Annealing Genetic-based Preposition Substitution for Text Rubbish Example Generation

Chen Li, **Xinghao Yang**, **Baodi Liu**, **Weifeng Liu*** and **Honglong Chen**
China University of Petroleum (East China)
lc621yeah@163.com, yangxh@upc.edu.cn, thu.liubaodi@gmail.com, liuwf@upc.edu.cn, chenhl@upc.edu.cn

Abstract

Modern Natural Language Processing (NLP) models expose under-sensitivity towards text rubbish examples. The text rubbish example is the heavily modified input text which is nonsensical to humans but does not change the model's prediction. Prior work crafts rubbish examples by iteratively deleting words and determining the deletion order with beam search. However, the produced rubbish examples usually cause a reduction in model confidence and sometimes deliver human-readable text. To address these problems, we propose an Annealing Genetic based Preposition Substitution (AGPS) algorithm for text rubbish sample generation with two major merits. Firstly, the AGPS crafts rubbish text examples by substituting input words with meaningless prepositions instead of directly removing them, which brings less degradation to the model's confidence. Secondly, we design an Annealing Genetic algorithm to optimize the word replacement priority, which allows the Genetic Algorithm (GA) to jump out the local optima with probabilities. This is significant in achieving better objectives, i.e., a high word modification rate and a high model confidence. Experimental results on five popular datasets manifest the superiority of AGPS compared with the baseline and expose the fact: the NLP models can not really understand the semantics of sentences, as they give the same prediction with even higher confidence for the nonsensical preposition sequences.

1 Introduction

Recent researches have shown that Deep Neural Networks (DNNs) are vulnerable to external perturbation in Natural Language Processing (NLP) field for text classification. Adversarial text examples have attracted much attention from researchers [Papernot *et al.*, 2016; Jia and Liang, 2017; Li *et al.*, 2020; Kwon and Lee, 2022; Wang *et al.*, 2022]. The Adversarial text example is the slightly disturbed original input. These modifications are usually imperceptible to humans but can trigger DNNs' false prediction, which exposes DNNs' over-sensitivity to small changes. On the contrary, the rubbish example refers to the heavily modified example, where it is totally unrecognizable to humans but has nearly no effect on the model's prediction. Figure 1 lists several rubbish examples in our experiments, where the input words are replaced with meaningless prepositions, but the model still gives high confidence in the true label.

SNLI	
Premise	A man and a woman cross the street in front of a pizza and gyro restaurant.
Before	The people are standing still on the curb.
After	The within are without among on the until.
Label	Contradiction
Confidence	95%→100%
MR	
Before	It's a tour de force, written and directed so quietly that it's implosion rather than explosion you fear.
After	It's a among de among, among and among so among that it's among among than among you among.
Label	Positive
Confidence	92%→94%

Figure 1: The rubbish examples crafted by our AGPS.

Rubbish examples reflect the under-sensitivity of modern DNNs to a large number of text modifications. [Welbl *et al.*, 2020] emphasized that lack of sensitivity is a challenging issue for neural models, which usually leads to unreliable predictions in real text recognition tasks, such as spam filtering [Guzella and Caminhas, 2009], toxic comment detection [Risch and Krestel, 2020], resume recommendation [Roy *et al.*, 2020], and medical diagnose [Bakator and Radosav, 2018]. Rubbish samples allow the model to leverage spurious clues in the data enough to achieve a high level of performance without understanding task-related textual meaning. Models can achieve strong nominal accuracy on the training set containing rubbish samples by utilizing prediction shortcuts that can not represent a given NLP task, but this leads to severe failure of prediction on samples without these spurious clues. Therefore, exploring potential text rubbish examples is crucial to expose DNNs' vulnerabilities to avoid security

*Corresponding author

risks caused by such under-sensitivity.

The concept of rubbish example is primarily proposed by [Goodfellow *et al.*, 2014; Nguyen *et al.*, 2015] in the computer vision community by fooling images, which are nonsensical to humans as degenerate inputs but can be labeled with high confidence as a specific class by DNNs. In the NLP domain, [Feng *et al.*, 2018] proposed a text rubbish example generation approach named *input reduction*, which iteratively removes insignificant words from the input sentence while maintaining the model's prediction unchanged. After the word deletions, some of the crafted rubbish examples contain only one or two words, which lack reasonable information for humans to make any predictions but the model can confidently retain the original prediction. However, there are three challenges to the input reduction method. Firstly, it employs the beam search to determine the word deletion priority, while this is usually not guaranteed to produce a globally optimal solution, i.e., deleting the most words. Secondly, during the word removal process, the semantic information of the words is not considered, the remaining words often contain important information for humans to make a correct decision. For example, in the sentiment analysis task, the original input classified as "positive" (i.e., A fascinating and fun film) is transformed into a rubbish example (i.e., fascinating), where the word "fascinating" strongly indicates a positive sentiment. Thirdly, a high-quality rubbish example should not affect the prediction of the DNNs model, but the reduced text usually causes a reduction in average model confidence over most datasets.

In this paper, we propose AGPS, an Annealing Genetic based Preposition Substitution algorithm for text rubbish sample generation to address these problems. Specifically, we first construct a set of substitution candidates by carefully collecting nonsensical prepositions. Each input word can be replaced with any candidate in the set of candidate substitutions, and our goal is to achieve more substitutions without reducing model confidence. In the second step, we present a hybrid Annealing Genetic algorithm to optimize the word replacement priority. Population-based search strategy can realize distributed search in the whole solution space, and skip the sub-optima combined with the Metropolis mechanism of the simulated annealing algorithm. This can make it a greater probability to approach the best rubbish sample, i.e., replacing the maximum number of input words. Extensive experiments demonstrate that the proposed AGPS generates more effective rubbish examples with higher model confidence. Our main contributions are summarized as follows:

- We propose a novel Annealing Genetic based Preposition Substitution to generate text rubbish examples. The AGPS replaces the words with nonsensical prepositions to alleviate the decrease in the model's confidence.
- We design an effective annealing genetic algorithm to optimize the word replacement priority, which steps out of the local optima with a probabilistic strategy. This is significant to find the global optimal of the objective function, i.e., a high word modification rate and high model confidence.
- We evaluate the effectiveness of our AGPS on five popular datasets by attacking seven representative DNNs models. Experimental results manifest that the AGPS outperforms the baseline and also shows good properties in retraining and transfer attacks.

2 Related Works

In this section, we first introduce the adversarial sample and the rubbish sample, then we briefly review the classical genetic algorithm and the simulated annealing algorithm.

2.1 Textual Adversarial Attack

There is a growing body of research on NLP adversarial samples, where researchers have studied various invariant text transformations for different tasks. [Papernot *et al.*, 2016] first focused on the adversarial samples in Natural Language Processing (NLP) field for text classification and proposed to generate adversarial input sequences on Recurrent Neural Networks (RNN). [Jia and Liang, 2017] fooled reading comprehension systems by linking a distracting sentence to the input paragraph. [Ebrahimi *et al.*, 2017] presented a token-flipping method, which crafted adversarial samples based on the gradients of the one-hot input vectors. [Zang *et al.*, 2019] developed a sememe-based word substitution method and particle swarm optimization-based search algorithm, which treats word-level attacks as combinatorial optimization problems. [Wang *et al.*, 2022] proposed SemAttack, a unified and effective semantic adversarial attack framework that leverages diverse semantic perturbation functions to generate natural adversarial text.

2.2 Rubbish Examples

Unlike the large volume of research on adversarial examples, the effect of rubbish examples is greatly underestimated. In the text domain, [Feng *et al.*, 2018] introduced rubbish samples to explore the limitations of the NLP interpretation approaches. They achieved this by iteratively removing the least important words from the input while keeping the model predictions constant. The importance of each word is evaluated utilizing gradient-based approximation, and the word deletion optimization is completed using beam search. As a result, valid inputs are gradually transformed into rubbish samples, which lack prominent information to support humans to make a convincing decision but keep the original prediction of DNNs with high confidence. However, employing input reduction to craft rubbish examples has three challenges. Firstly, without the local optimal jump-out strategy, beam search might neglect potential global optimal solutions. Secondly, the rubbish example may contain significant words for effective human predictions since input reduction does not consider the semantics of the words. Thirdly, experiments indicate that in most cases, the reduction of input causes a decrease in model confidence, severely affecting the quality of generated rubbish samples.

2.3 Genetic Algorithm

Inspired by biological evolution in nature, the Genetic Algorithm (GA) is a heuristic algorithm that iteratively searches for the optimal solution by imitating natural selection. Specifically, the quality of population members is evaluated by the

fitness function in each generation. Following the principle of survival of the fittest, parents are usually selected based on the fitness of individuals in the current generation. The next generation is generated through crossover and mutation. It has been proven effective in solving the optimization problems such as text clustering [Song and Park, 2009], text classification [Bidi and Elberrichi, 2016], and textual adversarial attacking [Alzantot *et al.*, 2018]. However, there is a contradiction between the convergence speed and the global optimal solution of the basic genetic algorithm, and the basic genetic algorithm is prone to premature convergence.

2.4 Simulated Annealing Algorithm

The Simulated Annealing algorithm (SA) is inspired by the physical annealing procedure of metals. Generally, the SA starts from a high temperature with the initial molecules and aims to optimize the position of these molecules with the temperate drops. At each temperature, the simulation must proceed long enough for the system to reach equilibrium. The Metropolis principle allows the algorithm to jump out of the local optimal by chance. It accepts all solutions that are superior to the current solution and accepts inferior solutions by a probability related to the annealing temperature. The SA has been used in solving the optimization problems such as text summarization [Mosa *et al.*, 2019] and textual adversarial attacking [Yang *et al.*, 2021]. However, SA is highly dependent on the parameters (e.g., initial temperature, chilling temperature, and sampling times at each temperature), and inappropriate cooling methods or too fast cooling speed will also limit the solution quality of the algorithm.

3 Methodology

In this section, we discuss the details of the proposed AGPS algorithm, including the problem definition of black-box rubbish example generation (§ 3.1), the preposition-based word substitution (§ 3.2), and the annealing genetic optimization (§ 3.3). Figure 2 shows the workflow of our AGPS.

3.1 Problem Definition

Similar to the adversarial attack, the black-box rubbish example attacker can not access the model architecture, parameters, or gradients. We can only query the target model with supplied inputs to obtain the prediction results and confidence scores, which are closer to a real-world scenario.

Given an input space containing K samples $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K\}$ and an output space $\mathcal{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_L\}$ containing L labels, the classifier F needs to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ from any input sample $\mathbf{X} \in \mathcal{X}$ to the correct label $\mathbf{Y}_{true}$, by maximizing the posterior probability:

$$\underset{\mathbf{Y}_i \in \mathcal{Y}}{\operatorname{argmax}} P(\mathbf{Y}_i|\mathbf{X}) = \mathbf{Y}_{true} \tag{1}$$

The rubbish sample generation method aims to make heavy modifications to the input $\mathbf{X}$ to produce rubbish example $\mathbf{X}^*$, which is unreadable to humans but does not change the classifier's prediction:

$$\underset{\mathbf{Y}_i \in \mathcal{Y}}{\operatorname{argmax}} P(\mathbf{Y}_i|\mathbf{X}^*) = \mathbf{Y}_{true} \tag{2}$$

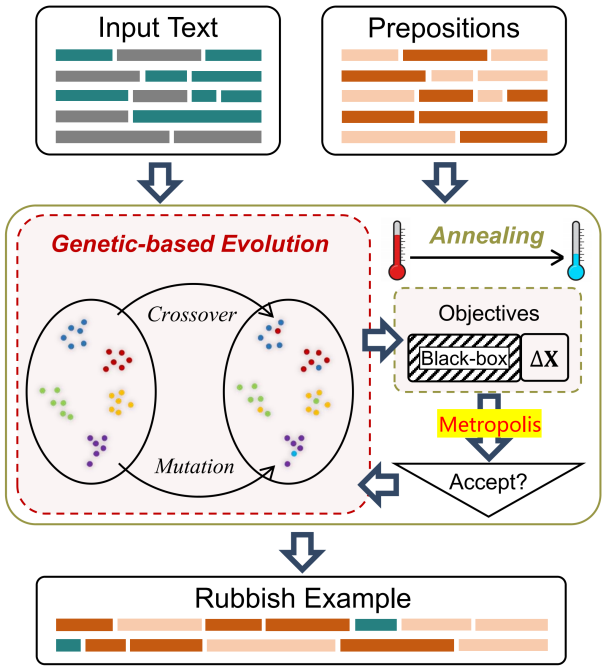

Figure 2: The framework of the proposed AGPS algorithm.

Formally, the objectives are maximizing the difference $\triangle\mathbf{X}$ between input $\mathbf{X}$ and rubbish output $\mathbf{X}^*$ as shown in Eq. (3)

$$\triangle\mathbf{X} = len(\mathbf{X}^* \neq \mathbf{X}) \tag{3}$$

and simultaneously ensuring the classifier gives high confidence in the original label

$$P_{true} = \begin{cases} F(\mathbf{X}^*) & , \ \mathbf{Y} = \mathbf{Y}_{true} \\ 0 & , \ \mathbf{Y} \neq \mathbf{Y}_{true} \end{cases} \tag{4}$$

Eq. (4) indicates that if the predicted label is not equal to the original label, we regard the modification as a failure and set the confidence to 0. The final objective function balances Eq. (3) and Eq. (4) with a parameter δ

$$\mathcal{J}(F, \mathbf{X}, \mathbf{X}^*, \delta) = v \times (\triangle\mathbf{X} + \delta \times P_{true}) \tag{5}$$

where v is a constant to adjust the function value to an appropriate range. Therefore, the black-box rubbish example generation problem is equal to maximizing the objective function $\mathcal{J}$. To optimize the objective function, we need to address two problems, i.e., (1) how to select substitution candidates, and (2) how to determine the word attack priority.

3.2 Preposition-based Word Substitution

In this work, we propose to make preposition-based word substitution to produce human unreadable modifications, i.e., $\triangle\mathbf{X}$, because the prepositions usually contain no real semantic information. To this end, we collect 37 commonly used prepositions as a set of meaningless substitutions $\mathbb{S}$. The attack starts with a Part-of-Speech (POS) tagging for each word. To achieve efficiency, we only replace four kinds of input words, i.e., only nouns, adjectives, verbs, and adverbs, and filter out all the other words. For each modifiable word w_i, we search for the substitution word $w_i^{'}$ from $\mathbb{S}$.

Every $w_i^{'} \in \mathbb{S}$ is a potential candidate for replacing the original word w_i. To select the best candidate from $\mathbb{S}$, we define the candidate score as the original label probability:

$$S_{w_i^{'}} = P(\mathbf{Y}_{true}|\mathbf{X}_i^{'}), \forall w_i^{'} \in \mathbb{S} \tag{6}$$

where

$$\mathbf{X}_i^{'} = \{w_1, w_2, \cdots, w_i^{'}, \cdots, w_n\} \tag{7}$$

Then we identify the candidate $w_i^{'}$ that maximizes the prediction probability of the original label as the best substitution word, i.e.,

$$w_i^* = \underset{w_i^{'} \in \mathbb{S}}{\operatorname{argmax}} S_{w_i^{'}} \tag{8}$$

This solves the first problem.

3.3 Annealing Genetic Algorithm

After candidate selection, each input word can be replaced as a determined preposition. Then we need to optimize the word modification priority. In this work, we propose an annealing genetic algorithm to determine the attacking sequence, which allows the traditional GA to jump out of local optimal with the Metropolis principle. Specifically, after the crossover, the proposed annealing genetic accepts some individuals with a worse objective value than their parents by a probability. This allows the evolution proceeds to the global optimal solution. Besides, it can also avoid the excessive dependence of the SA on parameters and improve the solution stability.

To detail our algorithm, we first introduce the subroutine *Crossover* and *Mutation*.

- *Crossover*: The input of this subroutine is two parent individuals, i.e. P_1 and P_2, which are two sentences. It scans each modifiable word in P_1 in turn and then exchanges words with the corresponding position of P_2, where the input words are independent of each other and have the same crossover probability. This process also follows the substitution principle in section 3.2, i.e., only exchanging nouns, adjectives, verbs, and adverbs.
- *Mutation*: We adopt a single point mutation strategy. The input of the subroutine is a single individual, which can be the original input sentence or its modification. It randomly selects a modifiable word w_i in the original sentence to replace. The substitution candidate $w_i^{'}$ belongs to $\mathbb{S}$. The best substitution word selection method follows Eq. (8).

The optimization procedure is given in Algorithm 1. Next, we describe our algorithm step by step.

Firstly, we **initialize** the first generation by repeating the mutation subroutine for N times to get N population members $\mathbb{X}^0 = \{\mathbb{X}_1^0, \mathbb{X}_2^0, \cdots, \mathbb{X}_N^0\}$ as shown in line 2-4 of Algorithm 1. Then, we obtain the fitness of each population member in the initial generation by querying the victim model function $\mathcal{J}$.

We adopt the proportional **selection** approach to select the parents to breed the next generation population. To make the well-performing population members have a higher probability of being selected as parents, the i^{th} individual in the current population is selected with a probability proportional p_i

Algorithm 1: The proposed AGPS algorithm

Input: Input sentence $\mathbf{X}_{ori} = (w_1, \cdots, w_n)$
Input: DNNs classifier F
Output: Rubbish example $\mathbf{X}^*$

1 Initialization: the population size $N = 40$, the number of iteration times $G = 15$, the temperature $T = 1000$, the attenuation factor $\alpha = 0.85$, the balance parameter $\delta = 2.5$, the initial rubbish example $\mathbf{X}^* = \mathbf{X}_{ori}$;
/* Initialize the first generation */
2 **for** $i = 1, \cdots, N$ **do**
3 $\quad \mathbb{X}_i^0 \leftarrow Mutation(\mathbf{X}_{ori})$;
4 $\quad y_i^0 = \mathcal{J}(\mathbb{X}_i^0)$;
5 $\mathbf{X}^* = \mathbb{X}^0_{argmax_j y_j^0}$; ▷ Optimal individual
/* The Annealing Genetic Starts */
6 **for** $g = 1, \cdots, G$ **do**
7 $\quad$ **for** $i = 1, \cdots, N$ **do**
8 $\quad\quad$ Select parents P_1 and P_2 from $\mathbb{X}^{g-1}$ with p_{select} in Eq. (9);
9 $\quad\quad$ $child \leftarrow Crossover(P_1, P_2)$;
/* Metropolis accept crossover */
10 $\quad\quad$ $\mathbb{X}_i^g = P_{argmax_j \mathcal{J}(P_j)}$;
11 $\quad\quad$ **if** $\mathcal{J}(child) \geq \mathcal{J}(\mathbb{X}_i^g)$ **then**
12 $\quad\quad\quad$ $\mathbb{X}_i^g = child$;
13 $\quad\quad$ **else**
14 $\quad\quad\quad$ $p = e^{-(\mathcal{J}(\mathbb{X}_i^g) - \mathcal{J}(child))/T}$;
15 $\quad\quad\quad$ $r = random(0, 1)$;
16 $\quad\quad\quad$ **if** $r < p$ **then**
17 $\quad\quad\quad\quad$ $\mathbb{X}_i^g = child$;
18 $\quad\quad$ $child_m \leftarrow Mutation(\mathbb{X}_i^g)$;
/* Metropolis accept mutation */
19 $\quad\quad$ **if** $\mathcal{J}(child_m) \geq \mathcal{J}(\mathbb{X}_i^g)$ **then**
20 $\quad\quad\quad$ $\mathbb{X}_i^g = child_m$;
21 $\quad\quad$ **else**
22 $\quad\quad\quad$ $p = e^{-(\mathcal{J}(\mathbb{X}_i^g) - \mathcal{J}(child_m))/T}$;
23 $\quad\quad\quad$ $r = random(0, 1)$;
24 $\quad\quad\quad$ **if** $r < p$ **then**
25 $\quad\quad\quad\quad$ $\mathbb{X}_i^g = child_m$;
26 $\quad$ $T = \alpha \times T$; ▷ Porportional annealing
/* Optimal individual preservation */
27 $\quad$ **for** $i = 1, \cdots, N$ **do**
28 $\quad\quad$ $y_i^g = \mathcal{J}(\mathbb{X}_i^g)$;
29 $\quad$ $\mathbb{X}_{best}^g = \mathbb{X}^g_{argmax_j y_j^g}$;
30 $\quad$ **if** $\mathcal{J}(\mathbb{X}_{best}^g) > \mathcal{J}(\boldsymbol{X}^*)$ **then**
31 $\quad\quad$ $\mathbf{X}^* = \mathbb{X}_{best}^g$;
32 **return** $\mathbf{X}^*$

to its fitness $\mathcal{J}(\mathbb{X}_i)$ in line 8.

$$\frac{\mathcal{J}(\mathbb{X}_i)}{\sum_{i=1}^{N} \mathcal{J}(\mathbb{X}_i)} \tag{9}$$

In the **population evolution** step, we perform crossover and mutation operations on the selected parent members. Specif-

ically, the new child sample is synthesized by calling the *Crossover* subroutine. To decide whether the child is accepted or not, we follow the typical Metropolis principle.

$$p = \begin{cases} 1 & , \ \mathcal{J}(child) \geq y \\ e^{-(y-\mathcal{J}(child))/T} & , \ \mathcal{J}(child) < y \end{cases} \quad (10)$$

If the fitness score of the child is higher than that of the parents, we directly accept it as the next generation population member. Otherwise, we accept an inferior sample by Eq. (10) as shown in lines 14-17. If not accepted, pick the parent with higher fitness to be the child for the next iteration in line 10. Finally, the *Mutation* subroutine is implemented to the resulting child and the Metropolis principle is employed to determine whether to accept this mutation. As these two subroutines are based on individuals rather than populations, we iteratively perform the *Crossover* and *Mutation* subroutine to generate N population members of the new generation.

In line 26, we select the proportional **cooling** strategy to reduce the temperature

$$T = \alpha \times T \quad (11)$$

where the attenuation factor $\alpha = 0.85$.

Then we **record** the best individuals after each iteration by computing the fitness of all individuals in each generation and preserving optimal individuals to promote the population evolution. Finally, the optimization will be terminated if the evolution reaches the upper bound or it generates a good rubbish example. This solves the second problem.

4 Experiments

We provide the source code in the GitHub[1] to ensure that all the results in this section are reproducible.

4.1 Datasets and Victim Models

We assess the attack performance on five public datasets, such as Stanford Sentiment Treebank (SST-2), Movie Reviews (MR), Stanford Natural Language Inference (SNLI), Quora Question Pairs (QQP), and Microsoft Research Paraphrase Corpus (MRPC).

- SST-2 [Socher *et al.*, 2013] consists of 67349 training examples and 1821 testing samples, and each example is a movie review with binary classes. The task is to predict if the text comment belongs to positive or negative emotions.
- MR [Pang and Lee, 2005] is also a sentiment classification dataset, containing 8530 training data and 1066 test data. Similar to SST-2, all the examples belong to positive or negative comments for movies.
- SNLI [Bowman *et al.*, 2015] is a popular question inference corpus with 550152 examples for training and 10000 examples for testing, where each example consists of a question pair. The two questions are duplicate or not duplicate.
- QQP [Shankar *et al.*, 2017] is another question inference database with the same labels as SNLI, i.e., duplicate and not duplicate. It covers 363846 and 390965 examples in the train set and test set, respectively.
- MRPC [Wang *et al.*, 2018] includes 3668 sentence pairs for model training and 1725 sentence pairs for testing, which can be divided into two categories, i.e., the two sentences are semantically equivalent or not equivalent.

We attack seven victim models to test the capability of our AGPS, such as CNN, LSTM, BERT (bert-base-uncased) [Devlin *et al.*, 2018], DistilBERT (distilbert-base-uncased, distilbert-base-cased) [Sanh *et al.*, 2019], RoBERTa (roberta-base) [Liu *et al.*, 2019], ALBERT (albert-base-v2) [Lan *et al.*, 2019], and XLNet (xlnet-base-cased) [Yang *et al.*, 2019]. We download the models from HuggingFace[2].

4.2 Baseline Method

To evaluate the effectiveness of our AGPS, we compare it with Input Reduction (IR) [Feng *et al.*, 2018]. To the best of our knowledge, this is the single rubbish example generation work in the text field. The IR crafts the rubbish samples by iteratively removing unimportant words from the input with beam search. The objective are (1) craft short rubbish text examples that lack enough information for a human to make a decision, and (2) keep the model's prediction unchanged.

4.3 Evaluation Metrics

We evaluate the quality of rubbish samples with the following three metrics.

Modification rate. The percentage of modified words. Since we only modify the input words with semantic meanings, including nouns, adjectives, verbs, and adverbs, we only take these words into account when reporting the word modification rate.

Model confidence. The true label probability of the classifier for the rubbish sample.

Semantic similarity. The semantic similarity between the original sample and the rubbish sample. Following [Jin *et al.*, 2019; Morris *et al.*, 2020a], we employ the universal sentence encoder (USE) [Cer *et al.*, 2018] with cosine similarity to estimate the semantic similarity.

4.4 Experimental Setup

The parameter settings for our AGPS are given in the initialization, i.e., line 1, of Algorithm 1. For the baseline, we use the author-recommended parameter settings. We randomly select 500 examples from each dataset to implement the text rubbish attack for a fair comparison. In the natural language inference task (i.e., SNLI), we only modify the hypothesis, while keeping the premise unchanged. All experiments are implemented on the NLP attack package TextAttack [Morris *et al.*, 2020b].

4.5 Experimental Results

The experimental results of model confidence, modification rate, and semantic similarity are shown in Table 1. We ask the following three questions to manifest the contributions as claimed in the Introduction section.

[1]https://github.com/soar-create/AGPS

[2]https://huggingface.co/models

Dataset	Model	ACC	Length	Con			Mod		Sim	
				Original	IR	AGPS	IR	AGPS	IR	AGPS
SST-2	CNN	84.60%	8.44	87.59%	70.26%	**92.79%**	74.86%	**95.66%**	0.63	**0.57**
	LSTM	96.20%	8.44	95.06%	95.60%	**98.87%**	79.34%	**95.47%**	0.63	**0.58**
	BERT	98.80%	8.44	99.29%	98.89%	**99.61%**	79.19%	**95.79%**	0.63	**0.59**
	RoBERTa	97.20%	8.44	98.65%	97.43%	**99.55%**	79.35%	**96.01%**	0.62	**0.59**
	DistilBERT	98.60%	8.44	99.35%	98.73%	**99.68%**	79.12%	**95.08%**	0.63	**0.59**
	AlBERT	97.80%	8.44	97.38%	96.23%	**97.65%**	79.16%	**96.06%**	0.63	**0.58**
MR	CNN	98.00%	18.62	88.54%	67.29%	**93.99%**	92.34%	**94.84%**	**0.56**	**0.56**
	LSTM	89.60%	18.62	86.36%	91.08%	**95.30%**	92.58%	**94.66%**	**0.56**	**0.56**
	BERT	99.80%	18.62	**99.68%**	98.28%	98.56%	92.36%	**93.70%**	**0.56**	0.57
	RoBERTa	95.20%	18.62	94.38%	90.89%	**98.37%**	91.28%	**94.64%**	**0.56**	0.57
	XLNet	98.20%	18.62	97.68%	96.79%	**99.24%**	92.33%	**94.76%**	**0.56**	0.57
SNLI	BERT	95.60%	22.19	96.17%	87.07%	**98.64%**	75.88%	**98.75%**	0.63	**0.58**
	DistilBERT	86.80%	22.19	91.40%	83.94%	**92.45%**	73.69%	**98.39%**	0.64	**0.58**
	AlBERT	92.60%	22.19	92.88%	78.51%	**95.73%**	78.35%	**98.49%**	0.62	**0.58**
QQP	BERT	96.20%	22.00	97.00%	97.95%	**99.53%**	89.20%	**89.29%**	**0.57**	0.66
	DistilBERT	95.20%	22.00	97.06%	95.14%	**98.26%**	88.95%	**90.05%**	**0.58**	0.65
	AlBERT	98.20%	22.00	98.18%	98.56%	**99.50%**	88.62%	**89.38%**	**0.58**	0.65
MRPC	DistilBERT	91.00%	39.03	88.54%	68.34%	**91.93%**	90.32%	**95.93%**	**0.61**	0.66
	XLNet	95.40%	39.03	93.53%	83.87%	**96.49%**	94.27%	**95.29%**	**0.60**	0.65
Aervage		—	—	94.67%	89.20%	**97.17%**	84.80%	**94.85%**	0.6000	**0.5968**

Table 1: The average word modification rate (Mod), the average model confidence (Con), and the average semantic similarity (Sim) of different algorithms on five text classification datasets. The best results are highlighted in bold. The "ACC" column represents the original accuracy of models, and the "Original" column represents the original average model confidence without attacks.

MR Example: Positive (99%)
Chicago is sophisticated, brash, sardonic, completely joyful in its execution.
IR: Positive (99%)
joyful.
AGPS: Positive (100%)
before is as, across, through, up for in its within.
QQP Example: Duplicate (99%)
Question 1: How can I lose 4kg weight? Question 2: What are the ways of losing weight?
IR: Duplicate (97%)
Question 1: weight? Question 2: weight?
AGPS: Duplicate (99%)
Question 1: out can I to under besides? Question 2: What are the up of down besides?

Table 2: The rubbish examples crafted by IR and AGPS.

Question 1: Is the preposition substitution better than word deletion in keeping the model's confidence? To answer this question, we list the model confidence on the text rubbish examples generated by IR and our AGPS as well as the original sentences in Table 1. From the Con part, we can see that our AGPS outperforms the baseline by a large margin, i.e. improves by 7.97% on average. Besides, the seven victim models even exhibit higher confidence (by 2.5%) in our AGPS rubbish examples than the original examples. This strongly validates the superiority of preposition-based word replacement in generating text rubbish examples.

Question 2: Is the Annealing Genetic algorithm superior to the beam search in skipping the local optimal? To reply to this question, we list the word modification rate in Table 1, as the common objective of beam search and the annealing genetic is to modify the maximal number of words. The results in Table 1 show that our AGPS improves the Mod by 10.05% in comparison with the baseline. A higher word modification rate indicates a higher quality of the rubbish example because either word deletion or preposition substitution can improve semantic confusion for human readers.

Question 3: Can our AGPS avoid generating semantic consistency text? We answer this question by comparing the semantic similarity in Table 1 'Sim' column. The results imply that our AGPS achieves comparable semantic similarity with IR, i.e., superior to the IR by only 0.0032. However, we observe that the IR frequently preserves important label information in the final rubbish examples. Table 2 exhibits two rubbish examples crafted by IR and AGPS. Intuitively, the IR examples containing 'joyful' clearly suggest a positive sentiment for humans, while AGPS properly solves this problem.

Sentence (Label: positive)	Confidence
Ultimately , it ponders the reasons we need stories so much .	0.95
Ultimately , it ponders the reasons we need stories so **within** .	0.95
Ultimately , **it** **at** the reasons we need stories so within .	0.97
Through , it at the reasons we need stories so within .	0.99
Through , it at the **into** we need stories so within .	0.99
Through , it at the into we need **for** so within .	1.00
Through , it at the into we **from** for so within .	1.00

Figure 3: A replacement path for an MR example with visualization of attributions for each word token in the sequence. Darker color indicates more importance. The substitutions are highlighted in bold.

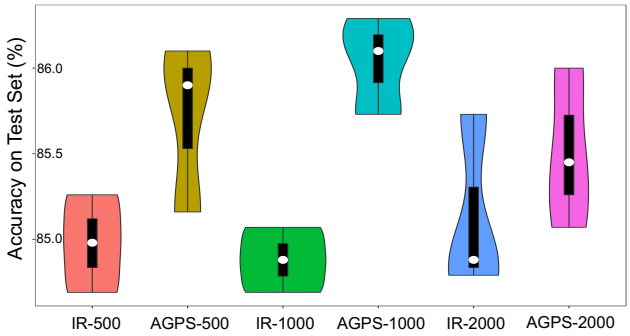

Figure 4: Adversarial retraining results on MR dataset.

Overall AGPS achieves better performance than the baseline on these three metrics across different datasets, which demonstrates the superiority of our approach.

4.6 Attribution Visualization and Analysis

Feature attribution is a popular strategy for model interpretation. Figure 3 visualizes the word replacement path and the dynamic changes in word importance. Our AGPS example satisfies the definition of rubbish example: (1) humans can not make any predictions for the nonsensical preposition sequence, but (2) the model's confidence is enhanced from 95% to 100%. Besides, we observe three interesting properties of DNNs models. Firstly, the model tends to pay more attention to those unmodified words (e.g. 'it' in the third step), although it is not important before. Secondly, sometimes replacing the most important words does not reduce the model's confidence but increases it, e.g., the 'Ultimately → Through' improves the confidence by 2%. These phenomenons bring challenges to the interpretation methods, which interpret the model properties relying on the word importance [Feng *et al.*, 2018; Ghorbani *et al.*, 2019]. We hope these findings promote the development of model interpretation theory.

4.7 Adversarial Training

Adversarial training is a prevalent defense strategy on adversarial robustness by incorporating adversarial examples into the training set. In this part, we randomly generate and join {500, 1000, 2000} MR rubbish samples as the negative sample set to the training data and retrain the BERT model. Then

Metric	Before	Retrain_{500}	Retrain_{1000}	Retrain_{2000}
Mod	96.03%	75.11%	76.72%	77.46%
Con	98.78%	97.84%	93.29%	93.39%
Sim	0.57	0.65	0.65	0.64

Table 3: Defense performance test after model retraining.

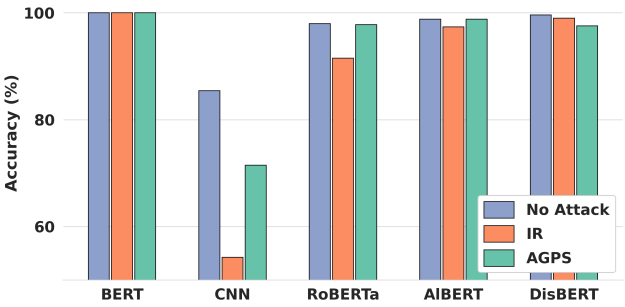

Figure 5: Transferability test results. BERT is the victim model.

we re-attack the retrained model to test the defense performance. As shown in Table 3, adversarial training improves the difficulty of attack, because attacking the retrained model with fewer modification rates, lower model confidence, and the rubbish example carries more semantics. Besides, Figure 4 shows the multi-retrain classification accuracy on the test set. Figure 4 illustrates that AGPS brings greater robustness improvement than the baseline.

4.8 Transferability

For rubbish examples, transferability refers to whether the rubbish sample designed for a model F_1 can also hold the same prediction on another unknown model F_2. We evaluate the transferability on SST-2 dataset. Specially, we collect the rubbish examples crafted for BERT and then test the transferability on four unknown models (CNN, RoBERTa , AlBERT, and DistilBERT). The experimental results in Figure 5 illustrate that both IR and AGPS exhibit high transferability on most models (i.e., RoBERTa, ALBERT, DistilBERT), and our AGPS outperforms IR in most cases.

5 Conclusion

In this paper, we proposed an innovative AGPS algorithm for generating text rubbish examples. The AGPS employs the preposition substitution strategy instead of word deletions to reduce the loss of model confidence. We also designed an annealing genetic algorithm to determine the word modification priority, which allows the optimization to jump out of local optima. The research exposes the under-sensitivity of neural models: the input can be modified to the nonsensical word sequences, but the model even exhibits higher original label confidence. In the future, our works can be employed to test the interpretation methods, and we hope these results encourage further work in improving the robustness and interpretability of natural language models.

Acknowledgments

This work was supported in part by the Yunnan Provincial Major Science and Technology Special Plan Projects (Grant 202202AD080003), in part by the Outstanding Youth Science Foundation Project of Shandong Province (Overseas) (Grant No.2023HWYQ-070), and in part by the Independent Innovation Research Project (Grant No.22CX06059A).

References

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

[Bakator and Radosav, 2018] Mihalj Bakator and Dragica Radosav. Deep learning and medical diagnosis: A review of literature. *Multimodal Technologies and Interaction*, 2(3):47, 2018.

[Bidi and Elberrichi, 2016] Noria Bidi and Zakaria Elberrichi. Feature selection for text classification using genetic algorithms. In *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, pages 806–810. IEEE, 2016.

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

[Cer *et al.*, 2018] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

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

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

[Feng *et al.*, 2018] Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[Ghorbani *et al.*, 2019] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.

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

[Guzella and Caminhas, 2009] Thiago S Guzella and Walmir M Caminhas. A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, 36(7):10206–10222, 2009.

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

[Jin *et al.*, 2019] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2, 2019.

[Kwon and Lee, 2022] Hyun Kwon and Sanghyun Lee. Ensemble transfer attack targeting text classification systems. *Computers & Security*, 117:102695, 2022.

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

[Li *et al.*, 2020] Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. Contextualized perturbation for textual adversarial attack. *arXiv preprint arXiv:2009.07502*, 2020.

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

[Morris *et al.*, 2020a] John X Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. Reevaluating adversarial examples in natural language. *arXiv preprint arXiv:2004.14174*, 2020.

[Morris *et al.*, 2020b] John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*, 2020.

[Mosa *et al.*, 2019] Mohamed Atef Mosa, Arshad Syed Anwar, and Alaa Hamouda. A survey of multiple types of text summarization with their satellite contents based on swarm intelligence optimization algorithms. *Knowledge-Based Systems*, 163:518–532, 2019.

[Nguyen *et al.*, 2015] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[Pang and Lee, 2005] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005.

[Papernot *et al.*, 2016] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting adversarial input sequences for recurrent neural networks.

In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 49–54. IEEE, 2016.

[Risch and Krestel, 2020] Julian Risch and Ralf Krestel. Toxic comment detection in online discussions. In *Deep learning-based approaches for sentiment analysis*, pages 85–109. Springer, 2020.

[Roy *et al.*, 2020] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, and Rocky Bhatia. A machine learning approach for automation of resume recommendation system. *Procedia Computer Science*, 167:2318–2327, 2020.

[Sanh *et al.*, 2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[Shankar *et al.*, 2017] Iyer Shankar, Dandekar Nikhil, and Csernai Kornel. First quora dataset release: question pairs (2017). *URL https://www. quora. com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs*, 2017.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 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, 2013.

[Song and Park, 2009] Wei Song and Soon Cheol Park. Genetic algorithm for text clustering based on latent semantic indexing. *Computers & Mathematics with Applications*, 57(11-12):1901–1907, 2009.

[Wang *et al.*, 2018] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[Wang *et al.*, 2022] Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. Semattack: natural textual attacks via different semantic spaces. *arXiv preprint arXiv:2205.01287*, 2022.

[Welbl *et al.*, 2020] Johannes Welbl, Po-Sen Huang, Robert Stanforth, Sven Gowal, Krishnamurthy Dj Dvijotham, Martin Szummer, and Pushmeet Kohli. Towards verified robustness under text deletion interventions. *ICLR 2020*, 2020.

[Yang *et al.*, 2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[Yang *et al.*, 2021] Xinghao Yang, Weifeng Liu, Dacheng Tao, and Wei Liu. Besa: Bert-based simulated annealing for adversarial text attacks. In *IJCAI*, pages 3293–3299, 2021.

[Zang *et al.*, 2019] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. *arXiv preprint arXiv:1910.12196*, 2019.