TextHacker: Learning based Hybrid Local Search Algorithm for Text Hard-label Adversarial Attack

Zhen Yu[1], Xiaosen Wang[1,2], Wanxiang Che[3], Kun He[1]
[1] School of Computer Science and Technology, Huazhong University of Science and Technology
[2] Huawei Singular Security Lab, [3] Research Center for SCIR, Harbin Institute of Technology

EMNLP 2022

Introduction

DNNs in NLP tasks are known to be vulnerable to adversarial examples, in which imperceptible modification on the correctly classified samples could mislead the model.

Hard-label Attack: a kind of Black-Box Attack. Attacker can only access the model hard prediction label, which is more applicable in real-world scenarios but also more challenging.

Background: due to the limited information (i.e., only the prediction labels) for hard-label attacks, it is hard to estimate the word importance, leading to relatively low effectiveness and efficiency on existing hard-label attacks.

Algorithm

Algorithm 1: The TextHacker Algorithm

Input: Input sample x, target classifier f, query budget T, reward r, population size S, maximum number of local search N
Output: Attack result and adversarial example

```
▷ Adversary Initialization
Construct the candidate set C(w_i) for each w_i ∈ x
x_1 = x, x_1^adv = None
for t = 1 → T do
    x_{t+1} = WordSubstitution(x_t, C)
    if f(x_{t+1}) ≠ f(x) then
        x_1^adv = x_{t+1}; break
if x_1^adv is None then
    return False, None                         ▷ Initialization fails
▷ Perturbation Optimization
Initialize the weight table W with all 0s
x_{i+1}^adv = LocalSearch(x_i^adv, C, W)
P^1 = {x_1^adv, ..., x_i^adv, ..., x_S^adv}
t = t + S − 1; g = 1
while t ≤ T do
    P^g = P^g ∪ {Recombination(P^g, W)}
    for each text x_g^adv ∈ P^g do
        With x_1^adv = x_g^adv for i = 1 → N:
            x_{i+1}^adv = LocalSearch(x_i^adv, C, W);
            WeightUpdate(x_i^adv, x_{i+1}^adv, f, W)
        P^g = P^g ∪ {x_{N+1}^adv}
        t = t + N
    Construct P^{g+1} with the top S fitness in P^g
    Record global optima x^best with the highest fitness
    g = g + 1
return True, x^best                            ▷ Attack succeeds
```

Methodology

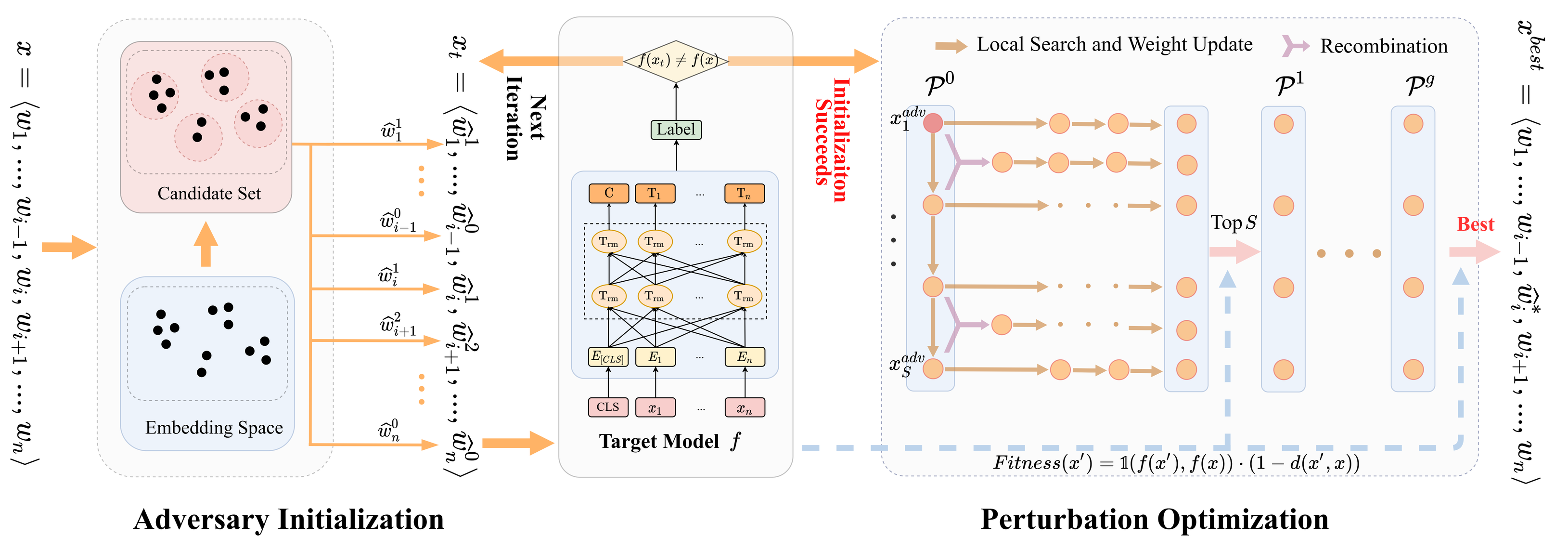

Figure 1: The overall framework of the proposed TextHacker.

We adopt the hybrid local search algorithm with weight table, a population based algorithm that contains local search, weight update and recombination operators, to minimize the adversary perturbation.

- **Local search** greedily substitutes unimportant word with the original word or critical word using the weight table to search for better adversarial example from the neighborhood.
- **Weight update** highlights the important words and positions by assigning different reward for each operated word, which helps the local search select more critical positions and synonyms to substitute.
- **Recombination** crafts non-improved solutions by randomly mixing two adversarial examples, which globally changes the text to avoid poor local optima.

Experiments

Model	Attack	AG's News Succ.	AG's News Pert.	IMDB Succ.	IMDB Pert.	MR Succ.	MR Pert.	Yelp Succ.	Yelp Pert.	Yahoo! Answers Succ.	Yahoo! Answers Pert.
BERT	GA	40.5	13.4	50.9	5.0	65.6	10.9	36.6	8.6	64.2	7.6
	PSO	45.8	12.1	60.3	3.7	74.4	10.7	47.9	7.5	64.7	6.6
	HLBB	54.7	13.4	77.0	4.8	65.8	11.4	57.1	8.2	82.0	7.7
	TextHoaxer	52.0	12.8	78.8	5.1	67.1	**11.1**	58.3	8.5	83.1	7.6
	TextHacker	**63.2**	**11.9**	**81.5**	**3.4**	**73.1**	11.4	**63.2**	**6.7**	**87.2**	**6.3**
Word CNN	GA	70.0	12.1	59.6	5.9	72.9	11.1	44.4	9.0	62.0	8.7
	PSO	83.5	10.4	55.6	4.2	80.7	10.7	45.6	7.4	52.7	7.0
	HLBB	74.0	11.7	74.0	4.2	71.1	11.2	67.1	7.6	78.7	7.8
	TextHoaxer	73.5	11.5	76.5	4.6	71.1	**10.7**	68.1	8.0	78.6	7.8
	TextHacker	**81.7**	**10.2**	**77.8**	**3.0**	**78.3**	11.1	**75.4**	**6.4**	**84.5**	**6.3**
Word LSTM	GA	45.5	12.4	50.8	5.7	67.2	11.2	40.7	8.1	51.2	8.6
	PSO	54.2	11.6	42.5	4.5	73.0	10.9	44.5	6.7	43.3	7.3
	HLBB	56.8	12.7	72.1	4.1	68.3	11.2	61.0	6.6	70.8	8.3
	TextHoaxer	56.5	12.3	73.5	4.5	67.9	**10.7**	61.8	6.7	70.1	8.1
	TextHacker	**64.7**	**11.2**	**76.2**	**3.0**	**75.2**	11.2	**65.4**	**5.5**	**75.5**	**6.9**

Table 1: Attack success rate (Succ., %) ↑, perturbation rate (Pert., %) ↓ of various attacks on three models using five datasets for text classification under the query budget of 2,000. ↑ denotes the higher the better. ↓ denotes the lower the better. We **bold** the highest attack success rate and lowest perturbation rate among the hard-label attacks.

Experiments

Attack	SNLI Succ.	SNLI Pert.	MNLI Succ.	MNLI Pert.	MNLIm Succ.	MNLIm Pert.
GA	67.2	14.6	67.6	12.6	66.9	12.2
PSO	70.7	15.0	72.0	12.9	70.8	12.4
HLBB	57.2	**14.0**	58.3	**12.2**	58.6	**11.8**
TextHoaxer	61.0	14.1	64.0	12.4	63.8	12.0
TextHacker	**70.3**	15.0	**68.3**	12.8	**69.0**	12.4

Table 2: Evaluation for textual entailment under the query budget of 500.

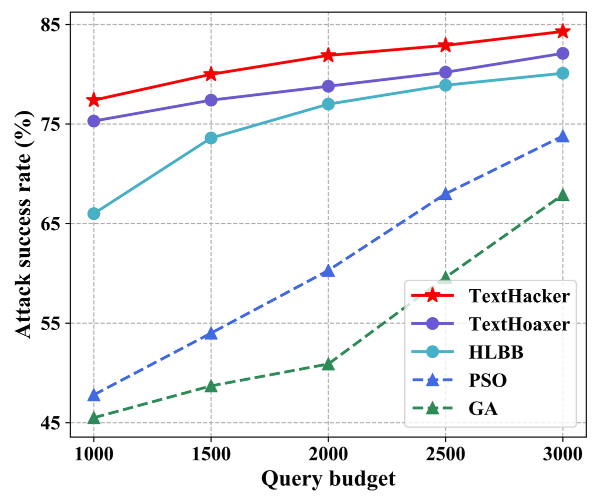

Figure 2: Evaluation on BERT using IMDB under various query budgets.

Attack	Succ.	Pert.	Sim.	Gram.
GA	50.9	5.0	79.3	0.9
PSO	60.3	3.7	81.8	0.7
HLBB	77.0	4.8	84.9	0.6
TextHoaxer	78.8	5.1	**85.8**	0.6
TextHacker	**81.5**	**3.4**	82.3	**0.4**

Table 3: Evaluation on adversary quality on BERT using IMDB.

Attack	Succ.	Pert.	Sim.	Gram.	Time
HLBB	65.0	5.7	82.1	0.5	8.7
TextHoaxer	65.0	5.2	**82.2**	0.4	9.3
TextHacker	**75.0**	**3.1**	80.9	**0.3**	**5.7**

Table 4: Evaluation on Amazon Cloud APIs under the query budget of 2,000.

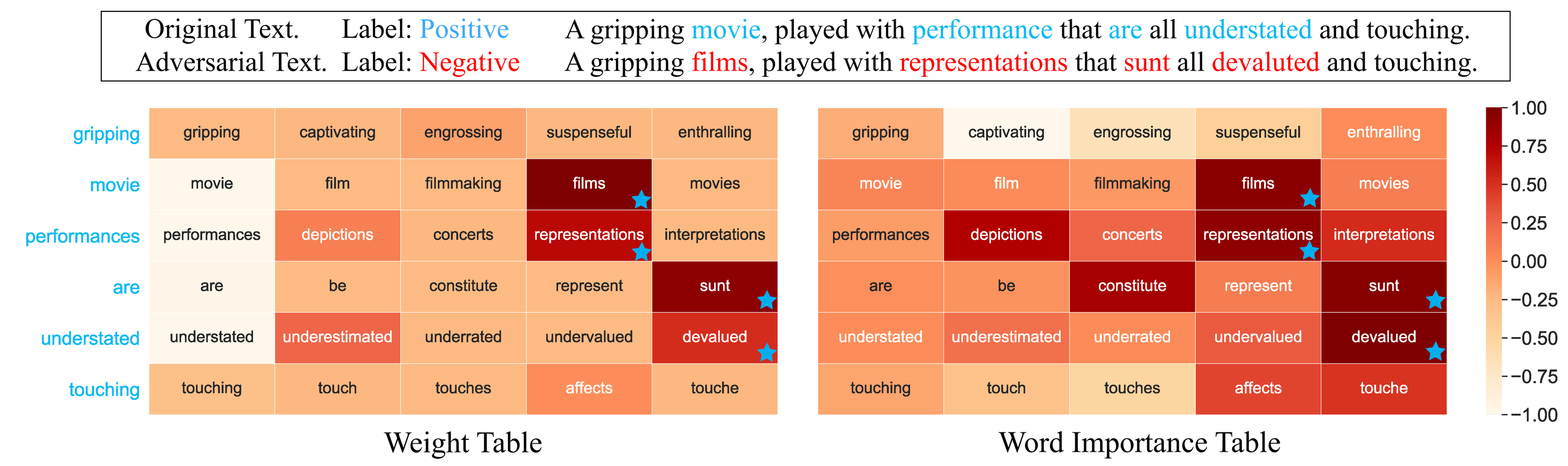

Figure 3: Visualization of the weight table in TextHacker and the word importance table from the victim model, representing the word importance of nouns, verbs, adjectives, adverbs, and their candidate words in the original text. The original words are highlighted in Cyan, with each row representing the candidate words. The substituted words are highlighted in **Red** with marker ⋆. A darker color indicates a more important word.

Conclusion

- We propose a novel text hard-label attack, called TextHacker, which captures the words that have higher impact on the adversarial example via the changes on prediction label to guide the search process at the perturbation optimization stage.
- Extensive evaluations for two typical NLP tasks, namely text classification and textual entailment, using various datasets and models demonstrate that TextHacker achieves higher attack success rate and lower perturbation rate than existing hard-label attacks and generates higher-quality adversarial examples.