	BAE
AG News	84.99	64.95	79.43	14.27
Yelp	90.47	92.23	93.47	31.50
IMDB	98.16	98.70	99.03	57.13

Table 1: Attack success rates of the different attacks on fine-tuned BERT-base-uncased models.

top 50 candidates of the model to build the candidate set and tries to enforce semantic similarity by requiring $cos_{use}(s, s') \geq 0.936$.

4 Setup

We use the BERT-base-uncased model provided by the HuggingFace Transformers library (Wolf et al., 2019) for all our experiments and rely on the TextAttack library (Morris et al., 2020b) for the implementations of the different attacks.

We fine-tune BERT for two epochs on AG News, Yelp,[5] and IMDB. To evaluate the attacks, we randomly sample 1000 examples from each test-set for running the attacks. The clean accuracies of our models are 94.57% on AG News, 97.31% on Yelp, and 93.77% on IMDB. The *attack success rates*, defined as the percentage of attack attempts that produce adversarial examples, for the different attacks are shown in Table 1. It is worth noting that the average sequence length on IMDB is 279, compared to 44 and 46 on AG News and Yelp, which makes IMDB easier to attack (see Appendix E).

Further, it is interesting that BAE, which requires a much higher sentence similarity than BERT-Attack, is considerably less effective despite being otherwise similar. However, is a high sentence similarity sufficient to ensure semantic similarity? This is a part of what we investigate using a human evaluation.

5 Quality of Adversarial Examples

To investigate the quality of adversarial examples, we conduct a human evaluation on the word substitutions performed by the different attacks. In the following, we call a word substitution a *perturbation*. Then, we perform a probabilistic analysis to generalize the results on individual perturbations to attacks, which usually consist of multiple perturbations.

[5]We restricted ourselves to examples in Yelp which have fewer than 80 words to save computing resources.

5.1 Human Evaluation

For the human evaluation, we rely on labor crowd-sourced from Amazon Mechanical Turk.[6] We collect 100 pairs of [*original word*, *attack word*] for every attack and another 100 pairs for every attack where the context is included with a window size of 11. For the word-pairs, inspired by Morris et al. (2020a), we asked the workers to react to the following claim: *"In general, replacing the first word with the second word preserves the meaning of the sentence."* For the words with context, we presented the two text fragments on top of each other, highlighted the changed word, and asked the workers: *"In general, the change preserves the meaning of the text fragment."* In both cases the workers had seven answers to choose from: "Strongly Disagree", "Disagree", "Somewhat Disagree", "Neutral", "Somewhat Agree", "Agree", "Strongly Agree". We convert these answers to a scale from 1-7, where higher is better. Finally, to measure voter agreement, we calculate the average number of workers who voted within ± 1 of the mean score for a perturbation. Screenshots and more details about the two evaluations can be found in Appendix F.

Table 2 shows the results of this human analysis. Our evaluation shows that humans generally tend to disagree that the newly introduced word preserves the meaning. This holds for all attacks, and regardless of whether we show the word with or without context. Critically, in our human evaluation, we display the words and passages that are changed and ask the evaluators to assess exclusively these pieces of text. Conversely, human studies asking whether two long text documents that differ only on a few words are similarJin et al. (2020); Li et al. (2020), are likely to obtain a higher agreement since the evaluators will hardly consider the details closely enough.

Regarding the different attacks, it becomes clear from the results in Table 2 that building a candidate set from the first 48 or 50 candidates proposed by a language model (as in BERT-Attack and BAE) does not work without an additional constraint on the word similarity. The results on BAE further make it clear that a high sentence similarity according to the USE score is no guarantee for semantic similarity. PWWS and TextFooler receive similar scores for word similarity, but the drop in score for PWWS when going from word similarity to text similarity

[6]https://www.mturk.com/

Attack	Word Similarity			Text Similarity		
	Avg. (1-7)	Above 5 (%)	Above 6 (%)	Avg. (1-7)	Above 5 (%)	Above 6 (%)
TextFooler	**3.88**	**22**	**7**	**3.47**	**24**	**12**
PWWS	3.83	21	6	2.70	13	6
BERT-Attack	2.27	4	4	2.55	7	3
BAE	1.64	0	0	1.85	3	2

Table 2: Average human scores on a scale from 1-7 and the percentage of scores above 5 and 6 (corresponding to the answers "Somewhat Agree" and "Agree") for the different attacks and when the words were shown with (text similarity) or without (word similarity) context.

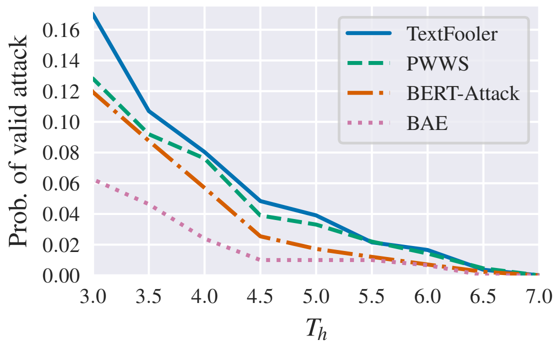


Figure 1: Probability that an attack is valid according to our probabilistic analysis, for the different attacks and for different thresholds T_h.

indicates that while the synonyms retrieved from WordNet are usually related to the original word, the relation is often wrong in the given context. TextFooler receives the highest scores in this analysis, but even for TextFooler, just 22% and 24% of the perturbations were rated above 5, which corresponds to "Somewhat Agree".

The voter agreement on these results is 3.57 out of 5 for the words with context and 6.78 out of 10 for the words without context.

5.2 Probabilistic Estimation of Valid Attacks

Our human evaluation is based on individual perturbations. However, an attack usually changes multiple words. Therefore, to understand how many of the successful attacks are valid attacks, we need to define *valid perturbations* and *valid attacks*.

Definition 5.1 (Valid Perturbation)**.** A *valid perturbation* is a perturbation that receives a human score above some threshold T_h.

Definition 5.2 (Valid Attack)**.** A *valid attack* is an attack consisting of valid perturbations only.

Sensible values for T_h are in the range 5-6, which corresponds to "Somewhat Agree" to "Agree". In order to get an estimate for the percentage of valid attacks, we perform a simple probabilistic analysis. Let A_{val}, P_{val} and A^i_{val} denote the events of a valid attack, a valid perturbation and a valid attack given that there are exactly i perturbations, respectively. Further, let $p(i)$ denote the probability that an attack perturbs i words. Using this notation, we can approximate the probability that a successful attack is valid as

$$\begin{aligned} p(A_{val}) &= \sum_{i=1}^{N} p(i)p(A^i_{val}) \\ &\approx \sum_{i=1}^{N} p(i)p(P_{val})^i, \end{aligned} \tag{1}$$

where N is the maximum number of allowed perturbations. With the data from our human evaluation and the collected adversarial examples, we can obtain an unbiased estimate for this probability as

$$\hat{p}(A_{val}) = \sum_{i=1}^{N} \hat{p}(i) \left(\frac{\text{count}[S_h \geq T_h]}{n_{pert}} \right)^i, \tag{2}$$

where S_h is the average score of the workers for a perturbation, n_{pert} is the total number of perturbations analyzed by the workers for any given attack, and $\hat{p}(i)$ can be estimated using counts.

The results of this analysis are shown in Figure 1 as a function of the threshold T_h. It can be seen that if we require an average score of 5 for all perturbations, we can expect around 4% of the successful attacks from TextFooler to be valid, slightly less for PWWS, below 2% for BERT-Attack, and just around 1% for BAE. In other words, between 96% and 99% of the successful attacks can not be considered valid according to the widely accepted requirement that adversarial examples should preserve semantics.

This analysis assumes that perturbations are independent of each other, which is not true because every perturbation impacts the following perturbations. Nevertheless, we argue that this approximation tends to result in optimistic estimates on the true number of valid attacks for the following reasons: 1) When an attack is already almost successful, all attacks except for PWWS try to maximize sentence similarity on the last perturbation, making the last perturbation generally weaker. 2) We assume that in a sentence with multiple changes, a human is generally less likely to say that the meaning is preserved, even if the individual perturbations are considered valid.

6 Adversarial Defense

We have shown that current attacks use lenient constraints and, therefore, mostly produce adversarial examples that should not be considered valid, but finding suitable thresholds on the constraints is difficult. Before discussing realistic thresholds, we show that we can defend against a large proportion of adversarial examples even for permissive constraints (in terms of the validity of the perturbations).

Our defense consists of a gradient-based data augmentation procedure followed by a post-processing step.

Data Augmentation

1. Initialize thresholds $t_{rr} \in (0, 100]$, which corresponds to the maximum percentage of words to augment in an input sequence, and $t_{cv} \in (0, 1)$, which represents the minimum cosine similarity between the original and the perturbed word.

2. During training, for every input s in a batch, the importance I of a word w consisting of vectors $\mathbf{v}_j \in \mathbb{R}^{768}$ in BERT's initial embedding is estimated as

$$I_w = \sum_{\mathbf{v}_j \in w} \mathbf{v}_j \cdot \nabla_{\mathbf{v}_j} L(\boldsymbol{\theta}, s, y), \qquad (3)$$

   where $\boldsymbol{\theta}$ are the parameters of BERT, L is the loss function and y is the label. Using this importance metric, the t_{rr} percent of most important words is marked; and the union of the words considered as stop-words by the four attacks is filtered out.

3. Then, for each word marked as important according to (2), a candidate set $\mathcal{C} = \{w'_1, ..., w'_n\}$ is built with the 50 nearest neighbors in the counter-fitted embedding space, which also present a cosine similarity greater than t_{cv}. To account for the fact that all attacks tend to favor words with low cosine similarity, the replacement $w'_i \in \mathcal{C}$ for the original word w is chosen from the candidate set with probability:

$$p(w'_i) = \frac{1 - cos_{cv}(w, w'_i)}{\sum_{w'_j \in \mathcal{C}} 1 - cos_{cv}(w, w'_j)}. \qquad (4)$$

   The augmented batch is then appended to the original batch, increasing the batch size by a factor of two.

This data augmentation procedure makes the model more robust against attack words with cosine similarity greater than t_{cv}. If we expect BERT to be robust against these kinds of replacements, this is the least we should do. Otherwise, we cannot expect the model to generalize to the attack's input space, which is significantly larger than the input space during fine-tuning.

The second step of our defense is a post-processing step based on ensembling. This step builds on the robustness to random substitutions obtained from data augmentation.

Post-processing

1. For every text that should be classified, N versions are created. In each version, t_{rr} percent of the words (which are not stop-words) are selected uniformly at random. Then, as in the data augmentation step, each of these words w_i is exchanged by another uniformly sampled word from a candidate set $\mathcal{C}$ consisting of the 50 nearest neighbors with cosine-similarity above t_{cv} with respect to w_i.

2. Finally, the output logits are added up for the N versions and the final prediction is made according to the maximum value. Formally, let $l_j(s)$ denote the value of the j-th logit for some input s; the prediction y_{pred} is made according to

$$y_{pred} = \arg\max_j \sum_{i=1}^{N} l_j(s_i). \qquad (5)$$

7 Defense Evaluation

First, we apply to all attacks the constraint $cos_{cv}(w_i, w'_i) \geq 0.5 \ \forall i$ and run the attacks on the following configurations: a model trained normally (N); a model trained using our data augmentation

Dataset	Method	Clean Acc. (%)	Attack Success Rate (%)			
			TextFooler	PWWS$_{cv50}$	BERT-Attack$_{cv50}$	BAE$_{cv50}$
AG News	N	94.57	84.99	16.38	20.72	0.32
	DA	**94.82**	52.37	10.73	18.61	–
	DA+PP	93.84 ± 0.07	**3.93** ± 0.41	**2.55** ± 0.31	**3.73** ± 0.29	–
	DA+MA$_5$	93.72 ± 0.12	14.11 ± 0.48	4.61 ± 0.41	7.52 ± 0.48	–
	N+PP	87.89 ± 0.16	10.32 ± 0.48	5.0 ± 0.31	5.59 ± 0.36	–
Yelp	N	**97.31**	90.47	33.26	49.53	0.41
	DA	97.10	29.79	10.52	16.49	–
	DA+PP	96.59 ± 0.06	**4.37** ± 0.39	**2.54** ± 0.15	**4.86** ± 0.33	–
	DA+MA$_5$	95.40 ± 0.10	10.23 ± 0.59	4.62 ± 0.36	7.38 ± 0.38	–
	N+PP	94.50 ± 0.08	6.07 ± 0.47	5.22 ± 0.48	7.35 ± 0.61	–
IMDB	N	93.77	98.16	65.77	88.44	3.07
	DA	**94.21**	48.31	29.49	40.91	–
	DA+PP	92.59 ± 0.06	**5.81** ± 0.45	**4.53** ± 0.26	**7.83** ± 0.37	–
	DA+MA$_5$	92.49 ± 0.12	12.05 ± 0.87	8.36 ± 0.36	13.0 ± 0.64	–
	N+PP	88.35 ± 0.09	10.52 ± 0.46	9.3 ± 0.39	13.3 ± 0.55	–

Table 3: Effectiveness of defense procedure for different attacks modified with the constraint $cos_{cv}(w_i, w_i') \geq 0.5\ \forall i$.

procedure (*DA*); and a model trained with data augmentation that uses our post-processing method (*DA+PP*). Additionally, we provide a baseline for our post-processing procedure by masking 5% of all tokens with the `[MASK]` token (*DA+MA*$_5$; for details see Appendix B). Furthermore, we show the impact of applying the post-processing step without data augmentation (*N+PP*). Given that the post-processing step is probabilistic, we run the evaluation 10 times for each combination of dataset and attack. We report the mean and standard deviation of accuracy and attack success rates across the 10 runs.

7.1 Results

The results of the evaluation are shown in Table 3. We can see that simply using the data augmentation step of our adversarial defense already prevents up to two-thirds of the attacks without losing accuracy. This result indicates that adversarial examples for text classification are closely related to the data on which the model is fine-tuned and that state-of-the-art attacks rely on examples that are out-of-distribution with respect to the training data. When we additionally apply our post-processing procedure, between 70% and 92% of the remaining attacks are reverted. The *DA+PP* configuration reaches the lowest attack success rate across datasets and attack, while reducing the clean accuracy by only 1.18% in the worst case (IMDB). Finally, when we compare *DA+MA*$_5$ to *N+PP*, we see that the former reverts significantly fewer attacks than *DA+PP* and the latter degrades the clean accuracy. These results demonstrate the validity of our method as a defense against adversarial attacks.

In terms of the performance of the attacks, these results show that with the constraint on cosine similarity of words applied, TextFooler is by far the most effective attack, at least before post-processing. There is a simple reason for this, TextFooler already has that constraint and is the only attack out of the four to choose its candidate set directly from the counter-fitted embedding used to calculate the cosine similarity. On the other end of the spectrum, BAE's attacks success rate drops close to zero. This is because the intersection of the set of words proposed by the MLM, the set of words with cosine similarity greater than 0.5, and the set of words keeping the USE score above 0.936 is small, leaving very few valid candidates. A similar observation can be made for PWWS, although not as pronounced.

There is one more reason why TextFooler is more effective compared to the other attacks, despite an additional constraint on the USE score. While attacking a piece of text, this constraint on the USE score is not checked between the current perturbed text s' and the original text s, but instead between the current perturbed text s' and the previous version s''. This means that by perturbing one word at a time, the effective USE score between s and s' can be a lot lower than the threshold suggests. When discussing the effect of raising thresholds to higher levels in the next section, we do so by relying on TextFooler as the attack because it is the most effective, but we adjust the

Dataset	Method	Attack Success Rate (%)				
		TF_{cv50}	TF_{cv50}^{use88}	TF_{cv70}^{use85}	TF_{cv70}^{use90}	TF_{cv80}^{use90}
AG News	N	88.79	24.95	22.52	11.63	7.51
	DA	55.58	16.11	10.79	7.12	4.50
	DA+PP	4.49 ± 0.39	3.31 ± 0.28	2.07 ± 0.16	1.91 ± 0.17	0.99 ± 0.17
Yelp	N	91.40	49.22	42.59	25.18	11.09
	DA	38.46	13.74	10.34	7.78	2.87
	DA+PP	5.04 ± 0.35	3.9 ± 0.34	2.12 ± 0.21	2.28 ± 0.17	0.71 ± 0.13
IMDB	N	98.38	82.51	79.16	61.77	42.76
	DA	51.58	37.95	28.51	24.73	19.48
	DA+PP	5.81 ± 0.26	5.78 ± 0.4	3.56 ± 0.32	3.14 ± 0.28	2.67 ± 0.16

Table 4: Effectiveness of defense procedure for different combinations of thresholds.

constraint on the USE score to always compare to the original text. We believe this is the right way to implement this constraint, and more importantly, it is consistent with how we gathered data from Amazon Mechanical Turk.

7.2 Adjusted Thresholds

Next, we adjust the thresholds on the similarity constraints of the TextFooler (TF) attack such that the generated adversarial examples are better aligned with human judgement. In the notation used in Table 4, TF_{cvX}^{useY} corresponds to TextFooler with $cos_{cv}(w_i, w_i') \geq 0.X \; \forall i$ and $cos_{use}(s, s') \geq 0.Y$. A special case is TF_{cv50}, which corresponds to TextFooler without the constraint on the USE score.

As expected, stronger constraints on the generation of adversarial examples rapidly reduce the success rate of the attack. In particular, TF_{cv50}^{use88}, which corresponds to TextFooler with the same constraints as in the original implementation but without allowing the adversarial text to drift away from the original text, already decreases the attack success rate significantly. Regarding our proposed defense, data augmentation already decreases the attack success rates from 84.99 to 16.11 on AG News, from 90.47 to 13.74 on Yelp, and from 98.16 to 37.95 on IMDB. If we apply post-processing, we can revert most of the attacks across all datasets and attack configurations.

All in all, we see that when increasing the thresholds on the constraints (refer to Figure 5 in Appendix F to see that these are still not particularly strong constraints), the success rate of the attack drops significantly in all cases. This makes evident that when evaluated in a fair setup, where the adversarial examples are required to be semantically similar to the original sentence, BERT is considerably more robust than previous work suggests.

7.3 Data Augmentation vs. Adversarial Training

While adversarial training provides the model with data from the true distribution generated by an attack, our data augmentation procedure only approximates that distribution. The goal is to trade robustness for speed. However, similar to Ivgi and Berant (2021), we find that our procedure can even be superior to true adversarial training in some cases.

We compare two different strategies for adversarial training. ADV_{naive} denotes the simplest procedure for adversarial training in text classification: collect adversarial examples on the training set and then train a new model on the extended dataset consisting of both adversarial examples and original training data. We use TextFooler to collect these adversarial examples. On the complete training set, this results in $103'026$ adversarial examples on AG News, $179'335$ on Yelp, and $23'831$ on IMDB. For a more complex adversarial training, we follow Meng et al. (2021) by creating adversarial examples on the fly during training. We denote this method as ADV.

We compare the performance of data augmentation and adversarial training in Table 5. Interestingly, ADV_{naive} does not result in an improvement on Yelp and IMDB. We hypothesize that this is because Yelp and IMDB are easier to attack, resulting in weaker training data for the extended dataset. For example, 26% of the created adversarial examples on Yelp differ by only one or two words from the original text. On AG News this holds for just 11% of the adversarial examples. Furthermore, the average word replacement rate on Yelp is 16% compared to 24% on AG News. The same argument would also explain why, surprisingly, we reach higher robustness on Yelp and IMDB with

Dataset	Method	Clean Acc. (%)	Training Time (h:min)	Epochs	Attack Success Rate (%)		
					TextFooler	PWWS_{cv50}	$\text{BERT-Attack}_{cv50}$
AG News	Normal	94.57	0:19	2	84.99	16.38	20.72
	DA	**94.82**	5:33	12	52.37	10.73	18.61
	ADV	92.83	160:15	12	**34.54**	**6.50**	**9.38**
	ADV_{naive}	94.26	45:14	2	56.20	12.50	17.44
Yelp	Normal	**97.31**	0:32	2	90.47	33.26	49.53
	DA	97.10	9:08	12	**29.79**	**10.52**	**16.49**
	ADV	95.94	107:56	5	59.52	14.64	25.52
	ADV_{naive}	96.65	56:53	2	95.12	33.09	47.61
IMDB	Normal	93.77	0:17	2	98.16	65.77	88.44
	DA	94.21	5:31	12	48.31	29.49	40.91
	ADV	92.00[6]	–	3[6]	75.3[6]	–	–
	ADV_{naive}	93.16	34:19	2	100.00	62.75	88.79

Table 5: Comparison of data augmentation and adversarial training.

our data augmentation procedure compared to ADV. On IMDB, presumably due to the longer sequence lengths, we used the results from Meng et al. (2021) where available. It should also be mentioned that we trained ADV for fewer epochs on Yelp due to computational constraints.

Finally, the training times reported in Table 5 clearly show the large gains in compute time that our defense method provides in comparison to adversarial training. Considering that the training data increases by a factor of two, the overhead per epoch is only around 50% compared to normal training. Compared to ADV, we reach a speedup per epoch of almost 30×.

8 Limitations

In practice, our post-processing step cannot be decoupled from a black-box attack. It would be interesting to see how successful an attack can be when the whole system, including post-processing, is regarded as a single black-box model. We hypothesize that our defense would remain effective because the attack can rely much less on its search method for finding the right words to replace. We leave this analysis for future work.

One potential inconvenience of our defense is that it can not be applied if a deterministic answer is required. However, in many applications, such as spam filtering or fake news detection, we are only interested in making a correct decision as often as possible while being robust to a potential attack.

[6]Results taken from Meng et al. (2021).

9 Conclusion

Using a human evaluation, we have shown that most perturbations introduced through adversarial attacks do not preserve semantics. This is contrary to what is generally claimed in studies introducing these attacks (Jin et al., 2020; Ren et al., 2019; Garg and Ramakrishnan, 2020; Li et al., 2020). We believe that the main reason for this discrepancy is that recent research has focused on optimizing the success rate of textual adversarial attacks and has neglected the importance of preserving semantic meaning. However, in order to find meaningful adversarial examples that could help us better understand current models, we need to bring semantic preservation back into the equation.

Our experiments show that when semantic preservation is enforced, a state-of-the-art model like BERT is much more robust against adversarial attacks than reported in the existing literature. By using a simple data augmentation procedure that approximates the attack perturbations, a significant amount of adversarial examples can be prevented. This result emphasizes that the vulnerability of BERT against adversarial attacks stems mainly from the use of out-of-distribution data at inference time. In comparison to adversarial training, our data augmentation method is almost 30× more computationally efficient, and thus, it easily scales to large datasets and multiple epochs of training. Finally, our novel post-processing step completes our defense procedure and shows that most attacks can be prevented in a probabilistic setting without a severe impact on clean accuracy.

Ethical Considerations

In our experiments, we did not notice any sensitive or offensive information in our datasets or generated adversarial examples. However, one should note that it is still possible that the language models or augmentations used in our paper might generate sensitive or even offensive texts in rare cases. Hence, necessary precautions should be addressed when using our method in conditions like healthcare or large-scale scenarios.

References

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

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE Computer Society.

Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *IEEE Security and Privacy Workshops (SPW)*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for english. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. In *EMNLP*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

W Keith Hastings. 1970. Monte carlo sampling methods using markov chains and their applications.

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 *EMNLP-IJCNLP*.

Maor Ivgi and Jonathan Berant. 2021. Achieving model robustness through discrete adversarial training. *arXiv preprint arXiv:2104.05062*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *EMNLP-IJCNLP*.

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 *AAAI*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. In *EMNLP*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Zhao Meng, Yihan Dong, Mrinmaya Sachan, and Roger Wattenhofer. 2021. Self-supervised contrastive learning with adversarial perturbations for robust pretrained language models. *arXiv preprint arXiv:2107.07610*.

Zhao Meng and Roger Wattenhofer. 2020. A geometry-inspired attack for generating natural language adversarial examples. In *COLING*.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *EMNLP Findings*.

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

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

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

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *ACL*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021a. Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *UAI*.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021b. Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *AAAI*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *ACL*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *NeurIPS*.

Dataset	N	Reverted Attacks (Mean/Std) (%)		
		TextFooler	PWWS_{cv50}	BERT-Att_{cv50}
AG News	4	92.13 / 0.65	75.39 / 3.35	78.7 / 1.94
	8	92.49 / 0.79	76.27 / 2.87	79.94 / 1.54
	16	92.81 / 0.53	78.24 / 1.95	80.17 / 0.85
	32	92.97 / 0.24	76.57 / 1.61	81.07 / 0.88
Yelp	4	83.94 / 1.49	74.31 / 3.28	68.56 / 3.02
	8	85.33 / 1.32	75.88 / 1.4	70.5 / 1.97
	16	85.81 / 1.26	76.37 / 1.88	70.81 / 1.12
	32	86.26 / 0.74	76.96 / 0.79	71.31 / 2.16
IMDB	4	87.2 / 1.13	84.19 / 1.43	80.36 / 1.27
	8	87.96 / 0.92	84.62 / 0.88	80.85 / 0.91
	16	87.86 / 0.77	85.2 / 0.68	82.09 / 0.78

Table 6: Effectiveness of post-processing for different number of versions.

Dataset	Method	Clean Acc. (%)	Reverted (%)
AG News	MA_5	93.62	63.24
	MA_{10}	92.14	62.76
	MA_{20}	87.30	57.34
	MA_{30}	76.25	50.01
Yelp	MA_5	95.19	59.00
	MA_{10}	93.98	61.42
	MA_{20}	90.53	60.83
	MA_{30}	86.91	59.25
IMDB	MA_5	92.47	71.74
	MA_{10}	89.90	68.67
	MA_{20}	83.51	62.56
	MA_{30}	78.76	59.52

Table 7: By masking random tokens instead of exchanging words, more than half of the attacks can be reverted. However, the clean accuracy drops.

A Number of versions in post-processing

In order to understand the impact of the number of versions N created during the post-processing step, we can make the following analysis: Let us consider the augmented inputs as instances of a discrete random variable X. For $x \in X$ and a classification problem with K classes, let $l_{correct}(x)$ denote the value of the logit corresponding to the correct label and $l_j(x)$ denote the value of the j-th logit corresponding to a wrong label, such that $j \in \{1, ..., K-1\}$. We are only interested in the differences $g_j(x) = l_{correct}(x) - l_j(x)$. Ideally, we would like to make a decision based on the expectations of $g_j(X)$. An attack should be reverted if and only if

$$\mathrm{E}[g_j(X)] = \sum_{x \in X} g_j(x) p_X(x) \geq 0 \quad \forall j, \tag{6}$$

where $p_X(x) = \frac{1}{|X|}$. Because we cannot enumerate over all instances x, we approximate this with sums over just N instances

$$\sum_{i=1}^{N} \frac{g_j(x_i)}{N} \geq 0 \quad \forall j. \tag{7}$$

These are unbiased estimates of the expectations in (6) for any choice of N. By multiplying with N and plugging in the definition of $g_j(x)$, it can be verified that a decision based on (7) reverts the same attacks as a decision based on (5). The expectation estimates become more and more accurate as we increase N. Since we are making a discrete decision based on whether the expectations are ≥ 0, the estimate is more likely to be correct with more samples. If we assume that the true expectation is positive in most cases, this means we can generally expect a higher number of reverted attacks for higher N. Being more precise on the estimate also means we generally tend to make the same decision every time on the same example, therefore reducing the variance in the reverted attack rate. Table 6 shows results on reverted attacks for 4, 8, 16 and 32 versions (4, 8, and 16 on IMDB because of memory constraints) and generally confirms this. However, the results are already quite good with just four versions, so this is a trade-off between speed and accuracy, as creating N versions increases the batch size during inference by a factor N.

B Baseline for post-processing

Instead of replacing words with other words in Step 2 of our defense procedure, one could also think of other ways of slightly perturbing the adversarial examples to flip the label back to the correct one. To show that our method is superior to such simple perturbations, Table 7 shows the results of a baseline procedure in which we replace randomly chosen words with the `[MASK]` token. The reverted column shows an average over all attacks. Indeed, a significant portion of attacks can be reverted by masking just 5% of the words. However, further improving on that by masking more tokens fails, and the clean accuracy drops substantially. This is contrary to our procedure, in which we exchange 40% of the words with just a minimal decrease in accuracy.

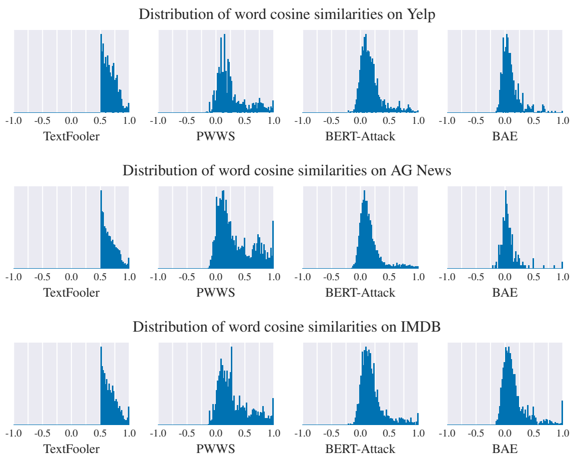


Figure 2: Distribution of cosine similarities of words.

Dataset	Attack	Orig. Word		Att. Word	
		Occ.	GT (%)	Occ.	GT (%)
AG News	TextFooler	736	67.31	18	24.63
	PWWS	889	60.04	24	16.06
	BERT-Att.	585	65.92	344	22.91
	BAE	617	52.66	4	9.31
Yelp	TextFooler	4240	72.79	19	44.60
	PWWS	5715	74.56	13	33.76
	BERT-Att.	4521	75.27	3398	35.55
	BAE	4601	76.03	44	41.87
IMDB	TextFooler	1362	69.25	47	41.48
	PWWS	1598	67.73	66	34.80
	BERT-Att.	1408	70.22	1016	36.27
	BAE	1221	67.44	23	45.12

Table 8: Median word occurrences of original words and attack words in training set (Occ.) and percentage of times that words have the highest relative frequency in ground truth class (GT).

C Word Frequencies

We observe that attacks frequently introduce words that rarely occur during training. Table 8 shows median word occurrences (Occ. column) of original words and attack words in the training set for different attacks. The results are striking and a further justification for using data augmentation. It is also interesting to see that BERT-Attack acts differently in that regard. We assume this is because BERT-Attack has the weakest constraints (no constraint on cosine similarity of words and a weak constraint on USE). This could allow BERT-Attack to find more effective perturbations than other attacks that have to choose from a set of more similar words and then rely on the ones the model does not know.

Table 8 further shows that attacks often use words with higher relative frequency in other classes. Column GT reveals the percentage of times that the original words and attack words have the highest relative frequency (word occurrences in class divided by the total number of words in the same class) in the ground truth class. It can be observed that attacks often introduce words with higher relative frequency in a different class. This raises whether there is some justification in the model's decision to change its prediction. After all, for a simpler model based on word statistics, we would not be surprised about a change in prediction if sufficiently many words are exchanged with words that appear more often in other classes.

D Cosine Similarities of Words

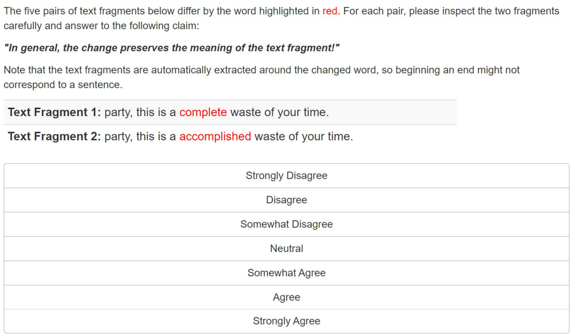
In a counter-fitted embedding, perfect synonyms are supposed to have a cosine similarity of 1, and perfect antonyms are supposed to have a cosine similarity of 0. Figure 2 shows the distribution of cosine similarities for the four attacks on all datasets.

E Sequence Length vs Attack Success Rate

Longer input sequences are, in general, easier to attack. To see this, take an attack without constraint on sentence similarity (PWWS, for example). Assuming a maximum replacement rate of 0.4, the number of potential adversarial examples for an input sequence of length l is $(0.4 \cdot l)^{|\mathcal{C}|}$, where $|\mathcal{C}|$ is the size of the candidate set.

F Details on Human Evaluation

We relied on workers who completed at least 5000 HITs with over 98% success rate. For the word-pairs, we showed the workers 100 pairs of words in a google form. In order to ensure a good quality of work, we included some hand-designed test cases at several places and rejected workers with strange answers on these word-pairs. These test cases were [*good*, *bad*], [*help*, *hindrance*] (expected answer "Strongly Disagree" or "Disagree") and [*sofa*, *couch*], [*seldom*, *rarely*] (expected answer "Strongly Agree" or "Agree"). In a first test run, surprisingly, many workers agreed on antonyms like good and bad, which is why we added a note with an example and emphasized that this is about whether the meaning is preserved and not about whether both words fit into the same context. Workers were paid 2.0$ for one HIT with 100 pairs and 4 test cases. We showed every pair of words to ten


The five pairs of text fragments below differ by the word highlighted in red. For each pair, please inspect the two fragments carefully and answer to the following claim:

"In general, the change preserves the meaning of the text fragment!"

Note that the text fragments are automatically extracted around the changed word, so beginning an end might not correspond to a sentence.

Text Fragment 1: party, this is a complete waste of your time.

Text Fragment 2: party, this is a accomplished waste of your time.

Strongly Disagree
Disagree
Somewhat Disagree
Neutral
Somewhat Agree
Agree
Strongly Agree

Figure 3: Screenshot of the human evaluation used to evaluate words with context.

workers and calculated the mean. A screenshot of the form can be found in Figure 4. For the words with context, we used the internal Amazon form because it allowed for a clearer presentation of the two text fragments (see Figure 3). We always presented five pairs of text fragments in one HIT and rejected workers that submitted the hit within less than 60s to ensure quality. Workers were paid 0.5$ for one HIT with five pairs. We showed every pair of text fragments to five workers and calculated the mean.

F.1 Metrics vs. Human

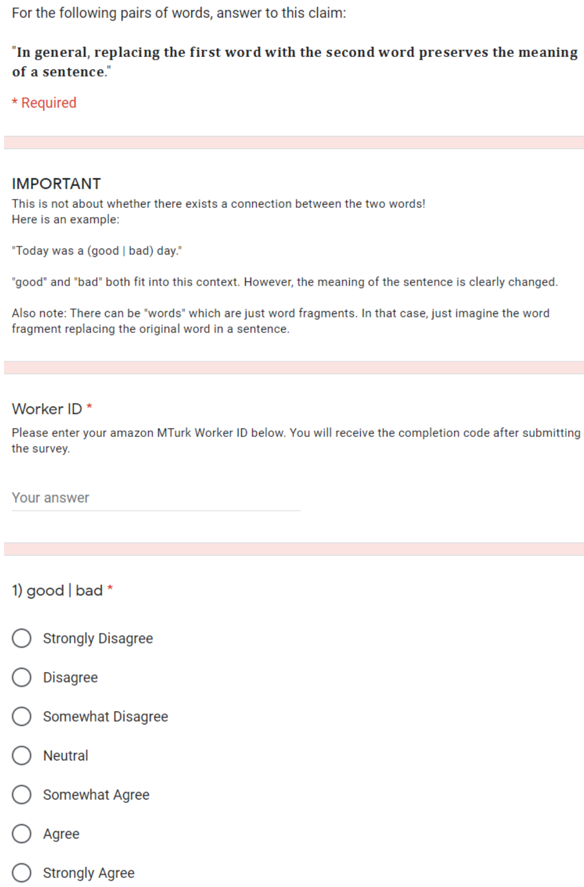
Figure 5 shows the probability that a perturbation is considered valid (for $T_h = 5$) as a function of cosine similarity of words and as a function of USE score. The plots are based on the 400 words with context from the different attacks which were judged by humans. We use left-aligned buckets of size 0.05, i.e., the probability of a valid perturbation for a given cosine similarity x and metric $m \in \{cos_{cv}(\cdot,\cdot), cos_{use}(\cdot,\cdot)\}$, is estimated as

$$\frac{\text{count}[(S_h \geq T_h) \wedge (m \in [x, x+0.05))]}{\text{count}[m \in [x, x+0.05)]}. \quad (8)$$

It can be observed that there is a strong positive correlation between both metrics and the probability that a perturbation is considered valid, confirming both the validity of such metrics and the quality of our human evaluation. However, the exact probabilities have to be interpreted with care, as the analysis based on one variable does not consider the conditional dependence between the two metrics.

G Datasets

For our experiments, we use three different text classification datasets: AG News, IMDB, and Yelp.


For the following pairs of words, answer to this claim:

"In general, replacing the first word with the second word preserves the meaning of a sentence."

* Required

IMPORTANT
This is not about whether there exists a connection between the two words!
Here is an example:

"Today was a (good | bad) day."

"good" and "bad" both fit into this context. However, the meaning of the sentence is clearly changed.

Also note: There can be "words" which are just word fragments. In that case, just imagine the word fragment replacing the original word in a sentence.

Worker ID *
Please enter your amazon MTurk Worker ID below. You will receive the completion code after submitting the survey.

Your answer

1) good | bad *

Strongly Disagree
Disagree
Somewhat Disagree
Neutral
Somewhat Agree
Agree
Strongly Agree

Figure 4: Screenshot of the Google form used to evaluate similarity of words.

On Yelp, we only used examples consisting of 80 words or less. Especially comparing to ADV would have been much harder otherwise. Statistics of the three datasets are displayed in Table 9.

Dataset	Labels	Train	Test	Avg Len
AG News	4	120’000	7’600	43.93
Yelp	2	199’237	13’548	45.69
IMDB	2	25’000	25’000	279.48

Table 9: Statistics of the three datasets.

AG News (Zhang et al., 2015) is a topic classification dataset. It is contructed out of titles and headers from news articles categorized into the four classes “World”, “Sports”, “Business”, and “Sci/Tech”.

Yelp (Zhang et al., 2015) is a binary sentiment classification dataset. It contains reviews from Yelp. Reviews with one or two stars are considered negative, reviews with three or four stars are considered positive.

IMDB is another binary sentiment classification dataset. It contains movie reviews labeled as posi-

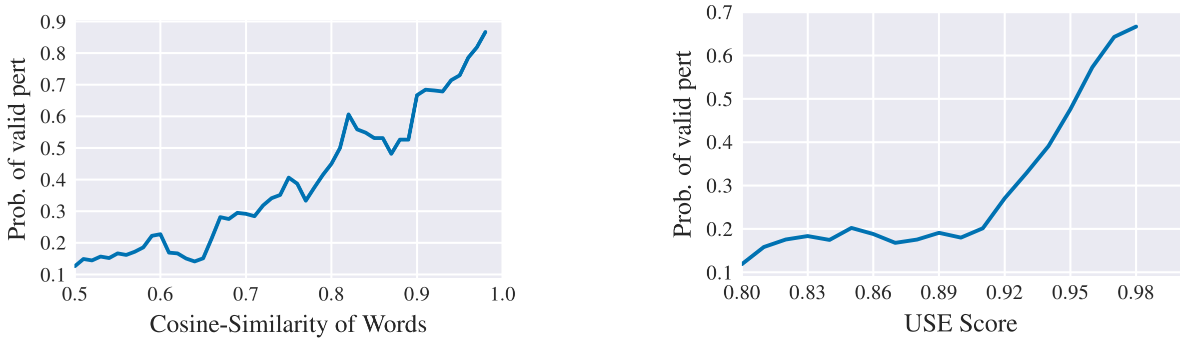


Figure 5: The probability that a perturbation is considered valid by a human, as a function of cosine similarity of words (left) and USE score (right). T_h is set to 5, i.e. an average score of 5 is required to be considered valid.

tive or negative.

H Implementation

Training We use *bert-base-uncased* from huggingface[8] for all our experiments. The normal models were fine-tuned for two epochs with a learning rate of 2e-5. We restrict the maximum input length to 128 tokens on AG News and Yelp. For IMDB, the maximum input length is set to 512. For the training with data-augmentation, we train for 12 epochs with a starting learning rate of 2e-5 and a linear schedule. We evaluate the robustness on an additional held-out dataset after every epoch. For a threshold of 0.5 on the cosine similarity of words, the robustness reaches its peak after the last epoch. However, we find that two or three epochs are already enough for larger thresholds on the cosine similarity of words. All our experiments are conducted on a single RTX 3090.

Attacks We use TextAttack[9] for the implementations of all attacks, including the ones with adjusted thresholds. For adversarial training, we adapt the code from Meng and Wattenhofer (2020).

[8] https://huggingface.co/transformers/

[9] https://textattack.readthedocs.io/en/latest/