# Finding a Needle in the Adversarial Haystack: A Targeted Paraphrasing Approach For Uncovering Edge Cases with Minimal Distribution Distortion

**This paper contains real-world cases which are offensive/hateful in nature.**

**Aly M. Kassem  Sherif Saad**
School of Computer Science, University of Windsor
{kassem6,sherif.saad}@uwindsor.ca

## Abstract

Adversarial attacks against Language models (LMs) are a significant concern. In particular, adversarial samples exploit the model's sensitivity to small input changes. While these changes appear insignificant on the semantics of the input sample, they result in significant decay in model performance. In this paper, we propose Targeted Paraphrasing via RL (TPRL), an approach to automatically learn a policy to generate challenging samples that improve the model's performance. TPRL leverages FLAN-T5, a language model, as a generator and employs a self-learned policy using a proximal policy optimization to generate the adversarial examples automatically. TPRL's reward is based on the confusion induced in the classifier, preserving the original text meaning through a Mutual Implication score. We demonstrate & evaluate TPRL's effectiveness in discovering natural adversarial attacks and improving model performance through extensive experiments on four diverse NLP classification tasks via Automatic & Human evaluation. TPRL outperforms strong baselines, exhibits generalizability across classifiers and datasets, and combines the strengths of language modeling and reinforcement learning to generate diverse and influential adversarial examples.

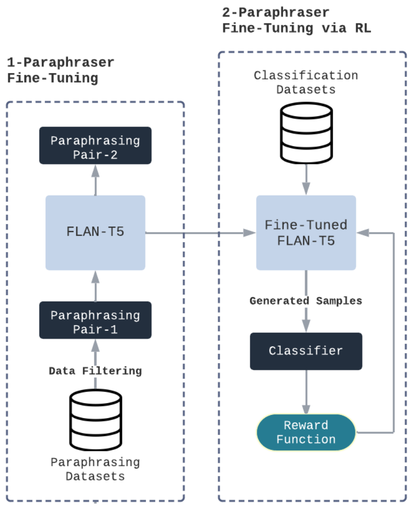



Figure 1: Components of our framework **TPRL** for Natural Adversarial Generation. (1) Employing Data filtering and then paraphrasing fine-tuning. (2) Targetted paraphrasing through employing RL on classification Datasets.

## 1 Introduction

LMs have made impressive advancements in classification, question-answering, and machine translation. However, they are susceptible to adversarial attacks, which exploit their vulnerability to small input changes (Jia and Liang, 2017; Jin et al., 2020; Alzantot et al., 2018; Wallace et al., 2019; Jia et al., 2019; Cheng et al., 2019). These attacks introduce variations not encountered during training. Two approaches to address these vulnerabilities are data augmentation-based techniques (Liu et al., 2020; Wang and Yang, 2015; Kobayashi, 2018; Yu et al., 2018) and adversarial training-based approaches (Zhu et al., 2019; Yoo and Qi, 2021; Xu et al., 2019). Expanding training data using predesigned samples generated by data augmentation methods can assist classifier training. However, generated samples may lack an adversarial nature (Altinisik et al., 2022), leading to confusion and inaccurate classification. Adversarial training-based approaches (Xu et al., 2019; Le et al., 2022; Deng et al., 2022; Iyyer et al., 2018; Alzantot et al., 2018) address this limitation by generating challenging examples. This improves the model's ability to handle difficult subsets of data, enhancing robustness and performance.

Generating diverse and semantically meaningful adversarial examples is challenging due to limited operations like word addition, deletion, or substitution. This lack of diversity in word-level generation often results in generated sentences with identical vectors to the original, offering little insight into the model's behavior.



*Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 552–572
March 17-22, 2024 

Recent studies (Le et al., 2022; Zhao et al., 2018) show that character manipulation or word swap methods can produce irrelevant and incoherent samples, altering the original text's meaning. These methods are unsuitable for real-world applications and can harm the model. Attacks lacking semantic significance expose the model's blind spots, are easily detectable and removable, and do not represent real-world examples encountered during deployment.

To address this, researchers have explored sentence-level generation methods. GAN-based approaches (Zhao et al., 2018; Wang et al., 2020a) show promising results in generating diverse examples but may result in irrelevant and obtaining labeled data is challenging. Another approach employs a paraphrasing technique using LMs. However, this method lacks targeted fine-tuning to break the classifier and has limited available styles. In this paper, we introduce a novel approach called TPRL that leverages FLAN-T5 for targeted paraphrasing, enhancing classifier performance through Reinforcement Learning. We employ a two-step process: first, we train a diverse paraphrasing model using FLAN-T5 and then fine-tune it with RL to preserve text meaning while generating targeted adversarial examples. We evaluate TPRL on four classification tasks, both automatically and with human evaluation, demonstrating its ability to improve classifier robustness and effectiveness. Our findings can be summarized as follows:

- Utilizing the generated examples for adversarial training improves classifier performance on original and adversarial test sets.
- Our work demonstrates that the learned policy for one classifier is universal and can be generalized to unseen classifiers in the same dataset.
- Experiments show that TPRL outperforms strong baselines and improves results on various models and datasets.

## 2 Background

This section briefly introduces and formalizes textual adversarial attacks for text classification and employs RL in language models for generating adversarial attacks and other tasks.

### 2.1 Textual Adversarial Attacks

Generating adversarial attacks against NLP models is more challenging than for vision models (Qiu et al., 2022). NLP models rely on discrete word representations, where even slight adjustments can drastically change the meaning or validity of a phrase. Unlike images, NLP models require a deeper understanding of context and language structure, making successful attacks difficult.

**Attacks Properties.** For a victim classification model, denoted as $F_\theta$, tested on dataset $D_t$ with samples $(x_t, y_t)$, "an adversarial attacker aims to perturb $x_t$ to maintain semantic similarity to humans but destroy its meaning when classified by the model." This generates an adversarial example, $x'_t$, that the model misclassifies.

### 2.2 RL In Language Models

Dynamically generating adversarial attacks at the sentence level is more compatible with the reinforcement learning (RL) paradigm. In the realm of NLP, RL has gained prominence for addressing undesirable behavior, including toxicity, social biases, offensive speech, and data memorization(Paulus et al., 2017; Rennie et al., 2017; Wu et al., 2016; Kassem et al., 2023). This is accomplished by using Proximal Policy Optimization (PPO) (Schulman et al., 2017) to optimize a Language Model (LLM) based on a reward model. Despite RL's success in mitigating undesirable behavior in LMs, its potential for generating adversarial attacks remains largely unexplored. This paper presents the first investigation of using RL with a language model for generating natural adversarial attacks.

## 3 Collecting Labeled Paraphrasing Pairs

This section will show the selected datasets for training the paraphrase model. Afterward, we will outline a systematic procedure for filtering those datasets to maximize the paraphrase pairs' diversity, similarity, and relevance.

### 3.1 Paraphrasing Datasets

In our initial stage, we gathered seven diverse paraphrase datasets, most undergoing thorough human judgment annotation to ensure high-quality paraphrasing examples. This comprehensive paraphrasing corpus comprises data from the APT dataset (Nighojkar and Licato, 2021), Microsoft Research Paraphrase Corpus (MSRP) (Dolan and Brockett, 2005), PARANMT-50M corpus (Wieting and Gimpel, 2018), TwitterPPDB (Lan et al., 2017), PIT-2015 (Xu et al., 2015), PARADE (He et al., 2020), and QQP (Iyer et al., 2017). We utilized a filtered version of the PARANMT-50M corpus as recommended by (Krishna et al., 2020). We retained sentence pairs with 4, 5, and 6 similarity labels from TwitterPPDB and sentences with semantic similarity labels of 5 and 4 from PIT-2015. For QQP, we selected samples labeled as duplicates. The merged dataset, totaling 560,550 samples, underwent filtering to ensure high-quality similarity and diversity for the subsequent stage.

### 3.2 Improving Diversity & Relevance Via Data Filtering

In the second stage, we choose training data for the paraphrase model, following (Krishna et al., 2020). Despite the availability of human annotations, it is still possible for noise and irrelevant samples to exist in the dataset. We employ aggressive filtering with four filters to address noise and irrelevant samples in the dataset. Firstly, we remove sentence pairs with over 50% unigram overlap, ensuring lexical diversity computed using SQUAD evaluation scripts based on the F1 score (Rajpurkar et al., 2016). Secondly, we discard pairs with less than 50% reordering of shared words, promoting syntactic diversity measured by Kendall's tau (Kendall, 1938). Thirdly, we eliminate pairs with less than 50% semantic similarity, measured by cosine similarity using the "all-MiniLM-L12-v2" model (Wang et al., 2020b; Reimers and Gurevych, 2019). Finally, we remove sentences with over 70% trigram overlap to improve diversity further. After applying these filters, the refined dataset contains 96,073 samples, split into training (76,857 samples), validation (9,608 samples), and testing (9,608 samples) sets. These filters ensure a diverse and representative sample for effective training and evaluation.

## 4 Targeted-Paraphrasing Via RL

Figure 1 illustrates the framework's structure. After filtering data for diverse and relevant paraphrase pairs, we proceed with the initial fine-tuning of the model. Subsequently, we utilize proximal policy optimization, a reinforcement learning technique, to fine-tune the model further. This approach generates paraphrases that exploit the classifier's weaknesses, resulting in complex and effective adversarial samples.

### 4.1 Paraphraser Model

We enhance the FLAN-5-large model by fine-tuning the filtered data for nine epochs. Employing the BERT-Score metric (Zhang et al., 2019), it achieves an F1-score of 75.925%, enhancing fluency, diversity, relevance, and paraphrasing ability. The LM also addresses task-irrelevant generation issues. Training on relevant pairs maximizes task-specific outputs. Utilizing the LM boosts paraphrasing capability and introduces new information about entities or objects in the input text, improving generation quality (Petroni et al., 2019). This optimized paraphrase is then used to create adversarial training samples.

### 4.2 Fine-Tuning Paraphraser Via RL

After the initial fine-tuning, the paraphraser model can generate various relevant and fluent paraphrases. To enhance its performance, our approach includes a guiding component via the reward function. This involves further fine-tuning the model using reinforcement learning (PPO), enabling it to produce targeted adversarial examples that confuse the classifier.

#### 4.2.1 Reward Function

Given tokens $x_{<t} = \{x_0, x_1, \ldots, x_{t-1}\}$ and accumulated hidden states $h_{\theta<t}$ before time step $t$. An auto-regressive language model (LM) is trained to maximize the probability of the next step token $\hat{x}_t$. LM as a generator $G$ selects the token that has the highest probability $x_t$ as the $t$-th step decoding output:

$$x_t \sim \text{argmax}_{\hat{x}_t} p(\hat{x}_t | x_{<t}) = \text{G}(x_{<t}, h_{\theta<t}) \quad (1)$$

In the reinforcement learning framework, we define the state at step *t* as all the sequences generated before t $s_t = x_{<t}$, and the action at step t as the t-th output token ($a_t = x_t$). The policy $\pi_\theta$ represents the probability of selecting token $x_t$ (action $a_t$) given the preceding state $s_t = x_{<t}$. This probability is derived from the softmax output of the hidden states $\pi_\theta(a_t|s_t) = \text{softmax}(h_{\theta<t})$, and this interpretation extends to the conditional case as well. The single-step reward for token $x_t^c$ at step $t$ can defined as follows:

$$R(x^c{}_t) = E_t \left[ \frac{\pi_{\theta_c}(a_t|s_t)}{\pi_\theta(a_t|s_t)} r(x^c{}_t) \right] - \beta \text{KL}(\theta || \theta_c) \quad (2)$$

Where $r(x^c{}_t)$ is the objective composed of the weighted sum of confusion and Mutual Implication. The KL penalty is applied per token using a reference model, which is the original model that does not receive the signal reward to prevent significant deviations.

**Confusion constraint with classifier.** To generate confusing samples that challenge the model, we randomly select pairs $(x, y)$ from the dataset $D$ and input them into the generator $G$. This process yields a novel pair $(\hat{x}, y)$ through generation. Subsequently, we fed these generated instances $(\hat{x}, y)$ into the classifier $C$ to obtain the likelihood of the true label $L(y|\hat{x}) = p_C(y|\hat{x}; \psi)$. Where $\psi$ represents the parameter of the classifier. We then incorporate $(1 - L(y|\hat{x}))$ into our reward function. This term aims to incentivize the generator to acquire a policy that produces confusing samples capable of decreasing the classifier's confidence. It can be viewed as searching for confusing samples within the classifier's space.

**Similarity constraint with Mutual Implication.** Previous research in word and sentence-level attacks often used embeddings like word2vec (Mikolov et al., 2013), and GloVe (Pennington et al., 2014), along with contextual embeddings such as BERTScore (Zhang et al., 2019) and BLEURT (Sellam et al., 2020). However, these methods often fail to capture inferential semantics, leading to irrelevant sample generation when used as a reward function. For example, BERTScore assigns a high similarity score to sentences like "I bought an iPad" and "I bought a Laptop," even though they refer to different products.

To address this, we propose an alternative approach: using Mutual Implication (Nighojkar and Licato, 2021), a similarity score that more accurately evaluates text generation by considering both relevance and semantic equivalence as the bi-directional relationship adds more constraints between the generated and original sentence. Let $A = a1, \ldots, an$ and $B = b1, \ldots, bm$ be two token sequences, and let $L_{NLI}$ be a language model trained on the Natural Language inference task (NLI) that produces three labels, Entailment, Contradiction, and Neutral. We estimate the Mutual Implication as follows:

$$\mathrm{MI}(A, B) = [L_{NLI}(A \Rightarrow B) + L_{NLI}(B \Rightarrow A)]/2. \tag{3}$$

Where $L_{NLI}$ is ALBERT XXLarge $v_2$ model (Lan et al., 2019) trained on various NLI datasets (Nie et al., 2019; Bowman et al., 2015; Williams et al., 2018; Nie et al., 2019, 2020).

**KL Penalty.** The policy of the fine-tuned model may deviate significantly from the old policy (the model before fine-tuning), potentially leading to a less coherent and relevant generation. To address this issue, we introduce a KL divergence penalty term to quantify the dissimilarity between these two policies. This step helps ensure that our optimization process remains within a trustworthy region. The KL divergence, calculated for the policies, is expressed as:

$$KL(\theta||\theta_c) = \sum_{i \in [1,t]} \pi_\theta(a_i|s_i) \cdot \log \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_c}(a_i|s_i)} \tag{4}$$

We deduct KL divergence with default value weight $\beta = 0.2$ as a penalty term in the reward function (Equation 2).

**Objective function.** We consolidate all constraints from previous sections into a unified objective. Our objective function blends the negative likelihood, which drives the generator to produce confusing samples, with the mutual implication, which ensures the similarity, naturalness of meaning, and semantic equivalence. This measures how closely the paraphrases convey the same meaning as the original text and vice versa. The final objective function is defined as:

$$r(\hat{a}_t) = \upsilon \cdot (1 - p_C(y|\hat{x}; \psi)) + \alpha \cdot MI(x, \hat{x}) \tag{5}$$

Here, $\psi$ represents the classifier's parameter, and $\upsilon$ as well as $\alpha$ serve as weighting factors, both set to 0.5. This balance ensures that the model neither generates overly confusing examples at the expense of relevance nor produces irrelevant results at the cost of coherence.

#### 4.2.2 Generation Settings

We applied several constraints to sentence generation, such as matching the generated sentence length to the original one to avoid irrelevant tokens. We use early stopping when enough complete candidate beams are available to prevent meaningless or off-topic tokens. These constraints enable the generator to introduce new, relevant information or replace existing tokens with a diverse range of novel and pertinent tokens. We also generate ten alternative samples and assess their adequacy and fluency using the Parrot Tool (Damodaran, 2021), selecting the most suitable one for reward computation to maximize fluency and relevance.

#### 4.2.3 Policy Optimization

We use a proximal policy optimization approach with top-p sampling 0.95, known as Natural Language Policy Optimization (Ramamurthy et al., 2022)(see Appendix A for more details). Given the reward and the definitions described above, we update our policy at $t$-th step as:

$$\theta_{\text{new}} = argmax_{\theta}\mathbb{E}\left[\min\left(r_t(\theta), \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\right) A_t\right] \tag{6}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$. The optimization objective is to find the new policy parameters that maximize expected rewards while keeping the policy update bounded within a certain range defined by the clipping parameter. This helps maintain stability during training. PPO also balances the trade-off between exploration and exploitation by encouraging actions that have higher estimated advantages while avoiding drastic policy changes that could disrupt learning. Further details on PPO & NLPO methods can be found in Appendix A. The model was trained for thirty epochs with a batch size of 32 and optimized using the Lion optimizer (Chen et al., 2023), using a learning rate of $4.9 \times 10^{-6}$; see Appendix F for more details about the hyperparameters.

### 4.3 Sampling Adversarial Texts

Once the paraphraser has been optimized vial RL, we can sample from the adversarial distribution to construct adversarial examples. We generate counterparts for each sample in the original training set to create adversarial samples. However, it is possible that some samples are irrelevant or not useful for training (Xu et al., 2019). Thus, we exclude those with a mutual implication score below 50%, ensuring that only semantically equivalent samples are included. Finally, These samples are added to the training set, creating an updated dataset. Using the same random seed, we train a new classifier from scratch, employing adversarial training.

For the classifiers, we began by fine-tuning the classification models (subsubsection 5.1.2) on the respective dataset (subsubsection 5.1.1) and reported their performance. Each model achieved a different performance and made different errors. We then leveraged these classifiers within the RL feedback loop.

## 5 Experiments

We assess the effectiveness of targeted paraphrasing via RL adversarial attacks (TPRL) on four distinct classification tasks: sentiment analysis, news topic classification, hate speech detection, and offensive speech detection. To this end, we carefully select relevant datasets, outline our implementation details, establish baseline methods, and specify evaluation metrics.

### 5.1 Experimental Settings

#### 5.1.1 Tasks & Datasets.

**Sentiment Analysis.** *SST-2 & SST-5* datasets for sentiment analysis in movie reviews from the Stanford Sentiment Treebank (Socher et al., 2013). SST-2 (N=6920)has binary sentiment labels (positive or negative), while SST-5 (N=8540) has more fine-grained sentiment labels (very positive, positive, neutral, negative, and very negative) with an average of 19 words per sample.

**New Topic Classification.** *AG News* dataset (Zhang et al., 2015) with a number of samples 120,000, categorizing news articles into four classes: World, Sports, Business, and Science/Technology, with an average of 38 words per sample.

**Offensive Speech Detection.** *SemEval2019* Task 6 (OffensEval) dataset (N=11916) (Zampieri et al., 2019) for offensive detection in tweets has binary classes: offensive and non-offensive tweets with an average of 19 words per sample.

**Hate Speech Detection.** *Hate speech* dataset (N=7071), a collection of sentences extracted from Stormfront, a white supremacist forum (de Gibert et al., 2018). Based on their content, sentences are categorized into two different classes, HATE and NoHate, with an average of 16 words per sample.

We eliminated all punctuation, mentions, hashtags, and URL links from the samples of all datasets. Furthermore, we employed lowercase for all samples. The maximum sequence length was implemented as the maximum length parameter in all BERT models. The training, validation, and test sets officially released by the creator of the datasets were utilized.

#### 5.1.2 Victim Models.

To assess the effectiveness of our approach across different models, we selected five popular pre-trained language models: *BERT-base*, *BERT-large* (Devlin et al., 2018), *RoBERTa-base*, RoBERTa-large (Liu et al., 2019), and DeBERTa-v3-large (He et al., 2021), which vary in architecture and size.

| Classifier | SST2 | SST5 | AG's News | HS‡ | OFF‡ |
|---|---|---|---|---|---|
| BERT$_{Base}$ | 90.88 | 53.52 | 93.97 | ⋆ | 84.76 |
| *+SCPN* | 89.67 | 51.71 | 93.26 | ⋆ | 83.60 |
| *+StyAdv* | 87.91 | 52.35 | 93.19 | ⋆ | 81.86 |
| *+UNTP* | ⋆ | 51.90 | 94.17 | ⋆ | ⋆ |
| ***+TPRL*** | **91.15** | 52.39 | **94.46** | ⋆ | **85.11** |
| BERT$_{Large}$ | 91.43 | 53.52 | 94.18 | ⋆ | 85.11 |
| *+SCPN* | 90.66 | 53.61 | 93.17 | ⋆ | 72.09 |
| *+StyAdv* | 90.17 | 23.07 | 93.57 | ⋆ | 72.09 |
| *+UNTP* | ⋆ | 54.84 | **94.42** | ⋆ | ⋆ |
| ***+TPRL*** | **92.58** | **54.93** | 94.36 | ⋆ | **85.58** |
| RoBERTa$_{Base}$ | 94.34 | 54.79 | 93.78 | 91.90 | 83.95 |
| *+SCPN* | 92.31 | 53.89 | 93.61 | 91.30 | 82.67 |
| *+StyAdv* | 91.81 | 52.21 | 93.32 | 91.55 | 82.32 |
| *+UNTP* | ⋆ | **56.19** | 93.78 | 91.90 | ⋆ |
| ***+TPRL*** | 94.00 | 56.15 | **93.93** | **92.45** | **85.00** |
| RoBERTa$_{Large}$ | 93.73 | 58.30 | 93.92 | 92.45 | 85.93 |
| *+SCPN* | 49.91 | 23.07 | 93.80 | 92.30 | 72.093 |
| *+StyAdv* | 49.91 | 23.07 | 93.73 | 91.75 | 72.09 |
| *+UNTP* | ⋆ | ⋆ | 93.86 | 92.45 | ⋆ |
| ***+TPRL*** | **94.72** | **58.95** | **94.21** | 92.05 | **84.53** |
| DeBERTa-V3$_{Large}$ | 94.89 | 58.46 | 93.92 | 89.50 | 84.65 |
| *+SCPN* | 93.52 | **59.23** | 93.75 | 89.50 | 84.76 |
| *+StyAdv* | 93.41 | 55.42 | ⋆ | **92.95** | 80.93 |
| *+UNTP* | ⋆ | 58.09 | 93.85 | 89.50 | ⋆ |
| ***+TPRL*** | **95.82** | 58.77 | **94.22** | 92.30 | **85.93** |

Table 1: The classifier-dataset experiments for five classifiers. We show the accuracy results on the original test set before & after we apply AT with TPRL & the three baseline methods. ‡ Refer to Hate dataset, OFF to Offensive dataset. * Refer to not deployed experiments. The best comparable performances are bolded

| Classifier/Framework | SCPN (%) | StyleAdv (%) | UNTP (%) | TPRL (%) |
|---|---|---|---|---|
| **BERT$_{Base}$** | 69.37 | 30.62 | 64.43 | 34.84 |
| **+AT** | 68.70 | 54.28 | 66.24 | 42.28 |
| **BERT$_{Large}$** | 70.86 | 29.97 | 64.89 | 41.51 |
| **+AT** | 68.00 | 51.73 | 64.35 | 44.36 |
| **RoBERTa$_{Base}$** | 95.53 | 29.29 | 67.52 | 37.27 |
| **+AT** | 74.01 | 57.42 | 67.27 | 43.41 |
| **RoBERTa$_{Large}$** | 82.49 | 28.02 | 89.78 | 40.27 |
| **+AT** | 75.55 | 52.28 | 90.19 | 49.22 |
| **DeBERTa-v3$_{Large}$** | 74.77 | 36.79 | 63.38 | 34.66 |
| **+AT** | 75.01 | 45.33 | 64.83 | 47.82 |

Table 2: The classifier-framework experiments for five classifiers. We show the accuracy results on the adversarial test set before & after we apply AT with TPRL & the three baseline methods.

**Baseline Methods.** TPRL is compared to SCPN (Iyyer et al., 2018) and StyleAdv (Qi et al., 2021), other sentence-level adversarial attack techniques. SCPN uses a seq2seq Bi-directional LSTM (Hochreiter and Schmidhuber, 1997) with the PARANMT-50M corpus and syntactic templates. StyleAdv utilizes the STRAP model (Krishna et al., 2020) for style transfer, incorporating five distinct styles. Also, we considered an untargeted paraphrasing model (UNTP) without the guiding component. See Appendix B for implementation details.

## 5.2 Evaluation Metrics

We thoroughly evaluated TPRL's effectiveness in five key areas, ensuring the following:

**(1) Improving Performance:** We evaluated TPRL's impact on the accuracy of the original test set, alone and in combination with other methods, to assess its overall performance enhancement. Emphasizing that we follow a model-performance perspective, our objective is to find the model's weaknesses and enhance them, not just attack or break the model.

**(2) Fluency and Quality:** We assessed fluency using perplexity (PPL) from GPT-2-XL (Radford et al., 2019) and a RoBERTa-large classifier trained on the CoLA corpus (Warstadt et al., 2019), to overcome the limitations of perplexity in evaluating fluency as the model provides accurate grammatical acceptability judgments.

(3) **Semantic Similarity (SIM):** We assessed semantic similarity between input sentence and generated samples using the "all-MPNet-Base-v2" embedding-based SIM model (Song et al., 2020; Reimers and Gurevych, 2019), known for its performance on semantic textual similarity (STS) benchmark (Muennighoff et al., 2022). We also used the mutual implication (MI) metric to capture inferential semantics comprehensively, addressing the limitations of STS.

(4) **Validity Via Human Evaluation:** We conducted human evaluations to determine the percentage of samples that produced adversarial examples without altering the original label.

(5) **Validity Via GPT-3.5:** To overcome human evaluation cost, we employed GPT-3.5, which has demonstrated comparable or superior performance to crowd-workers in-text annotation tasks (Gilardi et al., 2023; Törnberg, 2023; Chiang et al., 2023). This assessment validated the credibility of TPRL-generated samples.

## 5.3 Experimental Results & Discussion

We conducted comprehensive experiments to answer the following three overarching questions regarding TPRL:

### 5.3.1 Does TPRL Enhance The Performance?

Table 1 shows the evaluation of TPRL's accuracy against the three baseline methods and vanilla models on different benchmark datasets. Overall, TPRL

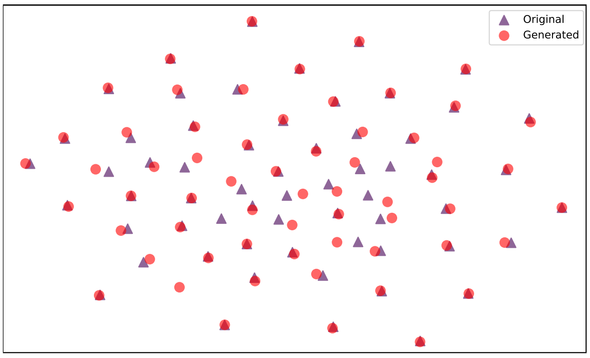



Figure 2: T-SNE visualization of the vectorized original and TPRL-adversarial sentences in the SST-2. The adversarial sentences (circles) mostly overlap with the original sentences (triangles), suggesting that generated sentences maintain the original class distribution.

| Classifier | Framework | PPL ↓ (%) | FL ↑ (%) | SIM ↑ (%) | MI ↑ (%) |
|---|---|---|---|---|---|
| $\text{BERT}_{\text{Base}}$ | SCPN | 567.66 | 50.57 | 74.49 | 80.58 |
| | StyleAdv | 670.47 | 57.65 | 74.62 | 58.34 |
| | UNTP | 498.61 | 85.20 | **77.33** | **92.41** |
| | TPRL | **368.41** | **87.1** | 73.56 | 89.9 |
| $\text{BERT}_{\text{Large}}$ | SCPN | 565.61 | 50.71 | 74.67 | 80.94 |
| | StyleAdv | 863.50 | 56.42 | 74.14 | 57.40 |
| | UNTP | 373.52 | 85.97 | **77.64** | **91.25** |
| | TPRL | **372.51** | **86.84** | 73.88 | 89.82 |
| $\text{RoBERTa}_{\text{Base}}$ | SCPN | 563.46 | 50.31 | 74.43 | 80.28 |
| | StyleAdv | 708.28 | 57.26 | 75.66 | 58.99 |
| | UNTP | **254.37** | 83.31 | **78.32** | 89.92 |
| | TPRL | 492.52 | **85.58** | 71.22 | **89.99** |
| $\text{RoBERTa}_{\text{Large}}$ | SCPN | 555.02 | 50.43 | 74.43 | 80.19 |
| | StyleAdv | 579.80 | 57.34 | 74.30 | 56.81 |
| | UNTP | **230.73** | 83.32 | 78.44 | 88.49 |
| | TPRL | 302.76 | **87.38** | **70.84** | **90.61** |
| $\text{DeBERTa-v3}_{\text{Large}}$ | SCPN | 560.70 | 50.29 | 74.28 | 79.91 |
| | StyleAdv | 845.76 | 57.69 | **74.92** | 57.46 |
| | UNTP | **372.44** | 73.21 | 68.86 | 73.14 |
| | TPRL | 393.06 | **87.27** | 73.90 | **89.67** |

Table 3: Automatic evaluation results showing the average of generated adversarial training samples of the five datasets across selected classifiers & baselines. The best comparable performances are bolded

consistently gained improved performance across the five classifiers and datasets. While UNTP outperformed SCPN and StyleAdv, it still lacked consistent effectiveness. In terms of the adversarial test set as shown in Table 3, SCPN did not enhance classifier performance, StyleAdv showed significant improvement but not consistently on the original test set, and UNTP had a small and sometimes non-existent improvement. However, TPRL achieved an ∼8% improvement on the adversarial test set while maintaining a balance in the tradeoff between the performance on both the original and adversarial sets. Note that the missing values in Table 1 because UNTP failed to meet the filtering criteria (<0.5 MI) for the SST2, Hate speech, and offensive speech datasets. Generator collapse[1] occurred during training for $\text{BERT}_{Base}$ in the hate speech dataset. For a fair comparison, AG's News with $\text{DeBERTa-V3}_{Large}$ was excluded due to low accuracy in adversarial training.

### 5.3.2 Does TPRL Generate Relevant Samples?

In an adversarial generation, maintaining topic and meaning while fooling the classifier is crucial since a sample can easily change the classifier's decision if the meaning changes. This concern is even more significant for sentence-level methods that create new sentences. We evaluated the generated samples using three approaches:

*(1) Automatic evaluation for similarity and fluency, including PPL for accurate fluency assessment. (2) Human evaluation for validity. (3) Visualization techniques for observing the geometric interpretation of samples.* Table 2 shows the results, with TPRL and UNTP achieving the lowest PPL and superior generation quality. TPRL also outperforms other baselines in fluency, ranging from 86% to 87%, across classifiers and datasets, thanks to including the MI score in the reward function to encourage natural sentence generation.

**Regarding The Relevance:** TPRL surpasses baseline methods in MI. For specific configurations, cosine similarity scores are low (∼74%), while MI scores are high (∼89%). This discrepancy arises because cosine similarity struggles to capture the inferential role accurately (further explanation given in section 6). See Appendix E for dataset-specific results.

**For Human evaluation:** Following (Qi et al., 2021), considering the cost, we conducted a validity evaluation on SST2. We randomly selected 100 adversarial samples for TPRL, SCPN, and StyleAdv (36, 33, and 31 samples, respectively). Each sample was evaluated by three annotators who determined if the sentiment matched the original example. The final decision was made by voting. The percentage of valid adversarial samples was TPRL 72%, SCPN 51.5%, and StyleAdv 32.2%. TPRL achieved the highest validity, confirming minimal distortion to the original distribution. See Appendix I for details about human evaluation.

**For Validity Via GPT-3.5.** To evaluate the similarity of larger generated samples, we employed GPT-3.5. Randomly choosing 100

[1]The language model-generator could not learn an effective policy to attack the classifier and produced such low reward values.

| Policy/Classifier | BERT$_{Base}$ (%) | BERT$_{Large}$ (%) | RoBERTa$_{Base}$ (%) | RoBERTa$_{Large}$ (%) | DeBERTa-v3$_{Large}$ (%) |
|---|---|---|---|---|---|
| None | 90.88 | 91.43 | 94.34 | 93.73 | 94.89 |
| Policy-BERT$_{Base}$ | 91.15 | 92.09 | 93.35 | $\star$ | 95.38 |
| Policy-BERT$_{Large}$ | 91.04 | 92.58 | **94.45** | 94.94 | **96.15** |
| Policy-RoBERTa$_{Base}$ | 90.06 | **93.24** | 94.0 | $\star$ | 95.60 |
| Policy-RoBERTa$_{Large}$ | **91.87** | 92.36 | $\star$ | 94.72 | 95.60 |
| Policy-DeBERTa-v3$_{Large}$ | 90.93 | **93.24** | 94.12 | **95.49** | 95.82 |

Table 4: Accuracy results of different classifiers trained with the examples generated by various attacking policies on the SST-2 dataset. Showing the universal policy. The best comparable performance policy for the classifier is bolded

samples from each framework in the SST-2 dataset, we rated them on a scale of 1 to 5, where 1 indicated significant dissimilarity, and 5 denoted substantial similarity. TPRL exhibited superior performance to the three baselines, receiving the highest similarity ratings across categories 5, 4, and 3. Further details can be found in Appendix G.

**For Visualization:** We randomly selected 60 samples from the SST-2 dataset and converted them into vectors using the Sentence-BERT 'all-MPNet-Base-v2' model. We used T-SNE (Van der Maaten and Hinton, 2008) to create a 2D representation of these vectors ((see Figure 2)). TPRL-generated samples closely resemble the original data, demonstrating that small changes in the semantic space can mislead the classifier, revealing vulnerabilities in LM-based classifiers. We consistently observed similar results across different datasets, classifiers, and random samples(refer to Appendix C for more details).

#### 5.3.3 Does TPRL Learned Attacking Policy Universal?

To investigate this question, we employed a fine-tuned generator to create samples targeting specific classifiers, like BERT-Base. Then, we fine-tuned another classifier, BERT-Large, using these generated samples to assess performance changes, repeating this process for five classifiers. Table 4 displays the outcomes in the SST-2 dataset(see Appendix D for the remaining datasets), revealing that most classifiers benefited from the samples generated by other classifiers, surpassing the naive baseline. This underscores the universality of the learned attacking policy. Notably, in certain instances, the improvement for the attacked classifier equaled that of the transferred classifiers despite each classifier having distinct errors prior to adversarial training(Appendix H).

## 6 Ablation & Analysis Studies

In this section, we conduct ablations and analyze TPRL's key components' impact on baseline improvements.

**TRPL Diversity Enhances Classifier Performance:** We explore diversity's impact on performance, comparing TPRL with the baseline methods. Similarity scores increased with reduced diversity post-adversarial training, while the MI metric sustains diversity. Human evaluations support MI's alignment. Also, we use GPT-3.5 to confirm TPRL's diversity; details in Appendix G.

**The Importance of Targeted Component**: Our TPRL implementation has a targeted component using RL. However, we evaluated using only the fine-tuned paraphraser without RL (UNTP), which showed inconsistent performance compared to TPRL (Table 1). Similarly, other baselines requiring a targeted component also yielded minor improvements. TPRL outperformed other models across most metrics (Table 3), emphasizing the importance of the targeted component.

**Number of Generated Samples Impact:** We explore the link between increasing sample size and the resultant performance improvements for both TPRL and a vanilla classifier. TPRL's average sample generation per dataset: SST-2: 359, SST-5: 2993, HS: 135, OFF: 407, AG's News: 3825. We examined the correlation between sample count (normalized by total set size) and TPRL's performance boost, finding no significant correlation (Pearson correlation of 0.144, $p \ll 0.05$). The insignificance stems from performance enhancements' variability due to classifier weaknesses. Thus, we introduced a Mutual Implication filtering criterion of 0.5 for selecting high-quality samples, improving model performance without irrelevant sampling.

| | |
|---|---|
| **Original:** i complain all the time about seeing the same ideas repeated in films over and over again, but the bourne identity proves that a fresh take is always possible. | |
| **Adversarial:** ive always been complaining that movies repeat the same ideas but the identity of bourne proves that you dont need to reinvent the wheel | |
| **Original:** something must be done to halt this terrible wave of immigrants entering britian | |
| **Adversarial:** in order to stop a terrible wave of immigration into britian it is necessary to do something | |
| **Original:** i live in oslo and race mixing is a huge problem | |
| **Adversarial:** theres a huge problem of race mixing in oslo | |
| **Original:** william shatner as a pompous professor is the sole bright spot | |
| **Adversarial:** the only bright spot of the movie is william shatner in a pompous professor | |
| **Original:** if you enjoy more thoughtful comedies with interesting conflicted characters; this one is for you | |
| **Adversarial:** this is a good comedy for those who like thoughtful comedies with interesting characters | |
| **Original:** any ways smearing a single dna steam with black dna is horrible | |
| **Adversarial:** any way it seems disgusting to smear one steam with black dna | |
| **Original:** we spend so much time on black people when the real problem is asians and hispanics | |
| **Adversarial:** we spend too much time and effort worrying about black while the asians and hispanics were the ones | |

Table 5: Comparing original & adversarial examples generated by TPRL.

**Sample adversarial texts.** Table 5 shows examples of generated samples by TPRL. TPRL introduces changes by paraphrasing the original sentence while preserving the original sentence's meaning and equivalence. Paraphrasing can be noticed in changing the writing style order of words and substituting or introducing new words simultaneously in one sentence, which is an advantage over word-level attacks that introduce limited operations for one sentence.

## 7 Conclusion

In this paper, we propose TPRL, an adversarial generation method designed to improve classification model robustness. Our innovation involves reinforcement training to learn various attack strategies automatically. We confirm TPRL's effectiveness in four classification tasks, consistently producing high-quality adversarial samples that represent edge cases with minimal distortion in the data distribution.

## Limitations

One limitation of our work is the reliance on a single scalar reward for optimization, despite the problem having dual objectives: confusion and maintaining similarity. We recommend investigating alternative techniques, such as Multi-objective Reinforcement Learning, to address this limitation. This approach has the potential to enhance performance by optimizing both objectives concurrently. Moreover, the datasets used in paraphrasing currently need longer sequences, approximately 256 tokens, which restricts our approach to generating adversarial samples for longer sequences.

## Ethics Statement

Enhancing classifier performance is of utmost importance, especially considering the prevalence of hate and offensive speech on social media platforms. Many users attempt to circumvent the classifier's detection capabilities by altering their writing style or incorporating unfamiliar words, thereby creating edge cases where the classifier needs to identify such content accurately. This paper presents an innovative approach to generating these edge cases and leveraging adversarial training to enhance the classifier's ability to detect and protect against such samples.


## Acknowledgements

The authors would like to thank the members of the Wicked Applied Security Practices (WASP) Lab from the universities of Windsor/Victoria for their support in the annotation process.

This research is supported by the Vector Scholarship in Artificial Intelligence, provided through the Vector Institute and Natural Sciences and Engineering Research Council of Canada (NSERC) by NSERC Discovery Grant. This research was enabled in part by support provided by Compute Ontario and the Digital Research Alliance of Canada.

## References


Enes Altinisik, Hassan Sajjad, Husrev Taha Sencar, Safa Messaoud, and Sanjay Chawla. 2022. Impact of adversarial training on robustness and generalizability of language models. *arXiv preprint arXiv:2211.05523*.

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

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

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, et al. 2023. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. *arXiv preprint arXiv:1906.02443*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Prithiviraj Damodaran. 2021. Parrot: Paraphrase generation for nlu.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Chuyun Deng, Mingxuan Liu, Yue Qin, Jia Zhang, Hai-Xin Duan, and Donghong Sun. 2022. ValCAT: Variable-Length Contextualized Adversarial Transformations Using Encoder-Decoder Language Model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1735–1746, Seattle, United States. Association for Computational Linguistics.

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

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, and James Caverlee. 2020. Parade: A new dataset for paraphrase identification requiring computer science domain knowledge. *arXiv preprint arXiv:2010.03725*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*.

Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. 2017. First quora dataset release: Question pairs. *data. quora. com*.

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, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Aly Kassem, Omar Mahmoud, and Sherif Saad. 2023. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4379, Singapore. Association for Computational Linguistics.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. *arXiv preprint arXiv:2010.05700*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. *arXiv preprint arXiv:1708.00391*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Hieu Le, Dieu-thu Le, Verena Weber, Chris Church, Kay Rottmann, Melanie Bradford, and Peter Chin. 2022. Semi-supervised adversarial text generation based on Seq2Seq models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 254–262, Abu Dhabi, UAE. Association for Computational Linguistics.

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020. Data Boost: Text Data Augmentation Through Reinforcement Learning Guided Conditional Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9031–9041. ArXiv:2012.02952 [cs].

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

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Animesh Nighojkar and John Licato. 2021. Improving paraphrase detection with the adversarial paraphrasing task. *arXiv preprint arXiv:2106.07691*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the Style of Text! Adversarial and Backdoor Attacks Based on Text Style Transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. Adversarial attack and defense technologies in natural language processing: A survey. *Neurocomputing*, 492:278–307.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Petter Törnberg. 2023. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv preprint arXiv:2304.06588*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, and Nathan Lambert. 2020. Trl: Transformer reinforcement learning. `https://github.com/lvwerra/trl`.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020a. CAT-Gen: Improving Robustness in NLP Models via Controlled Adversarial Text Generation. ArXiv:2010.02338 [cs].

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2557–2563.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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

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

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*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jingjing Xu, Liang Zhao, Hanqi Yan, Qi Zeng, Yun Liang, and Xu Sun. 2019. LexicalAT: Lexical-Based Adversarial Reinforcement Training for Robust Sentiment Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5518–5527, Hong Kong, China. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of nlp models. *arXiv preprint arXiv:2109.00544*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V

Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating Natural Adversarial Examples. ArXiv:1710.11342 [cs].

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*.

## A Natural Language Policy Optimization vs PPO

NLPO (Natural Language Policy Optimization) is proposed to address the problem of large action spaces in language generation tasks. (Ramamurthy et al., 2022) showed that existing RL algorithms struggle with these spaces, as seen in models like GPT-2/3 and T5 with their extensive vocabularies of 50K and 32K, respectively, and even more prominent with the recent models. NLPO introduces a masking policy that is periodically updated and applies a top-p sampling technique to mask out irrelevant tokens during training. This helps balance the inclusion of task-relevant information with the risk of reward hacking. By extending the PPO algorithm, NLPO aims to improve the stability and effectiveness of training language models. NLPO achieves that by NLPO utilizing top-p sampling via generating, which limits the token selection to the smallest set where the cumulative probability exceeds a given threshold parameter p (Holtzman et al., 2018).

## B Implementation Details For Baseline Methods

We utilized the codebase provided by the authors for the baseline methods. Nevertheless, certain aspects were not explicitly addressed in their paper or the baseline implementation. Despite this, we made efforts to adapt these aspects in order to ensure minimal disruption to the overall framework.

### B.1 SCPN

SCPN is an approach that leverages an LSTM model trained on a large back-translation corpus to generate paraphrases. These paraphrases are then parsed using the Stanford parser. In order to generate adversarial samples, SCPN employs ten different parsing templates. We adopted the same methodology as the authors by utilizing their pre-trained models and following their steps to generate parse trees for our datasets using the Stanford parser (Manning et al., 2014). However, the paper and codebase did not provide details on how to select the appropriate parsing template. We devised a strategy for choosing the most suitable parsing template to address this. Given that the model generates ten templates, we initially use the pre-trained model to generate multiple paraphrases of input "x" using each template. Subsequently, we individually query the victim model with each generated paraphrase. We then measure the confusion and mutual implication score for each paraphrase and select the sample that yields the highest scores as the chosen paraphrase. This process ensures we prioritize the paraphrase that maximizes confusion and mutual implication with the victim model.

### B.2 StyleAdv

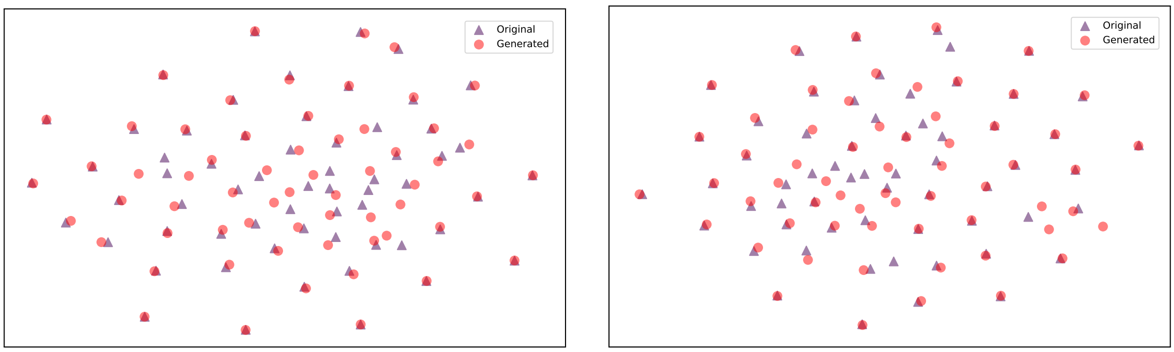
StyleAdv is an approach that leverages the power of STRAP (Style Transfer via Paraphrasing), a style transfer framework. This approach incorporates five distinct style transfer models, namely Bible, Poetry, Shakespeare, Lyrics, and Tweets, each capable of generating a unique style. To ensure consistency and reproducibility, we meticulously followed the procedure outlined in the paper and codebase for the adversarial generation of our datasets. However, we encountered a missing reference to the similarity model in the paper and codebase. Upon contacting the authors, they informed us that any similarity model would suffice. Consequently, we opted for the "all-MPNet-Base-v2" model, renowned for its exceptional performance on the semantic textual similarity (STS) benchmark (Muennighoff et al., 2022). We employed this model to measure cosine similarity, a reliable metric for comparing sentence similarity. The adversarial generation process unfolds: utilizing each style transfer model, we generate ten paraphrases for a single sentence, resulting in 50 paraphrases. Subsequently, we subject these generated paraphrases to classification by our classifier, measuring both the confusion and cosine similarity. If multiple examples cause the classifier to produce incorrect outputs, we select the adversarial example with the highest cosine similarity to the original input as the final choice.

## C Visualization Results

We utilized the T-SNE technique to generate a two-dimensional representation of the vectorized samples. To obtain these vectors, we employed the "all-MPNet-Base-v2" model and selected 60 random samples from each class for each dataset. We tried various numbers of samples, and 60 gives a clear observation of the phenomena.

Subsequently, T-SNE was applied to obtain the two-dimensional representation. Upon examining the figures depicting the proposed datasets, the same observation for TPRL-generated samples in SST-2 holds for the other datasets, which is the samples generated by TPRL displayed a striking resemblance to the original data, with instances of overlapping or partial overlap with the original sentences. This intriguing behavior sheds light on a crucial finding: even a minor shift in the semantic space relative to the original sentence can yield a sentence that successfully deceives the classifier.

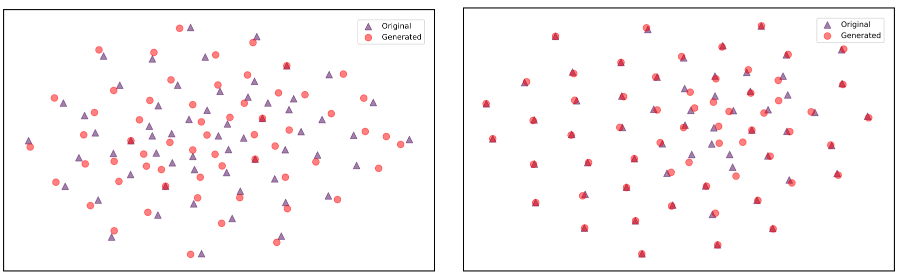
### C.1 SST-2



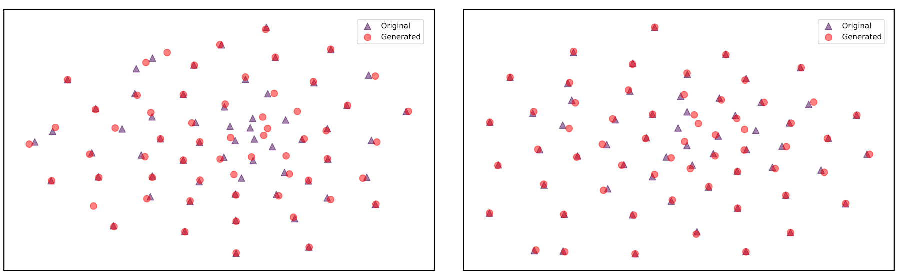

(a) Class-0 (b) Class-1

Figure 3: T-SNE visualization of the vectorized original and TPRL-adversarial sentences in the SST-2. The adversarial sentences (circles) mostly overlap with the original sentences (triangles), suggesting that generated sentences maintain the original class distribution.

### C.2 SST-5



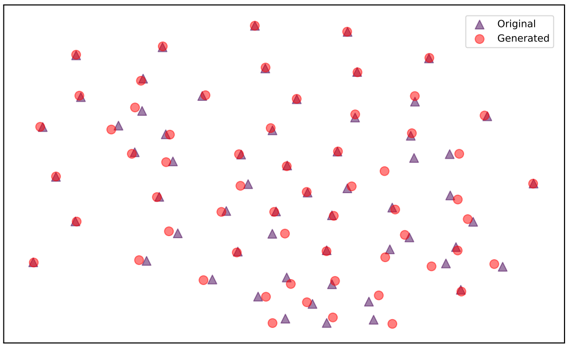

(a) Class-0-Train set (b) Class-1-Train set



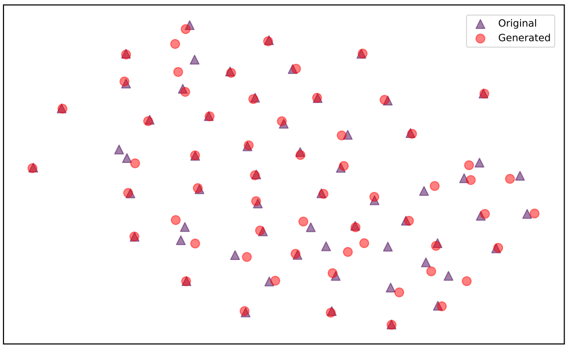

(c) Class-0-Test set (d) Class-1-Test set

Figure 4: T-SNE visualization of the vectorized original and TPRL-adversarial sentences in the SST-5. The adversarial sentences (circles) mostly overlap with the original sentences (triangles), suggesting that generated sentences maintain the original class distribution.

### C.3 Offensive Dataset



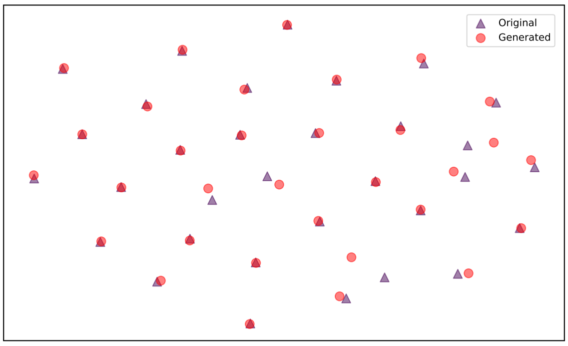

(a) Class-0




(b) Class-1

Figure 5: T-SNE visualization of the vectorized original and TPRL-adversarial sentences in OFF. The adversarial sentences (circles) mostly overlap with the original sentences (triangles), suggesting that generated sentences maintain the original class distribution.

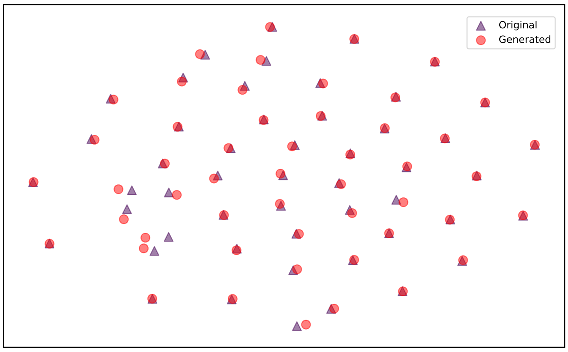
### C.4 Hate Dataset



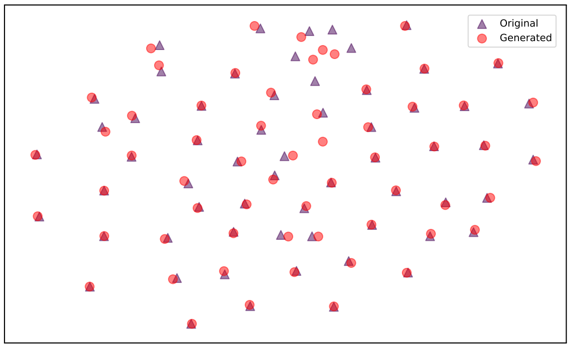

(a) Class-0

Figure 6: T-SNE visualization of the vectorized original and TPRL-adversarial sentences in HATE. The adversarial sentences (circles) mostly overlap with the original sentences (triangles), suggesting that generated sentences maintain the original class distribution.

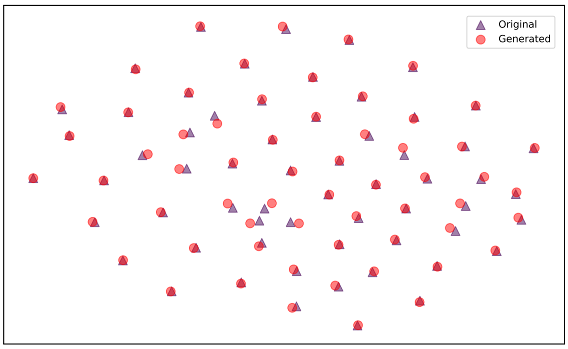
## C.5 Agnews Dataset



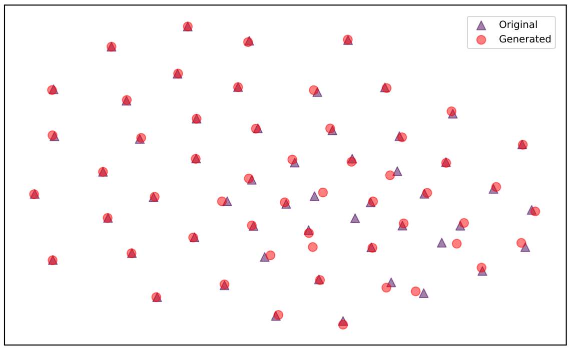

(a) Class-0




(b) Class-1




(c) Class-2




(d) Class-3

Figure 7: T-SNE visualization of the vectorized original and TPRL-adversarial sentences in AG's News. The adversarial sentences (circles) mostly overlap with the original sentences (triangles), suggesting that generated sentences maintain the original class distribution.

## D Universal Policy

As previously discussed, TPRL's learned policy demonstrates remarkable universality across multiple datasets and classifiers. In this section, we extensively analyze the learned policy's performance on each dataset, specifically focusing on its efficacy with four different classifiers. The following results highlight the consistent and impressive performance of the learned policy across diverse datasets and classifiers.

### D.1 SST-5

| **Policy/Classifier** | **BERT**$_{Base}$ (%) | **BERT**$_{Large}$ (%) | **RoBERTa**$_{Base}$ (%) | **RoBERTa**$_{Large}$ (%) | **DeBERTa-v3**$_{Large}$ (%) |
|---|---|---|---|---|---|
| None | 53.52 | 53.52 | 54.79 | 58.30 | 58.46 |
| Policy-BERT$_{Base}$ | 53.52 | 54.88 | **56.42** | $\star$ | 57.10 |
| Policy-BERT$_{Large}$ | 52.89 | 54.93 | 55.61 | $\star$ | 59.00 |
| Policy-RoBERTa$_{Base}$ | 51.62 | 54.84 | 56.15 | $\star$ | **59.54** |
| Policy-RoBERTa$_{Large}$ | **54.07** | **55.15** | $\star$ | **58.95** | 58.91 |
| Policy-DeBERTa-v3$_{Large}$ | 53.52 | 54.61 | 55.56 | $\star$ | 58.77 |

Table 6: Accuracy Results of Different Classifiers Trained With The Examples Generated By Various Attacking Policies On The SST-5 Dataset. Showing the Universal Policy. The best comparable performances policy for the classifier is bolded

### D.2 Offensive Dataset

| **Policy/Classifier** | **BERT**$_{Base}$ (%) | **BERT**$_{Large}$ (%) | **RoBERTa**$_{Base}$ (%) | **RoBERTa**$_{Large}$ (%) | **DeBERTa-v3**$_{Large}$ (%) |
|---|---|---|---|---|---|
| None | 84.76 | 85.11 | 83.95 | 85.93 | 84.65 |
| Policy-BERT$_{Base}$ | 85.11 | 85.00 | 83.95 | 72.09 | 84.76 |
| Policy-BERT$_{Large}$ | 85.11 | 85.58 | 84.76 | 84.76 | 84.53 |
| Policy-RoBERTa$_{Base}$ | 85.11 | 84.88 | 85.00 | **85.93** | 84.53 |
| Policy-RoBERTa$_{Large}$ | 84.30 | **85.93** | **85.23** | 84.53 | **86.04** |
| Policy-DeBERTa-v3$_{Large}$ | **85.81** | 85.81 | **85.23** | 83.02 | 85.93 |

Table 7: Accuracy Results of Different Classifiers Trained With The Examples Generated By Various Attacking Policies On The OFF Dataset. Showing the Universal Policy. The best comparable performances policy for the classifier is bolded

### D.3 Hate Dataset

| **Policy/Classifier** | **RoBERTa**$_{Base}$ (%) | **RoBERTa**$_{Large}$ (%) | **DeBERTa-v3**$_{Large}$ (%) |
|---|---|---|---|
| None | 91.90 | **92.45** | 89.50 |
| Policy-RoBERTa$_{Base}$ | 92.45 | 91.75 | **93.80** |
| Policy-RoBERTa$_{Large}$ | **92.75** | 92.05 | 92.60 |
| Policy-DeBERTa-v3$_{Large}$ | 90.65 | 91.45 | 92.30 |

Table 8: Accuracy Results of Different Classifiers Trained With The Examples Generated By Various Attacking Policies On The HATE Dataset. Showing the Universal Policy. The best comparable performances policy for the classifier is bolded

### D.4 AG's News Dataset

| Policy/Classifier | **BERT**$_{Base}$ (%) | **BERT**$_{Large}$ (%) | **RoBERTa**$_{Base}$ (%) | **RoBERTa**$_{Large}$ (%) | **DeBERTa-v3**$_{Large}$ (%) |
|---|---|---|---|---|---|
| None | 93.97 | 94.18 | 93.78 | 93.92 | 93.94 |
| Policy-BERT$_{Base}$ | **94.46** | **94.60** | 93.92 | 93.97 | **94.23** |
| Policy-BERT$_{Large}$ | 94.21 | 94.36 | 93.73 | 94.02 | 93.85 |
| Policy-RoBERTa$_{Base}$ | 94.17 | 94.28 | 93.93 | 93.78 | 91.01 |
| Policy-RoBERTa$_{Large}$ | 93.85 | 94.17 | 93.72 | **94.21** | 94.13 |
| Policy-DeBERTa-v3$_{Large}$ | 94.13 | 94.26 | **93.94** | 93.67 | 94.22 |

Table 9: Accuracy Results of Different Classifiers Trained With The Examples Generated By Various Attacking Policies On The AG's News Dataset. Showing the Universal Policy. The best comparable performances policy for the classifier is bolded

## E Automatic Evaluation

We adopted a comprehensive multi-perspective methodology to assess the quality of the generated adversarial samples, ensuring the following factors were taken into consideration: fluency, as determined by Perplexity (PPL) scores obtained from the GPT-2-XL language model (Radford et al., 2019). However, recognizing the inherent limitations of perplexity in accurately evaluating fluency, we supplemented this metric with the accuracy of a RoBERTa-large classifier, which was trained on the CoLA corpus (Warstadt et al., 2019). This classifier offers valuable insights into the grammatical acceptability of the generated samples. For measuring similarity, we utilized the "all-MPNet-Base-v2" embedding-based SIM model (Song et al., 2020; Reimers and Gurevych, 2019) to measure the semantic similarity between the input sentence and the generated samples. This model has demonstrated exceptional performance on the semantic textual similarity (STS) benchmark (Muennighoff et al., 2022), making it an ideal choice for our task. To further enhance our evaluation, we also integrated the mutual implication (MI) metric, which effectively captures the inferential role semantics. By incorporating the MI metric, we overcome the limitations of STS in fully capturing the inferential semantics, thereby providing a more comprehensive evaluation of the generated samples. The results for each dataset with each classifier are shown in the following table.

| **Dataset** | Classifier | BERT$_{\text{Base}}$ | | | | BERT$_{\text{Large}}$ | | | | RoBERTa$_{\text{Base}}$ | | | | RoBERTa$_{\text{Large}}$ | | | | DeBERTa$_{\text{Large}}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Attacker | PPL↓ | FL↑ | SIM↑ | MI↑ | PPL↓ | FL↑ | SIM↓ | MI↑ | PPL↓ | FL↑ | SIM↑ | MI↑ | PPL↓ | FL↑ | SIM↑ | MI↑ | PPL↓ | FL↑ | SIM↑ | MI↑ |
| SST-2 | SCPN | 467.84 | 58.47 | 72.98 | 73.20 | 461.367 | 58.68 | 73.08 | 73.49 | 442.519 | 57.29 | 71.78 | 70.67 | 442.95 | 58.23 | 71.95 | 70.16 | 434.443 | 57.88 | 71.99 | 70.80 |
| | StyleAdv | 1114.599 | 58.29 | **76.29** | 57.27 | 1173.938 | 57.05 | **74.57** | 53.44 | 1183.201 | 56.75 | **74.24** | 52.96 | 540.362 | 56.38 | **73.93** | 51.35 | 1281.874 | 56.49 | **73.33** | 50.31 |
| | TPRL | **293.436** | **88.97** | 67.18 | **86.11** | **327.538** | **86.36** | 72.21 | **87.18** | **396.76** | **87.59** | 72.94 | **85.02** | **405.21** | **86.58** | 58.14 | **90.11** | **438.16** | **87.0** | 70.81 | **85.8** |
| SST-5 | SCPN | 462.351 | 59.22 | 77.70 | 83.66 | 444.079 | 59.10 | 78.23 | 84.19 | 443.576 | 59.10 | 78.12 | 84.04 | 430.548 | 58.81 | 78.21 | 84.06 | 461.905 | 59.00 | 77.09 | 82.13 |
| | StyleAdv | **257.311** | 68.24 | **90.43** | 78.98 | **256.632** | 67.30 | **89.79** | 77.45 | **248.617** | 70.50 | **90.95** | 79.97 | **315.372** | 67.70 | **89.49** | 77.26 | **261.436** | 69.42 | **89.90** | 78.27 |
| | TPRL | 354.96 | **85.94** | 75.2 | **87.53** | 283.46 | **86.84** | 75.17 | **85.31** | 401.87 | **85.48** | 78.96 | **87.49** | 319.86 | **86.86** | 74.89 | **85.02** | 373.00 | **86.72** | 75.90 | **86.22** |
| HS | SCPN | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | 736.244 | 58.85 | 75.83 | 85.91 | 739.991 | 58.59 | 75.70 | 85.69 | 746.287 | 58.42 | 75.85 | 85.42 |
| | StyleAdv | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | **367.99** | 57.34 | **79.14** | 60.97 | 360.303 | 58.84 | **76.42** | 56.77 | 407.045 | 59.15 | **79.22** | 60.02 |
| | TPRL | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | $\star$ | 973.77 | **80.61** | 59.14 | **92.15** | **212.95** | **89.69** | 74.8 | **89.76** | **197.68** | **90.05** | 75.43 | **90.52** |
| OFF | SCPN | 780.052 | 47.44 | 73.43 | 79.51 | 801.278 | 47.46 | 73.39 | 80.21 | 827.18 | 47.41 | 73.74 | 80.32 | 790.174 | 47.58 | 73.61 | 80.40 | 817.611 | 47.20 | 73.93 | 80.81 |
| | StyleAdv | 1273.684 | 51.28 | 77.12 | 58.73 | 2069.405 | 52.09 | 77.45 | 58.48 | 1316.525 | 51.91 | 76.64 | 57.20 | 1298.085 | 52.84 | 77.46 | 58.27 | 1827.177 | 51.87 | 79.25 | 59.90 |
| | TPRL | **316.61** | **87.28** | 72.83 | **87.50** | **472.94** | **87.17** | 72.47 | **88.58** | **412.41** | **85.88** | 70.87 | **87.85** | **327.49** | **85.69** | 67.17 | **90.60** | **355.65** | **86.73** | 71.13 | **87.32** |
| AG's News | SCPN | 383.066 | 29.05 | 72.62 | 80.76 | **382.20** | 29.05 | 72.60 | 80.85 | 367.79 | 28.91 | 72.68 | 80.50 | 371.46 | 28.95 | 72.69 | 80.64 | **343.28** | 28.94 | 72.56 | 80.39 |
| | StyleAdv | **356.54** | 51.55 | 53.34 | 39.45 | 435.91 | 50.70 | 54.92 | 42.15 | 425.10 | 49.81 | 57.34 | 43.86 | 384.90 | 50.93 | 54.25 | 40.42 | 451.27 | 51.55 | 52.88 | 38.78 |
| | TPRL | 508.63 | **86.19** | **79.05** | **98.46** | 406.12 | **87.02** | **75.68** | **98.23** | **277.79** | **88.39** | **74.21** | **97.44** | **248.27** | **88.11** | **79.20** | **97.57** | 600.80 | **85.88** | **76.24** | **98.51** |

## F Hyperparameters Details

The model was trained for thirty epochs as we tried a range of epochs and picked up the best value that achieves a higher reward in the training environment. While a batch size of 32 was chosen empirically, as changing batch size did not affect performance. The training process employed the Lion optimizer, as proposed by (Chen et al., 2023), in their work on symbolic optimization. With a learning rate of $4.9 \times 10^{-6}$ suggested by (von Werra et al., 2020), the Lion optimizer demonstrated superior convergence compared to the commonly used Adam optimizer (Kingma and Ba, 2014).

## G GPT-3.5 Annotation Details

### G.1 Measuring Similarity Via GPT-3.5

To assess the similarity of larger generated samples, we used GPT-3.5. Using "from 1 to 5, how much is the generated sentence similar to the original (1 being very dissimilar and 5 being very similar)?" as a prompt. We randomly selected 100 samples from each framework in the SST-2 dataset. Ratings were given on a scale of 1 to 5, with 1 being very dissimilar and 5 being very similar. Results for TPRL: 5 (9%), 4 (43%), 3 (33%), and 2 (15%); SCPN: 5 (9%), 4 (36%), 3 (29%), 2.5 (1%), 2 (23%), and 1 (1%); StyleAdv: 4 (22%), 3 (33%), 2 (34%), and 1 (11%). TPRL achieved the highest similarity ratings in categories 5, 4, and 3, indicating similarity to the original samples. SCPN ranked second, while StyleAdv received the lowest ratings. These GPT-3.5 findings align with human evaluation results.

### G.2 Measuring Diversity Via GPT-3.5

To validate our observation regarding the diversity of the generated samples, we used GPT-3.5 to assess the diversity of generated samples for the three baselines. Using "from 1 to 5, how much is the generated sentence diverse from the original (1 being very non-diverse and 5 being very diverse)?" as a prompt. We randomly selected 250 sentences from SST-2. TPRL had 100 samples, SCPN had 77 samples, and StyleAdv had 73 samples. The scaling rates were as follows: TPRL: 5 (6%), 4 (34%), 3 (45%), 2 (12%), and 1 (3%). SCPN: 5 (11%), 4 (20%), 3 (19%), 2 (44%), and 1 (3%). StyleAdv: 4 (9%), 3 (15%), 2 (54%), and 1 (20%). These results confirm that TPRL generates more diverse samples, while cosine similarity fails to account for this diversity and considers it as a high similarity.

## H Classifiers Error Analysis

To demonstrate dissimilarities in errors across employed classifiers, we utilized the following methodology: We inspected the intersection of misclassified samples for each dataset to examine whether or not a sample was present in all classifiers' misclassification sets, which we have termed the AND operation. Additionally, to inspect whether a sample was present in any of the classifiers' misclassification sets, we searched for unique samples, which we have designated as the OR operation. Our analysis indicates that the AND operation ranges from 9% to 16%, with an average of 10.57%. Conversely, the OR operation ranges between 32% and 55%, averaging 41.49%. Following fine-tuning with transferred samples from differing classifiers, we evaluated whether the improved performance was solely achieved through shared samples, which yielded a shared sample average of 30%. Our analysis confirms that the policy shares both universal and model-specific features.

## I Human Annotation Details

We further conduct a human evaluation study of our attacks to examine to what extent are adversarial texts generated by TPRL truly imperceptible. We asked the annotators to follow the following instructions:

In this task, you will have two sentences, and you are required to say whether it has the same semantic meaning or not.

Semantic meaning in this context means that the two sentences have the same meaning, may be fully or partially. Let's look at the following example:

Ex.1.
S1: Ezekiel Ansah is wearing 3D glasses wout the lens,
S2: Wait Ezekiel ansah is wearing 3d movie glasses with the lenses knocked out.
Same Meaning: YES

Ex.2.
S1: Marriage equality law passed in Rhode Island,

S2: Congrats to Rhode Island becoming the 10th state to enact marriage equality.
Same Meaning: YES

EX.3.
S1. Finally saw the Ciara body party video
S2. ciara s Body Party video is on point
Same Meaning: NO

Ex.4.
S1. Now lazy to watch Manchester united vs arsenal
S2. Early lead for Arsenal against Manchester United
Same Meaning: NO

In the first sentence, the two sentences are fully equivalent. In the second sentence, the two sentences are partially equivalent. In the third and fourth sentences, the two sentences are fully not equivalent. So, if the sentences are fully/partially equivalent, we can consider them as the same meaning.

## J Hardware & Software Dependencies

For the paraphraser fine-tuning process, we utilized a cluster equipped with 4x V100 GPUs, each with 32GB of memory. To enhance the efficiency, we employed a zero-2 stage DeepSpeed framework (Rajbhandari et al., 2020). These models were fine-tuned using the HuggingFace library (Wolf et al., 2019) and PyTorch (Paszke et al., 2017). For RL fine-tuning, we utilized 2x V100 GPUs, each with 32GB of memory, and employed TRL (Transformer Reinforcement Learning) library(von Werra et al., 2020).