XAI-Attack: Utilizing Explainable AI to Find Incorrectly Learned Patterns for Black-Box Adversarial Example Creation

Markus Bayer*, **Markus Neiczer**†, **Maximilian Samsinger**†, **Björn Buchhold**†, **Christian Reuter***
*PEASEC / Technical University of Darmstadt, Darmstadt, Germany
†CID GmbH, Freigericht, Germany
bayer@peasec.tu-darmstadt.de

Abstract

Adversarial examples, capable of misleading machine learning models into making erroneous predictions, pose significant risks in safety-critical domains such as crisis informatics, medicine, and autonomous driving. To counter this, we introduce a novel textual adversarial example method that identifies falsely learned word indicators by leveraging explainable AI methods as importance functions on incorrectly predicted instances, thus revealing and understanding the weaknesses of a model. Coupled with adversarial training, this approach guides models to adopt complex decision rules when necessary and simpler ones otherwise, enhancing their robustness. To evaluate the effectiveness of our approach, we conduct a human and a transfer evaluation and propose a novel adversarial training evaluation setting for better robustness assessment. While outperforming current adversarial example and training methods, the results also show our method's potential in facilitating the development of more resilient transformer models by detecting and rectifying biases and patterns in training data, showing baseline improvements of up to 23 percentage points in accuracy on adversarial tasks. The code of our approach is freely available for further exploration and use.

Keywords: Explainability, Statistical and Machine Learning Methods, Evaluation Methodologies, Text Analytics, Validation of LRs, Other

1. Introduction

Adversarial examples are specially crafted inputs to machine learning models that aim to trick them into making incorrect predictions. These inputs can be created by deliberately modifying existing examples to fool the algorithm (Goodfellow et al., 2015). It is evident that such deceptions can have serious consequences if they are framed as attacks. One example is crisis informatics, where deep learning models are used by response teams to gather and analyse information about, e.g., a natural disaster or terrorist attack. An attacker could construct examples that have nothing to do with the incident, but are designed in such a way that the model recognises them as relevant, leading to poorer insights and lower confidence in the information gathered.

Beyond that, the exploration of adversarial examples is very important, as it can show where a model has learned spurious correlations or shortcuts. Many machine learning practices result in the preference of shortcut solutions (Geirhos et al., 2020). On the one hand, this can be desirable, as the models may be less prone to overfitting and generalize better (Rasmussen and Ghahramani, 2000). On the other hand, this can lead to solutions that are too simple, which, for example, result from patterns and biases in the training dataset that correlate with classes in the data but are not actually responsible for the class. The research field of adversarial examples not only enables us to recognise them, but also offers methods to correct them. A commonly used strategy is adversarial training, where the model is re-trained with adversarial examples. However, current methods suffer from small or very specific robustness gains, partly due to their narrow design and partly due to ineffective evaluation methods.

Therefore, this paper considers an optimal adversarial example as one that has a significant learning factor. Instead of following current methods, we aim to uncover the model's incorrectly learned patterns by analyzing its erroneous predictions. As second novelty, we also propose to use feature/token attribution explainable AI (XAI) methods (e.g. LIME or SHAP) as importance functions to highlight the model's incorrectly learned indicators. Incorporating XAI into research on adversarial examples offers the opportunity to make more sophisticated importance calculations and to make them more flexible by providing a framework that can easily replace them. Our method coupled with adversarial training allows for a more robust model by highlighting and erasing patterns of the training data identified by the model which are either not truly indicative or not solely responsible for class determination. Hence, our approach promotes models to favor decision boundaries that are intricate when necessary, and straightforward when appropriate.

Unfortunately, there is currently no optimal evaluation method to ascertain a model's true robustness. The greatest challenge in adversarial train-

LREC-COLING 2024, pages 17725–17738
20-25 May, 2024.

ing evaluation is that biases in training data often echo in the test data (Geirhos et al., 2020), leading to poor results. Therefore, we propose an out-of-distribution evaluation method that specifically addresses attack robustness, which, in combination with human and transfer evaluations, shows the performance of XAI-Attack with transformers.

Our work contributes the following aspects:

(C1) A novel method for creating adversarial examples based on identifying indicators for wrong predictions.

(C2) Proposal to utilize XAI methods as importance functions for adversarial example creation.

(C3) A novel out-of-distribution evaluation setting for adversarial training that enables a more accurate assessment of robustness.

The code of this study is freely available[1].

2. Related Work

There is no unique formal definition of adversarial examples in the literature. In this work, we follow the definition of Goodfellow et al. (2015), which states that an adversarial example is an instance that has been intentionally curated from existing examples to fool the machine learning model. We add for clarification that the new instance should be semantically similar to its original instance. An adversarial example is deemed semantically similar to its original instance if, despite any textual variations, the underlying meaning pertaining to its label remains unchanged. Some works, such as (Wang et al., 2022a), also imply that the adversarial examples should only be modified by a small change, ideally imperceptible to us humans, a prerequisite which we drop explicitly in our definition, as, for example, Brown et al. (2017) or Ebrahimi et al. (2018b). The adversarial examples can be created in black-box and white-box form, focusing on the model to be tricked, also known as the victim model. A white-box attack is one in which the internals of the model are completely transparent to the attacker (Biggio and Roli, 2018). In black-box attacks, the attacker can only query the victim model for an instance and get the prediction. Depending on the victim model, it outputs the prediction as soft labels, i.e. the softmax outputs, or as hard labels, i.e. only the class labels.

2.1. Feature Space

While adversarial examples are intensively studied in computer vision (Kurakin et al., 2017; Szegedy et al., 2014; Papernot et al., 2017), they are more difficult to create for textual data due to the discrete nature and semantic coherence of text (Garg and Ramakrishnan, 2020; Jia and Liang, 2017). This is also the reason for adversarial example methods being divided into data and feature space. In terms of feature space, they are often designed to maximise loss by adding noise in a white-box attack scenario. To solve the search for the right perturbation, several different maximisation methods and variants have been proposed in research (Bayer et al., 2023): PGD (Combettes and Pesquet, 2011; Goldstein et al., 2014), FreeAT (Shafahi et al., 2019), YOPO (Zhang et al., 2019), FreeLB (Zhu et al., 2020), VAT (Miyato et al., 2016, 2017), SMART (Jiang et al., 2020), ALUM (Liu et al., 2020), and more. These feature space methods are often integrated directly into the training process, with SMART achieving the highest adversarial training scores in many tasks (Bayer et al., 2023).

2.2. Data Space

However, our method acts in the data space. This includes, e.g., the work of Ebrahimi et al. (2018b), in which the letters of the input texts are flipped in such a way that the loss increases. For this, the gradients of the method and accordingly a white-box scenario are needed. Jin et al. (2020) propose the black-box method TextFooler, which works at the word level and replaces important words with synonyms that are chosen based on embedding similarity and the highest change in prediction confidence. BERT-Attack (Li et al., 2020) and BERT-based adversarial examples (BAE) (Garg and Ramakrishnan, 2020) can be seen as variants of this approach, where BERT is used to create a list of substitution words. BERT-Attack differs from BAE in that it differentiates between words and subwords. BAE, conversely, proposes a replace and insert operation and additionally uses a Universal Sentence Encoder to filter the generated tokens to ensure high semantic similarity to the original text. The textbugger method of Ye et al. (2022) also finds the important words and then creates either word-level changes with GLoVe embeddings or character-level changes based on rules. The authors propose a black-box and white-box variant. There are also methods for sentence-level adversarial examples, e.g. Iyyer et al. (2018), which paraphrase a sentence with certain syntactic structures, the Entailment Preserving Transformations method of Thorne and Vlachos (2019), which transforms the sentences based on templates, or the SSAE network by Li et al. (2023), which consists of two auto-encoders to preserve syntax as well as semantics and to insert pertubations into the latent space. The research field also includes specialised methods such as the work of Qaraei and Babbar (2022) for extreme multilabel text classification scenarios, Song et al. (2022) for text retrieval or Wan et al. (2022) and Ebrahimi et al. (2018a) for neural machine translation.

[1] https://github.com/PEASEC/XAI-Attack/

2.3. Robustness Evaluation

Research indicates that numerous methods exist for enhancing the robustness against adversarial examples, such as employing adversarial example generation alongside adversarial training, or detecting and filtering malicious inputs (Goyal et al., 2023; Wang et al., 2022b). Assessing adversarial robustness is crucial for determining the effectiveness of a defense method and, ultimately, the true robustness of a model. The efficacy of adversarial training is often gauged by evaluating the resulting model on the test or validation set of a task (Ebrahimi et al., 2018a; Jiang et al., 2020; Li et al., 2020). However, evaluating the adversarial robustness of a machine learning model solely on those sets may not always be sensible as they often share the same biases as the training data (Geirhos et al., 2020). There are some frameworks, such as CleverHans (Papernot et al., 2018) and FoolBox (Rauber et al., 2018), that provide benchmarks for robustness. These frameworks offer different adversarial attack methods that can be used to evaluate the robustness of a model with a given dataset. While this approach mitigates the issue of evaluating robustness using only test or validation data, it introduces another potential challenge: the adversarial attack methods implemented by the framework may not generate high-quality adversarial examples, potentially skewing the robustness assessment.

2.4. Research Gap

XAI-Attack can be classified in the group of word-level data space perturbation methods. Unlike other adversarial example methods, it addresses the weaknesses of the models by extracting indicators from wrong predictions. Thus, it does not require access to the internals and, when combined with adversarial training, can greatly improve robustness. It creates the attacks in a black-box form and only needs the hard labels, i.e. the labels predicted by the model, but not the softmax outputs.

As discussed before, we abstract from any specific importance functions by proposing to use XAI methods that are not only easily replaceable in our framework but also tend to be much more sophisticated. This way, we combine the process of creating adversarial examples with any feature/token attribution XAI method to find the most important words. Importance/influence functions in current research, e.g. (Li et al., 2020; Garg and Ramakrishnan, 2020; Jin et al., 2020), simply omit words and calculate how much this changes the prediction score. The XAI method LIME (Ribeiro et al., 2016), e.g., also omits words but calculates additional weights for the perturbed instances based on similarity to the original instance and then trains a locally interpretable model from which the importance scores are derived. Moreover, LIME does not require prediction results like the others, which makes our method suitable for a hard label environment. In our experiments we show the usage of LIME and SHAP (Lundberg and Lee, 2017).

Furthermore, our method is designed with the objective in mind of making models more adversarially robust so that they are less vulnerable to attacks while maintaining a high performance. Evaluating a model on the test set of the same task on which it was adversarially trained can be problematic due to the presence of biases in all task sets. Frameworks attempting to estimate adversarial robustness by using other adversarial example generators on a given set face the issue that these examples may be of low quality and invalid. To address this, we propose a novel evaluation setting by evaluating adversarial training on the Adversarial GLUE (Wang et al., 2022) dataset, which contains high quality and valid adversarial examples for GLUE tasks.

3. Attack Design

3.1. Problem Formulation and Requirements

In formulating the problem, we follow the example of Ye et al. (2022). Suppose we have a victim model f and a text instance x consisting of n words that has the label y and is correctly predicted by f. x' is an adversarial example of x, iff it is semantically similar to x and changes the prediction of the victim model f, i.e., $f(x') \neq f(x)$. It is constructed by inserting adversarial words into the instance x.

While our method for generating adversarial examples aims to train a robust deep learning model, it requires only rudimentary access to the model, making it potentially useful for constructing attacks as well. As previously described, the internals of the model cannot be viewed and the user receives no indication of the confidence in the prediction other than the hard label. In terms of the XAI method, we focus on LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017), which can explain a model prediction using only the hard labels and have proven to be advantageous in XAI research. It is also possible to use different variants that may require more access to the model. XAI-Attack only requires an feature/token attribution XAI method, i.e. one that provides a distribution over tokens indicating the correlation strength between input tokens and output.

In addition, the adversary needs a hold-out set. Ideally, this data does not come from the data the model was trained on, but from new annotated examples or a development set. However, it is also possible to separate a part of the training data for the generation of adversarial examples. Due to

the design of our method, the model is afterwards trained with the data kept out anyway.

3.2. XAI-Attack

XAI-Attack uses an XAI method to highlight words which indicate the wrong class. In the following, these words are called adversarial words because they are used to create adversarial examples. However, potential and real adversarial words must be distinguished according to our definition of adversarial examples. The words returned by the XAI method are potential adversarial words because their insertion can change the semantics, whereas real adversarial words do not. Therefore, these potential adversarial words are subsequently cleaned by a targeted filtering method. The resulting adversarial words are then used to create adversarial examples. The procedure is demonstrated in Figure 1 and described in detail below.

Step 1: Finding potential adversarial words

The first step of our method is to use a XAI method to find words that are responsible for false predictions. This is achieved by letting the victim model make predictions on the hold-out data. For each instance in this set that was incorrectly predicted, we apply a XAI method, which is able to highlight words responsible for the incorrect prediction. The word highlighted as most indicative of the wrong class is then considered as a potential adversarial word, i.e. a word that can potentially change the label without changing semantics.

This step takes advantage of a common problem with training deep learning models, as they tend to overfit quickly and find spurious correlations in the training data due to sampling and other biases. To illustrate, we anticipate a small example from a sentiment task. Using the XAI method, we can see that an example word responsible for changing the negative sentiment into a positive one is *like*. It stands to reason that the word *like*, in its verb meaning *to find someone or something pleasant or satisfying*, is an indicator of a positive mood. Nevertheless, the word *like*, in its meaning of e.g. the preposition as *similar to*, is not an indicator of positive sentiment. There are many words that are not edge cases like this one, but used as indicators of a class, since they occur frequently in the respective class, while not having semantic significance for the classification (see Section 4.5). From this viewpoint, it is apparent that our approach leans towards regularisers and eradicating inaccurate bias.

Step 2: Filter for label invariant adversarial words. [Optional]

There are words which in most cases truly denote one class, but can have a different meaning in very specific contexts, and which are also identified as potential adversarial words in step 1. The word *enjoy* in the sentiment task can serve as an illustrative example. While in most cases it is a word used in positive contexts, it can also be used in negative ones: "Hard to say who might enjoy this" (from SST2). If we now see *enjoy* as a real adversarial word, in adversarial training the model would be forced to reject it as an indicator of the positive class. In the best case, the model learns a more robust indicator, e.g., by distributing the weight of the decision of *enjoy* onto the context. In the worst case, however, it would discard a very valuable indicator and even learn further biases, resulting in poorer performance and less robustness.

Therefore, we try to filter out the words that could change the semantics in relation to the label. Besides no filtering of words, we propose two methods, one based on count of label changes and one based on indicator words for the correct class.

Count of label changes. In the count-based method, we analyze how many adversarial examples can be generated with a potential adversarial word. If the count exceeds a certain threshold relative to the class size, we assume that the semantics change in relation to the label and exclude all of those generated adversarial examples.

Indicator words for the correct class. The other method utilizes the additional data that was correctly predicted by the model. This data is explained using the XAI method and the resulting words are matched with the potential adversarial words. Potential adversarial words, which are an indicator of correct prediction, are excluded from further processing as they could semantically alter the instance in terms of the label.

Step 3: Creating adversarial examples

The adversarial words may change the label only for certain instances based on the position or context in the text. Therefore, we now check whether these words also change the labels of instances that were originally predicted correctly. That is, we insert the adversarial words into the correctly predicted instances of the additional data. There are many possibilities for where a word can be inserted, and although we also test random insertion, our main experiments are based on just prefixing them (recall that imperceptibility is not a criterion in this work). For more ideas on inserting an adversarial word into an instance, see Section Limitations. The instances whose labels have been changed by the insertion of adversarial words are then the resulting adversarial examples.

4. Experiments

4.1. Experiment Types

Our experimentation perspective lies in the questions (1) how often the method actually generates

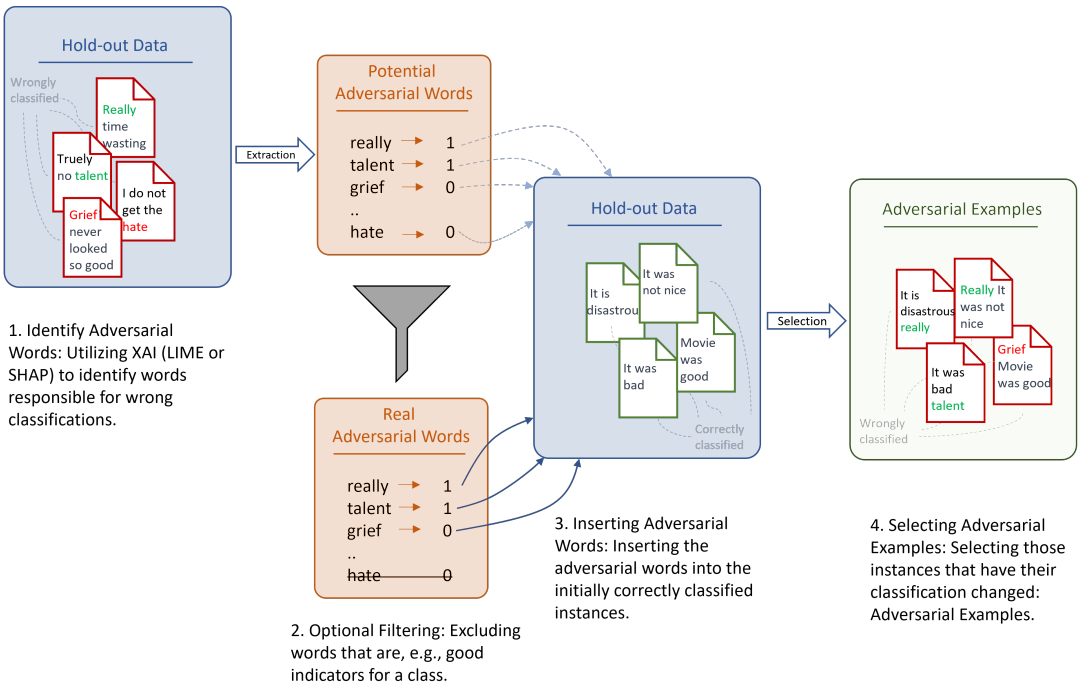

Figure 1: Illustration of XAI-Attack.

examples that do not change the label but the prediction, (2) whether a model trained with the examples becomes more robust, and (3) whether the adversarial examples can be transferred to other models.

Human Evaluation: To measure how often the method generates true adversarial examples, we perform a human evaluation in which the generated instances of each dataset are labelled.

Adversarial Testing and Training: The measurement of robustness is based on the benchmark of Wang et al. (2022). The benchmark consists of created and human-rated adversarial examples for various GLUE tasks. For each task, we test whether models trained with XAI-Attack adversarial examples from the normal GLUE tasks are more robust than those trained without or with adversarial examples from other methods. For completeness, we also include adversarial training experiments on standard GLUE tasks.

Adversarial Transfer: In the final setting of our experiments, we predict whether the adversarial examples are transferable to another model. To do this, we create the adversarial examples with distilBERT and then measure the impact on another distilBERT, a BERT and a RoBERTa model.

Unlike some previous work in the field of adversarial examples, we are not investigating the success rate as we believe that the success rate alone is no indication of the quality of the method.

4.2. Datasets & Model Settings

The tasks we focus on are the same as in (Wang et al., 2022) due to the adversarial testing experiment. That is, the experiments are conducted with the datasets SST-2, RTE, QNLI, MNLI and QQP from the GLUE benchmark (Wang et al., 2018). For the experiments we use distilBERT (Sanh et al., 2019) as the main model[2]. In terms of robustness measures, we employed only standard methods like weight decay and dropout, without incorporating any additional techniques.

4.3. Human Evaluation

For this section, first we go into the details of the quantitative experiment, then some adversarial examples are examined. The adversarial examples used in this section are generated with LIME and the optional filtering step based on the indicator words for the correct class (see Section 3.2).

For each task, we randomly selected 100 instances of the original dataset and 100 instances of the adversarial dataset with uniform class distribution. The resulting 200 instances for each task were randomly labelled by two independent annotators. A comparison of human performance can show us

[2]Main experiments: distilBERT-base-uncased | Transfer models: BERT-base and RoBERTa-base | Parameter: Standard Huggingface with 3 epochs, 500 warmup steps, learning rate of 1e-3, and weight decay of 0.01 | Implementations: BAE and BERT-Attack ($k = 7$) from TextAttack (Morris et al., 2020), SMART from the original implementation (Jiang et al., 2020)

Dataset	Original Data	Adversarial Data
SST2	0.8787	0.7534
RTE	0.9150	0.9600
QQP	0.8550	0.7338
QNLI	0.8994	0.7836
MNLI-mm	0.7626	0.8333
MNLI-m	0.7980	0.7828

Table 1: Human evaluation: Two annotators assessing 100 original instances and 100 adversarial instances (generated using LIME and indicator filtering) of each task, with the averaged accuracy.

whether the adversarial examples are mostly valid if the two metrics for the original and adversarial data of each task are similar (note that we do not test imperceptibility, but only whether an adversarial example changes the gold label). For the rationale and sizes in this experiment, we followed standard practice in the field (see (Li et al., 2020; Garg and Ramakrishnan, 2020; Jin et al., 2020)).

Table 1 shows the results of the human experiment, measured by the mean accuracy of the two annotators. The results of the original and the adversarial examples are very similar, bearing in mind that the classifier is wrong 100% of the time on the adversarial data. There are even tasks where the adversarial examples match the given labels more than the original data. This is of course due to variance in the selected sample, which was to be expected since not the entire dataset is labelled. Moreover, the agreement between the annotators (Cohen's kappa) is substantial for the non-adversarial data at 0.6681 and for the adversarial data at 0.6465 according to Landis and Koch (1977). Looking at the results as a whole, it is clear that the adversarial examples are of very high quality and only change the semantics in a few cases, but deceive the trained classifier in 100% of the cases.

This can be further illustrated by looking in detail at some adversarial examples. The adversarial examples listed in Table 2 were chosen because of the different insights that can be gained from them. The first instance shows a very subtle adversarial example, with only the letter "q" added to the instance. This already makes the classifier predict that the second sentence is a valid answer to the question of the first sentence. The second example also shows that such trained classifiers are very likely to memorise words and phrases from the training data. The word "enjoy" occurs 512 times in the positive training data and only 98 times in the negative training data of the sentiment task. It stands to reason that the model simply predicts almost every instance in which the word "enjoy" occurs as positive. While this is mostly true, there are some exceptions, such as this second example. Likewise, it is very probable that such a word changes the semantics of the instance when it is inserted, as we can see in the next example in the table. In such a case, the count-based filtering method might have excluded the word "enjoy". In sentiment prediction, it is also very odd to see that the word "better" is an adversarial word in the fourth example, turning a positive instance into a negative one, predicted by the classifier. The fifth example is a very interesting case of a wrong adversarial example. Here, the word "Java" is inserted into the first question, causing a semantic shift where the resulting question is a duplicate of the second question. While one might think that "Java" is not a good adversarial word, it is even more interesting to see that the insertion of "Java" in the second question also leads to a change in the prediction, while the semantics are not changed this time.

4.4. Adversarial Testing

The rationale behind the adversarial testing experiment is to test if models trained with adversarial examples from XAI-Attack are more robust than those trained with no or other adversarial examples. For this, we first generate adversarial examples for a standard GLUE task. Then, we test a model trained together with the task data and these adversarial examples on a separate dataset, that was deliberately created to measure robustness on the same task. The process is visualized in Figure 2.

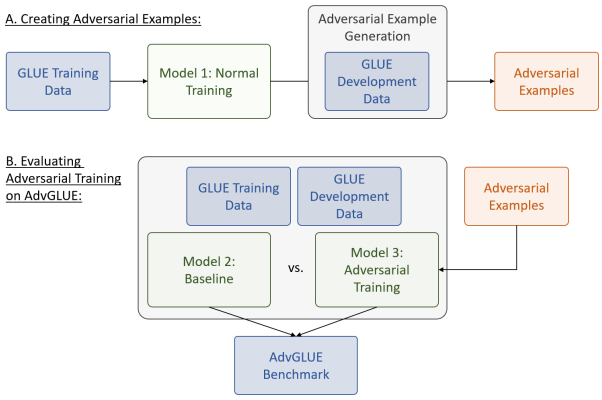

Figure 2: Adversarial testing illustration

In detail, we first train models based on the training data from the original GLUE datasets (non-adversarial). For these models, we then create adversarial examples for the development set of the same GLUE task (non-adversarial). Subsequently, a model based on the training set, the development set and additionally with the adversarial examples is tested on the benchmark development sets of the Adversarial GLUE benchmark (Wang et al., 2022). For the baseline, we omit the adversarial examples and train only on the GLUE training and development data.

Task	Instance	Label	Prediction	Valid
QNLI	Question: What issue has been plaguing the civil disobedience movement. Sentence: *q* It has been argued that the term "civil disobedience" has always suffered from ambiguity and in modern times, become utterly debased.	Not Entailment	Entailment	✓
SST2	*enjoy* not only unfunny , but downright repellent .	Negative	Positive	✓
SST2	*enjoy* but it could have been worse .	Negative	Positive	✗
SST2	*better* good old-fashioned slash-and-hack is back !	Positive	Negative	✓
QQP	Question1: *Java* What is abstract class and methods? Question2: What is abstract class and methods in java?	Not Duplicate	Duplicate	✗
QQP	Question1: What is abstract class and methods? Question2: *Java* What is abstract class and methods in java?	Not Duplicate	Duplicate	✓

Table 2: Examples of adversarial instances from different datasets (inserted adversarial words shown in blue). Valid adversarial examples are those that do not change the semantics in relation to the label.

As part of this experiment, we also compare our method with the state-of-the-art word-level adversarial example methods, BAE (Garg and Ramakrishnan, 2020) and BERT-Attack (Li et al., 2020), as well as with the adversarial training method SMART (Jiang et al., 2020). Regarding the ablations, we test five different settings of XAI-Attack. For three of them, the filtering techniques mentioned in Section 3.2 are used. Based on pre-evaluation experiments, we set the parameters for count-based filtering so that a word is discarded if it changes more than 30% of the class data, and for indicator-based filtering so that a word is an indicator of the correct class if it explains more than 1% of this class. In the fourth ablation scenario, adversarial words are randomly inserted into the instances. Finally, in the fifth scenario, we utilize the SHAP XAI method to identify adversarial words. No filtering is applied in either of the two scenarios.

The accuracy results of the baseline, XAI-Attack with the five different settings, BERT-Attack, BAE, and SMART are shown in Table 3. When analysing the results, it is important to bear in mind that the benchmark itself is adversarial, i.e. a trained model gets most examples wrong. The adversarial training method with our XAI-Attack shows significant improvements over the baseline, BAE, BERT-Attack, and SMART. XAI-Attack with indicator and without filtering is able to increase performance on all datasets compared to the baseline. In the case of the QPP dataset, XAI-Attack with no filtering even achieves an improvement of 23.08 accuracy points. This method also produces the highest results overall. While the adversarial example method based on count filtering seems to be the worst filtering type of XAI-Attack, sometimes degrading the results and showing weaker improvements, the indicator filtering method shows the most consistent improvements. The difference between the method without filtering (highest improvements, but somewhat inconsistent) and the method with indicator filtering (more consistent, but not highest improvements) can also be explained intuitively. Filtering excludes words from the list of potential adversarial words that may be important for predicting the correct class and in fact would change instances semantically, which is why the results consistently improve and the method without filtering may worsen the prediction quality in some cases. On the other hand, words that do not change instances semantically and would have had a high learning effect if they had been included in the adversarial training may also be excluded in this way, resulting in lower improvements than the more open method. In general, if the respective use case does not allow much tuning and validation (or its use in a general framework), we would recommend using the indicator filtering method, as we have noticed the most consistent improvements. However, if the use case allows for tuning and validation, we would also recommend trying to use XAI-Attack without filtering, as it might produce even better results. Besides, the results show that XAI-Attack has only mixed performance with SHAP (Lundberg and Lee, 2017), which is also evident in the adversarial words extracted with the method. We found that these have only little semantics with respect to the label. Finally, the results also highlight that random insertion can further improve already good results, as in the case of SST2, giving rise to further research into more sophisticated insertion methods.

4.5. Adversarial Training on Standard GLUE

While we demonstrated that XAI-Attack significantly enhances the resilience of transformers on the Adversarial GLUE benchmark, this subsequent experiment focuses on evaluating the effectiveness of adversarial training on the standard GLUE tasks.

Method	SST2	RTE	QQP	QNLI	MNLI-mm	MNLI-m
Baseline	0.3243	0.5802	0.5513	0.5945	0.2901	0.3636
XAI-Attack						
↳ No Filt.	0.4595 (↑)	**0.8025** (↑)	**0.7821** (↑)	**0.6554** (↑)	0.3827 (↑)	0.5785 (↑)
↳ Count Filt.	0.3851 (↑)	0.5161 (↓)	0.6795 (↑)	0.6351 (↑)	0.3642 (↑)	0.4545 (↑)
↳ Indicator Filt.	0.4527 (↑)	0.6420 (↑)	0.7307 (↑)	0.6283 (↑)	**0.4383** (↑)	**0.5868** (↑)
↳ Rand. Ins.	**0.5067** (↑)	0.6914 (↑)	0.7179 (↑)	0.5608 (↓)	0.3641 (↑)	0.4876 (↑)
↳ SHAP	0.3919 (↑)	0.4691 (↓)	0.5641 (↑)	0.5068 (↓)	0.3086 (↑)	0.3719 (↑)
BAE	0.4459 (↑)	0.5432 (↓)	0.7436 (↑)	0.5676 (↓)	0.3827 (↑)	0.4876 (↑)
BERT-Attack	0.4256 (↑)	0.5556 (↓)	0.6154 (↑)	0.6351 (↑)	0.4136 (↑)	0.4959 (↑)
SMART	0.5000 (↑)	0.5679 (↓)	0.5384 (↓)	0.4527 (↓)	0.3765 (↑)	0.3636 (-)

Table 3: Adversarial testing results on the adversarial GLUE tasks (accuracy). Best values are highlighted and arrows represent an increase or a decrease compared to the baseline, respectively.

Method	SST2	RTE	QQP	QNLI
Baseline	0.9037	0.5884	0.9027	0.8814
XAI-A.	0.9025	0.5704	0.8917	0.8706

Table 4: Adversarial training of XAI-Attack (LIME and indicator filtering) on standard GLUE tasks (accuracy).

In contrast to the adversarial testing experiment, we split the training data for each task to obtain a hold-out set of 10%. XAI-Attack then creates adversarial examples using LIME and indicator filtering. DistilBERT-base is subsequently trained using both the full training data and additional adversarial examples, and then compared to a model trained only with the full training data.

Table 4 displays the results for the validation sets of the tasks. It can be observed that both are quite similar, with an expected smaller decrease as XAI-Attack identifies biases in the models. These biases may originate from the training data and could also be present in the validation data. To further investigate the cause of the decrease in performance, we conduct a qualitative analysis of the adversarial examples in the SST2 sentiment task. Our primary focus is on identifying adversarial words that alter the prediction of most correctly classified instances. These words serve as a clear indicator of a particular class. Words such as "bored," "silly," "charmless," "insightful," and "terrific" are then disregarded due to their clear semantic relevance to sentiment classification. Our investigation highlights words that are overrepresented in both the training and validation datasets, despite their semantic neutrality. For instance, the word "strain" was found to shift 1511 positive instances to negative. This shift occurs because the classifier had erroneously learned to associate "strain" with the negative class, evidenced by its presence in 70 negative class instances compared to just 36 in the positive class within the training data. The benchmark problem then arises because the validation set includes three examples of the negative class and no examples of the positive class. A classifier trained using examples from XAI-Attack or other adversarial methods may, at best, perform equally well in this regard, even if it has overcome bias and learned more complex rules. Additional examples of this phenomenon are detailed in Table 5.

Word	Label Transf.	#Adv. Ex.	Train Repr. 0 - 1	Val. Repr. 0 - 1
strain	1 ⇒ 0	1511	70 - 36	3 - 0
pedestrian	1 ⇒ 0	778	13 - 3	1 - 0
slaps	1 ⇒ 0	849	52 - 23	2 - 0
earnest	0 ⇒ 1	397	42 - 96	0 - 3
innocence	0 ⇒ 1	346	17 - 28	0 - 1
provides	0 ⇒ 1	260	3 - 63	0 - 3

Table 5: Adversarial words (LIME) from the SST2 dataset that are overrepresented in the train and validation set. Train and validation representations are of the form negative (0) - positive (1).

4.6. Adversarial Transfer

In this section we want to test whether the adversarial examples of one model are transferable to other models, i.e. also valid adversarial examples for other models. To do this, on the original GLUE benchmark, we take the adversarial examples generated by XAI-Attack with LIME and indicator filtering on the distilBERT-base model and test whether they can also fool another distilBERT-base model, a BERT-base model and a RoBERTa-base model trained with the same data.

The results of this experiment are shown in Table 6. It is clear that all of the models can be deceived to a large extent and that the adversarial examples are not only valid for a specifically trained model. The distilBERT model remains the one with the lowest results, which was to be expected as it is the same model type. The much more compre-

Dataset	distilBERT	BERT	RoBERTa
SST2	0.1940	0.3421	0.6178
RTE	0.6136	0.6877	0.4293
QQP	0.3299	0.4768	0.7653
QNLI	0.3864	0.6090	0.6962
MNLI-mm	0.2709	0.3670	0.6061
MNLI-m	0.2672	0.3797	0.3797

Table 6: Accuracy results of the adversarial transfer experiment. Adversarial examples (XAI-Attack with LIME and indicator filtering) of a distilBERT model are tested with another distilBERT, a BERT and a RoBERTa model.

hensive BERT model is still very susceptible to the adversarial examples, which is reflected in the low accuracy results. The RoBERTa model, which has been trained for much longer and with much more data, performs slightly better. In the QQP task, it even achieves an acceptable result of 0.7653. In the other tasks, it is still well below the results of the human evaluation. This indicates that XAI-Attack is able to find biases in datasets that are adopted by all transformer models.

4.7. Summary of the Results

In the human evaluation, the instances of the original datasets and the adversarial examples are labelled and some cases are analyzed in more detail. It shows that the generated adversarial examples are mostly not semantics-changing with regard to the label, i.e. valid adversarial examples. From this, it can also be inferred that the adversarial examples are of high value for adversarial training. To underpin this, we propose an experiment in which we utilize the Adversarial GLUE benchmark (Wang et al., 2022) to test whether models trained with different adversarial examples are more robust against other adversarial attacks. The results show that adversarial training with XAI-Attack improves the robustness considerably even compared to state-of-the-art word-level adversarial attacks and training methods, such as BERT-Attack, BAE, and SMART. We also test different filtering strategies, of which the absence of filtering achieves the highest improvements and the indicator filtering achieves consistent improvements. Furthermore, the evaluation results of random insertion show that XAI-Attack's standard method of appending the adversarial words at the beginning can be improved (more on this in the Limitations). The adversarial training experiments verify that common benchmarks have biases that are present in all task sets and exemplify the need for more out-of-distribution evaluations. In the last experiment we demonstrate that adversarial examples created for one model are transferable to other models, showing that transformer-based models are generally susceptible to XAI-Attack examples.

5. Conclusion

Adversarial examples are of great importance in all fields, as they show the flaws of a model and can even be used to attack a system. In this study, a new adversarial example method using XAI is proposed. The method was evaluated in several experiments consisting of a human evaluation, a novel method to assess robustness and a transferability evaluation. These experiments show high quality adversarial examples, significant improvements in the robustness of the models and strong transferability to larger models.

5.1. Findings

Besides the apparent contributions of an adversarial example creation/training method based on XAI and a novel way of assessing robustness of adversarial training, by using a specialized adversarial example benchmark, this study also revealed more obscure findings.

Combining XAI and adversarial example research results in two innovations: On the one hand, the use of XAI methods allows for more sophisticated importance scores, a wider function space and fewer constraints on the model (no soft labels required) than the importance functions currently used. On the other hand, focusing on mislearned textual cues by explaining incorrectly predicted instances has a much higher success rate and ultimately the greatest learning effect when combined with adversarial training.

Additionally, this revealed novel insights of **learned biases of transformer models**. One might tend to overestimate the performance of these pre-trained models, as they score very high on common NLP benchmarks. However, inspection of the adversarial examples produced by our method shows that the trained models often make their predictions based on one word only, and do not produce more complex rules, also known as shortcut learning (Geirhos et al., 2020). While showing how fragile a trained model can be, this also shifts the light towards the benchmarks. Due to the high performance of the models on the test data, but the heavy reliance on individual words without semantics in relation to the label, it is evident that the **common test datasets have biases** which are also visible in the training and development data. While Section 4.5 provides even more evidence for this, we expect further research on this topic. An important implication for practical systems is, that one should always ensure that the validation and test sets should strictly represent the real data (involving constant re-evaluations).

Limitations

Our study encountered several limitations, in light of which we identify avenues for future research that extend and deepen the understanding of our results. For example, we have not tried XAI-Attack on larger transformer models such as GPT-3 (Brown et al., 2020). These models are expected to be much more robust against attacks, but since they can also be brittle in terms of the right prompting, we could imagine XAI-Attack finding adversarial examples for them as well. However, research into smaller language models remains important because, for example, they can be used on one's own hardware, are easier to interpret, can be easily fine-tuned, do not hallucinate and could be just as good or better in certain areas.

With regard to XAI-Attack itself, we would like to emphasise that it requires a hold-out set, which can be the development set or part of the training set. Further research could address the question of how much data is needed for the hold-out set and how the size affects adversarial training and ultimately the robustness of models. This can also be important for the question of how well XAI-Attack generalizes. The adversarial examples in the Adversarial GLUE benchmark are crafted using very diverse methods, resulting in all kinds of adversarial examples. The significant performance increase from adversarial training on these examples is a first indication that the model has indeed become more generally robust. However, we believe that this generalisability depends on how many adversarial examples are found and used for training. With very few adversarial examples, the model might only unlearn the biases for the specific adversarial words used in training. With enough adversarial examples (like in the experiments in the paper), on the other hand, the model tends to become more robust in general.

In its standard implementation, XAI-Attack inserts the adversarial words only at the beginning of the instances, which has already led to good results. However, in our experiments we also looked at inserting the words at a random position, which actually improved the results in one task significantly (see Section 4.4). Hence, it could be very interesting for further studies to investigate how the words could be inserted in a more informed way (e.g. by using language models to predict the most appropriate and coherent insertion). With this, the instances generated could also be more imperceptible as adversarial examples, which some works, such as Wang et al. (2022a), consider part of the definition of an adversarial example. Regarding the adversarial training, another direction would be to remove the identified adversarial words of some of the training instances, which could lead to more complex decision rules of the model as long as it is ensured that the label does not change. Taking this this idea even further, it might be interesting to let generative models create instances based on the adversarial words.

In our human evaluation experiments, the annotators' labelling performance on the normal dataset was not perfect. This may be partly due to the fact that our annotators are fluent in English but not native speakers. Furthermore, the results are comparable to the study by Nangia and Bowman (2019), who also performed human evaluations on the GLUE benchmark, showing that the tasks are not as easy as they might seem at first glance.

Finally, while we are confident that XAI-Attack will work well with other languages, we have only experimented with English examples. We look forward to adversarial research with other languages.

Ethics Statement

In our work, we have consistently prioritized ethics, continuously reassessing our approach to ensure responsible conduct. Research in the field of adversarial example generation inherently poses risks of misuse, notably in the form of attacks on machine learning models. This risk is especially pronounced in safety-critical domains, as we have outlined in the beginning. While our method could be repurposed for malicious use, we want to emphasize that the primary goal of this work is to generate adversarial examples that can be used to fortify machine learning models against such and other attacks, as we have demonstrated with the adversarial testing experiment in Section 4.4.

In addition, we advise reviewing the adversarial examples generated by XAI-Attack. This applies to any adversarial example method, as the examples may introduce new unwanted biases.

6. Acknowledgements

This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE and by the German Federal Ministry for Education and Research (BMBF) in the projects CYWARN (13N15407) and CYLENCE (13N16636). The calculations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt.

7. Bibliographical References

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2023. A Survey on Data Augmentation for Text Classification. *ACM Comput. Surv.*, 55(7):146:1–146:39.

Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognit.*, 84:317–331.

Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial Patch. *CoRR*, abs/1712.09665.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Antonio Emanuele Cinà, Kathrin Grosse, Ambra Demontis, Sebastiano Vascon, Werner Zellinger, Bernhard Alois Moser, Alina Oprea, Battista Biggio, Marcello Pelillo, and Fabio Roli. 2023. Wild Patterns Reloaded: A Survey of Machine Learning Security against Training Data Poisoning. *ACM Comput. Surv.*, 55(13s):294:1–294:39.

Patrick L. Combettes and Jean-Christophe Pesquet. 2011. Proximal Splitting Methods in Signal Processing. In Heinz H. Bauschke, Regina Sandra Burachik, Patrick L. Combettes, Veit Elser, D. Russell Luke, and Henry Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, volume 49 of *Springer Optimization and Its Applications*, pages 185–212. Springer.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018a. On Adversarial Examples for Character-level Neural Machine Translation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 653–663. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018b. HotFlip: White-box Adversarial Examples for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based Adversarial Examples for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6174–6181. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence*, 2(11):665–673. ArXiv:2004.07780 [cs, q-bio].

Tom Goldstein, Christoph Studer, and Richard G. Baraniuk. 2014. A Field Guide to Forward-backward Splitting with a FASTA Implementation. *CoRR*, abs/1411.3406.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A Survey of Adversarial Defenses and Robustness in NLP. *ACM Comput. Surv.*, 55(14s):332:1–332:39.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1875–1885. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2021–2031. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and Efficient Fine-tuning for Pre-trained Natural Language Models through Principled Regularized Optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2177–2190. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159.

Ang Li, Fangyuan Zhang, Shuangjiao Li, Tianhua Chen, Pan Su, and Hongtao Wang. 2023. Efficiently generating sentence-level textual adversarial examples with Seq2seq Stacked Autoencoder. *Expert Syst. Appl.*, 213(Part):119170.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6193–6202. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial Training for Large Neural Language Models. *CoRR*, abs/2004.08994.

Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial Training Methods for Semi-supervised Text Classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. 2016. Distributional Smoothing with Virtual Adversarial Training. ArXiv:1507.00677 [cs, stat].

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 119–126. Association for Computational Linguistics.

Nikita Nangia and Samuel R. Bowman. 2019. Human vs. Muppet: A Conservative Estimate of Human Performance on the GLUE Benchmark. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4566–4575. Association for Computational Linguistics.

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, Rujun Long, and Patrick McDaniel. 2018. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. ArXiv:1610.00768 [cs, stat].

Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 506–519. ACM.

Mohammadreza Qaraei and Rohit Babbar. 2022. Adversarial examples for extreme multilabel text classification. *Mach. Learn.*, 111(12):4539–4563.

Carl Edward Rasmussen and Zoubin Ghahramani. 2000. Occam's Razor. In *Advances in Neural Information Processing Systems 13, Papers from*

Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA, pages 294–300. MIT Press.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. 2018. Foolbox: A Python toolbox to benchmark the robustness of machine learning models. ArXiv:1707.04131 [cs, stat].

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3353–3364.

Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. 2022. Robust Text CAPTCHAs Using Adversarial Examples. In *IEEE International Conference on Big Data, Big Data 2022, Osaka, Japan, December 17-20, 2022*, pages 1495–1504. IEEE.

Junshuai Song, Jiangshan Zhang, Jifeng Zhu, Mengyun Tang, and Yong Yang. 2022. TRAttack": " Text Rewriting Attack Against Text Retrieval. In *Proceedings of the 7th Workshop on Representation Learning for NLP, RepL4NLP at ACL 2022, Dublin, Ireland, May 26, 2022*, pages 191–203. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

James Thorne and Andreas Vlachos. 2019. Adversarial attacks against Fact Extraction and VERification. *CoRR*, abs/1903.05543.

Juncheng Wan, Jian Yang, Shuming Ma, Dongdong Zhang, Weinan Zhang, Yong Yu, and Zhoujun Li. 2022. PAEG: Phrase-level Adversarial Example Generation for Neural Machine Translation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 5085–5097. International Committee on Computational Linguistics.

Wenqi Wang, Lina Wang, Run Wang, Aoshuang Ye, and Jianpeng Ke. 2022a. Better constraints of imperceptibility, better adversarial examples in the text. *Int. J. Intell. Syst.*, 37(6):3440–3459.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022b. Measure and Improve Robustness in NLP Models: A Survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4569–4586. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR*, abs/1910.03771.

Muchao Ye, Chenglin Miao, Ting Wang, and Fenglong Ma. 2022. TextHoaxer: Budgeted Hard-label Adversarial Attacks on Text. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 3877–3884. AAAI Press.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. 2019. You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 227–238.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

8. Language Resource References

Wang, Alex and Singh, Amanpreet and Michael, Julian and Hill, Felix and Levy, Omer and Bowman, Samuel. 2018. *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. Association for Computational Linguistics. PID https://gluebenchmark.com/.

Wang, Boxin and Xu, Chejian and Wang, Shuohang and Gan, Zhe and Cheng, Yu and Gao, Jianfeng and Awadallah, Ahmed Hassan and Li, Bo. 2022. *Adversarial GLUE: A Multi-Task Benchmark for Robustness Evaluation of Language Models*. arXiv. PID https://adversarialglue.github.io/. ArXiv:2111.02840 [cs].