Misleading Relation Classifiers by Substituting Words in Texts

Tian Jiang, Yunqi Liu, Yan Feng, Yuqing Li, Xiaohui Cui*
Key Laboratory of Aerospace Information Security and Trusted Computing,
Ministry of Education, School of Cyber Science and Engineering,
Wuhan University
{jiangtianjason, yunqi1028, fengyan1214, liyuqing20, xcui}@whu.edu.cn

Abstract

Relation classification is to determine the semantic relationship between two entities in a given sentence. However, many relation classifiers are vulnerable to adversarial attacks, which is using adversarial examples to lead victim models to output wrong results. In this paper, we propose a simple but effective method for misleading relation classifiers. We first analyze the most important parts of speech (POSs) from the syntax and morphology perspectives, then we substitute words labeled with these POS tags in original samples with synonyms or hyponyms. Experimental results show that our method can generate adversarial texts of high quality, and most of the relationships between entities can be correctly identified in the process of human evaluation. Furthermore, the adversarial examples generated by our method possess promising transferability, and they are also helpful for improving the robustness of victim models.

1 Introduction

Relation Extraction (RE) is to extract relationships contained in the text. It is an important part of Information Extraction (IE), and it is widely used in knowledge graph construction, information retrieval, and Q&A systems. Relation classification is an important link in the process of RE, and it aims to predict a relation between two entities in a sentence (Lyu and Chen, 2021). In most cases, the information of entities in samples has already been given (e.g., by manual marking), thus, the effect of relation classification will directly influence the overall performance of RE.

Deep Learning (DL) models have advantages in Natural Language Processing (NLP) tasks, but many Pre-trained language models are still facing potential threats from adversarial attacks (Wang et al., 2022). To check and further improve the robustness of DL models, there already exist diverse methods to generate adversarial examples currently. Moreover, the mechanisms of finding vulnerable places (tokens to be manipulated potentially) in texts and choosing perturbation types (e.g., Substitution, Insertion, Deletion, etc.) are gradually refined. However, some methods (Ren et al., 2019; Li et al., 2019) need to rank the importance of each word in finding vulnerable places, and some methods (Ebrahimi et al., 2018; Alzantot et al., 2018) ignore the characteristics of different parts of speech (POSs), so they treat each token in texts equally. Most of them need additional prerequisites (such as pre-trained models for word transformation) more or less in the process of implementation, and the essence (some POSs are more important than others in nature) of perturbing those words is seldom talked about in these works.

In this paper, we present a rule-based method[1] for generating adversarial texts against relation classification. We first analyze the most important POSs from the perspective of linguistics. To make small changes in original samples and avoid grammar mistakes, our method finds synonyms or hyponyms for tokens labeled with the most important POS tags, and it chooses words that have a great impact on the classification results of victim models. We conduct experiments on 2 datasets and generate adversarial examples against 3 classification models. Compared with other related studies, our method can generate samples of higher quality. These adversarial texts preserve the original relationships between entities well (relationships contained in 85.3% of the texts can be correctly judged by all 12 volunteers at the same time), and they also perform well in the experiment that studied transferability. Besides, an average of 71.59% adversarial examples can be correctly classified after retraining victim models using updated training sets (containing 5000 adversarial examples).

*Corresponding author

[1]Code, data, and models are available at https://github.com/JiangTianJason/Substitution_based_Attack

Findings of the Association for Computational Linguistics: ACL 2023, pages 3770–3785
July 9-14, 2023

2 Related Work

According to the target of perturbation imposed on input samples, the Textual Adversarial Attacks can be divided into 4 levels (Zeng et al., 2021), including **Character-Level**, **Word-Level**, **Character/Word-Level**, and **Sentence-Level**.

Character-Level. Methods (Gao et al., 2018; He et al., 2021) in Character-Level add perturbations to change the characters of important words. However, texts generated by these methods are usually with more grammar mistakes and are easy to be recognized as adversarial examples by human beings.

Word-Level. Probability Weighted Word Saliency (Ren et al., 2019) (PWWS) uses the *word saliency* (Li et al., 2016) strategy to determine the replacement order, and follows the order to substitute words with synonyms that have the greatest impact on output until the output change. Similar to PWWS, Genetic (Alzantot et al., 2018) (Gen) is also a score-based method for constructing adversarial examples. It uses the *Perturb Subroutine* to select the best replacement word, and employs the genetic algorithm to find successful adversarial examples with fewer modifications. To better retain sentence semantics, the masked-language-based attacks (Garg and Ramakrishnan, 2020; Li et al., 2020) replace or insert words based on their context.

Character/Word-Level. HotFlip (Ebrahimi et al., 2018) (HF) uses the gradient of models to estimate the change in loss, and it uses beam search to find a set of operations for creating white-box adversarial examples. Similar to HF, TextBugger (Li et al., 2019) (TB) is also a gradient-based method for creating adversarial texts in both white-box and black-box settings, and it provides more (5) options for manipulating characters/words in sentences.

Sentence-Level. Like methods in (Iyyer et al., 2018; Ribeiro et al., 2018), most Sentence-Level methods paraphrase texts using pre-trained models, and the syntactic changes between original and adversarial texts are usually significant.

For a better understanding of model robustness, DiagnoseAdv (Li et al., 2021b) leverages the salience-based analysis of adversarial examples, and it finds spurious correlation (between perturbed tokens and predicted labels) and Out-Of-Distribution (OOD, which means perturbed tokens do not appear in the training set) are the two main reasons for the incorrect output of victim models.

3 Methodology

3.1 Preliminary

According to the theory of syntax (Chomsky, 2014), a sentence can be divided into several constituents based on the different relationships between words, and there are 8 constituents (*Subject, Predicate, Object, Predicative, Object Complements, Attributive, Adverbial, Appositive*) which compose multiple sentence types (Simple Sentence, Compound Sentence, and Complex Sentence, etc.). Figure 1 shows the syntactic functions of some words or structures (phrases, clauses, etc.) perform in sentences.

From Figure 1, we can see *Nouns* appear more frequently than other types of words or structures. Since *Predicate* is a core constituent, we can treat *Verbs* as important as *Nouns*. Besides, *Adjectives* also appear in multiple types of constituents. Thus, the changes in *Nouns*, *Verbs*, and *Adjectives* have a higher possibility of leading relation classifiers to output wrong results.

Luckily, the definition of *lexical morphemes* in morphology (Yule, 2020) can support the correctness of the above analysis results. Lexical morphemes (including *Nouns, Verbs, Adjectives*, and *Adverbs*) can be used to express specific things, qualities, states, or actions, and provide the main meaning of a phrase or sentence. Besides, the above analysis results also explain why *Nouns, Verbs, Adjectives* are Top-3 POS tags in three perturbation types (Li et al., 2021a).

3.2 Proposed Method

To generate adversarial examples and mislead victim classifiers, we propose a rule-based method to substitute tokens in original samples. The 2 main steps of our method are as follows: (1) tagging all tokens with POSs and filtering tokens, (2) finding and adjusting substitution words for each token. Finally, we pick out words that have the greatest impact on the output, and substitute tokens with these words. Algorithm 1 describes how to modify tokens in original samples using our method.

In Algorithm 1, x represents the original sample, which contains all tokens (T), the first (h) and second entity (t), and the labeled relation (rel) between two entities. L is a set which contains types of tokens (*Noun*, *Verb*, etc.) to be modified. The victim model V can output the inference result (relationship rel) and confidence ($conf$) at the same time. Forbidden set F contains words that need to be skipped in processing, such as *linking verbs*

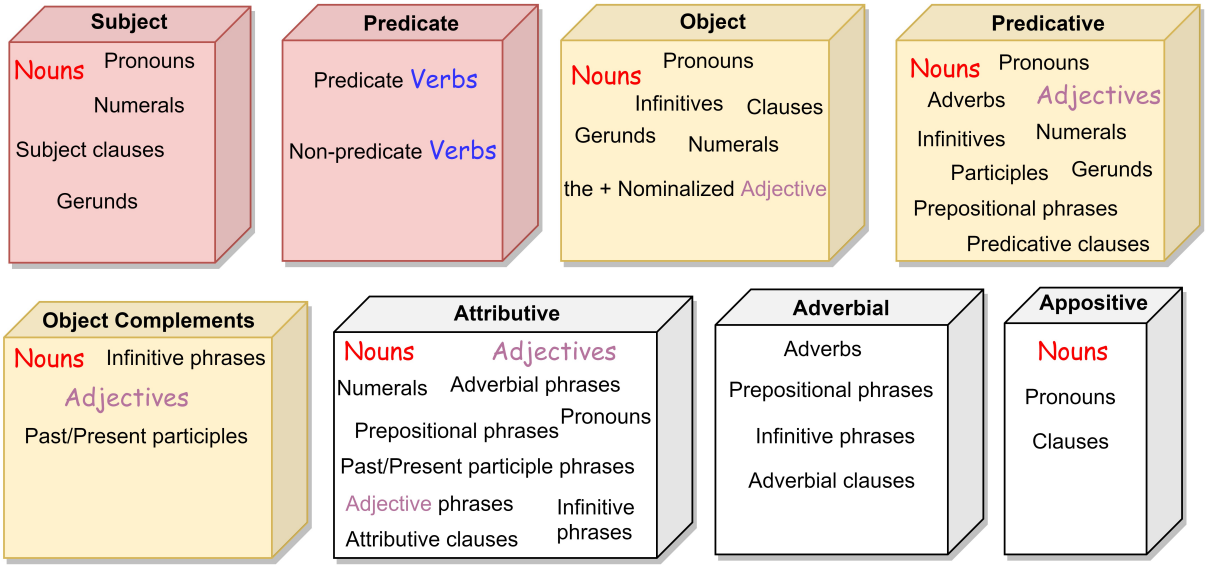


Figure 1: The different syntactic functions that some words or structures perform in sentences. *Subject* and *Predicate* are two core constituents in every sentence. *Object*, *Predicative* and *Object Complements* are three important constituents in some sentence patterns, and the absence of these constituents may result in semantic deficiencies and grammatical errors. *Attributive*, *Adverbial* and *Appositive* are the modifying elements in sentences. The absence of these three constituents will only make the sentence less rich in expression and will not cause grammatical errors.

(e.g., "is" and "are"), etc., and it can be customized by attackers manually.

3.2.1 Tagging and Filtering

Tagging tokens with POSs. The basis of our method is to tag all tokens with POSs accurately, and line 2 in Algorithm 1 shows this process. In daily use, there are 9 POSs (*Verbs, Nouns, Adjectives, Adverbs, Determiners, Prepositions, Pronouns, Conjunctions, Interjections*) which help us make a sentence, and NLTK (Bird et al., 2009) provides 36 tags[2] to label the above 9 POSs more specifically.

Filtering tokens. We use the condition (line 4 in Algorithm 1) to pick out tokens that meet our demands. Since our method can filter out most of the irrelevant tokens which are semantically meaningless, there is no need to rank the importance of tokens as in other studies, and we treat the reserved tokens equally.

3.2.2 Finding, Adjusting and Substituting

Finding and Adjusting substitution words. WordNet (Miller, 1998) can obtain all synonyms[3] or hyponyms[4] of the target token (line 7 in Algorithm 1). To avoid making grammar mistakes, we use Pattern (De Smedt and Daelemans, 2012) to adjust substitution words. Since the specific tags of POS are given in step (1), we can change the *inflection* (tenses, singular or plural, comparative or superlative, etc.) of *Verbs, Nouns, Adjectives, Adverbs* according to the pos_k tag.

Substituting tokens. For each token, we replace it with one adjusted synonym or hyponym in each substitution attempt, and choose one word for the final substitution according to the feedback information (output of victim models on the modified sample). In Algorithm 1, we use the condition ($rel' \neq rel$) in line 9 to choose a word that completely changes the result of relation classification, or use the condition in line 11 to choose a word that changes the confidence of the victim model significantly in judging correct relations.

3.3 Complete Process

To illustrate the complete process of our method, we give an example in Figure 2.

As shown in Figure 2, we first label each token with *pos* tag. Then we filter out *linking verb* (part of F in Algorithm 1), two entities ("Mary"

[2] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[3] WordNet uses the concept of *cognitive synonyms* (synsets) to group and find synonyms, and synsets are presented in the database with lexical and semantic relations.

[4] According to the hierarchical relationship between synsets in WordNet, the hyponyms sets of *nouns* and *verbs* can be found in the tree structure.

Algorithm 1: Generate an adversarial example by substituting tokens with synonyms or hyponyms

Input: Sample: x; Types of tokens to be modified L; Victim model V.
Output: Modified sample x'.

Initialize Forbidden set F, Entity set $E = \{h, t\}$, Relationship rel between h and t;
Get the POS tag pos_k for each token $t_k \in T = \{t_1, t_2, \ldots, t_n\}$;
for *each token* $t_k \in T$ **do**
  if $pos_k \in L$ *and* $t_k \notin F$ *and* $t_k \notin E$ **then**
    $max \longleftarrow None$; $score \longleftarrow 0$; $i_k \longleftarrow$ Get the index of token t_k in sample x;
    Get the confidence $conf$ that the model V infers the relationship as rel;
    Get the set $Y_k = \{y_{k1}, y_{k2}, \ldots, y_{km}\}$ containing synonyms and hyponyms of token t_k;
    for *each word* $y_{kn} \in Y_k$ **do**
      $y'_{kn} \longleftarrow$ Adjust y_{kn} according to the tag pos of token t; $t_k \longleftarrow y'_{kn}$; Get prediction result $(rel', conf')$ of sample x' from V; **if** $rel' \neq rel$ **then**
        return x';
      else if $conf' - conf > score$ **then**
        $score \longleftarrow conf' - conf$; $max \longleftarrow y'_{kn}$; Restore the token indexed as i_k with t_k;
    $t_k \longleftarrow max$;
return *None*

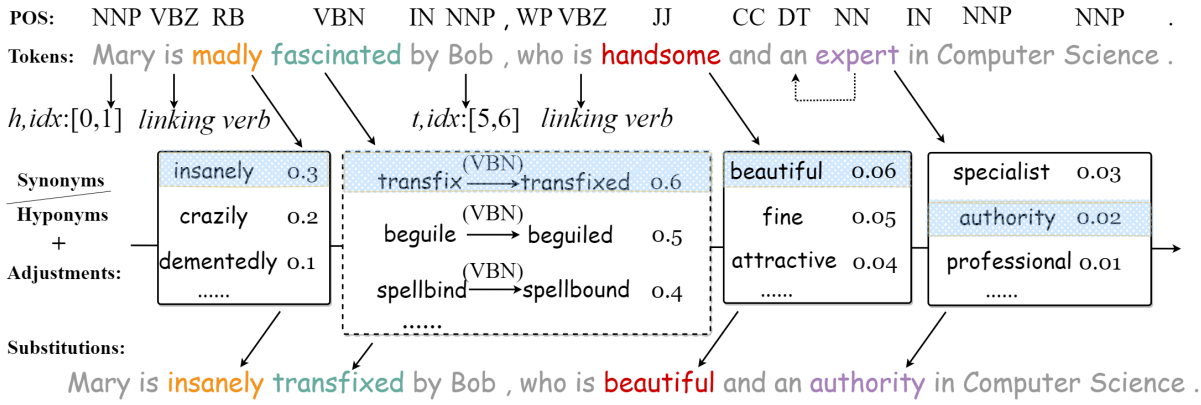


Figure 2: An example to illustrate our method. The workflow is oriented from up to down, left to right.

and "Bob", *idx* represents the index of the entity in sample) and other *proper nouns* (they are labeled with "NNP" in singular form or "NNPS" in plural form. However, they should not be perturbed for the sake of correctness[5], but many related works[6] usually ignore this). Next, we obtain synonyms and hyponyms of each chosen token, and adjust substitution words in order (adjust candidate words of "fascinated" according to the "VBN" tag). Words in the blue box have the most impact on victim models, and the values of confidence reduction are shown behind each word. Thus, we substitute tokens with these adjusted words (since the *Indefinite Article* "an" precedes the *Noun* "expert", we should choose "authority" that begins with a vowel).

4 Experiments and Analysis

To verify the effectiveness of our method, the experiments are conducted on 2 datasets, and 3 DL models are been used for the task of relation classification. Furthermore, we explore the most frequent POS tags of changed tokens in adversarial texts, and compare the similarity of syntactic structure between adversarial and original texts. To study the transferability, we cross-validate the attack effectiveness of adversarial examples against different

[5]For example, despite "lingo" being one synonym of "language", "natural lingo processing" does not represent a branch of computer science, and *proper nouns* are often something with a unique name.

[6]Method in (Tan et al., 2020), (Shi and Huang, 2020), etc.

target models. Besides, we invite volunteers to evaluate the quality of adversarial texts, and use our adversarial examples to improve the robustness of victim models.

4.1 Datasets and Victim Models

Wiki80 (Han et al., 2018) and **TACRED** (Zhang et al., 2017) are two datasets used in the experiments. The **Wiki80** dataset contains 80 relations derived from Wikipedia, and it has 50400 sentences for training, and 5600 sentences for validation (also for testing). **TACRED** (**TAC** **R**elation **E**xtraction **D**ataset, a dataset built over newswire, broadcast material, and web text collected by Linguistic Data Consortium[7]) contains 42 relations, and 68124/22631/15509 samples for training/validation/testing separately.

OpenNRE (Han et al., 2019) is a unified framework (composed of *Tokenization, Module, Encoder, Model, Framework* parts) for relation extraction, and we pre-train **Bert** (Devlin et al., 2019), **BertEntity** (Baldini Soares et al., 2019), and **PCNN** (Zeng et al., 2015) in the *Encoder* module to embed text for providing semantic features. More implementation details are introduced in Section A.

4.2 Evaluation Metrics

We use **Accuracy** (*Acc*) to evaluate the performance of victim models. **Accuracy** is the percentage of correctly classified samples in all samples.

Attack Success Rate (*ASR*) is used to evaluate the attack effectiveness, and it is the percentage of samples successfully fooling models in all samples.

Formal Similarity (*Dis*), **Semantic Similarity** (*Sem*), **Perplexity** (*Flu*), **Word Modification Rate** (*WMR*) and **Grammatical Errors** (*Err*) are 5 metrics used for evaluating adversarial examples. **Formal Similarity** and **Semantic Similarity** are used for measuring differences between adversarial and original texts. **Formal Similarity** (Levenshtein et al., 1966) is measured by the Levenshtein edit distance, and **Semantic Similarity** is measured by the Universal Sentence Encoder (USE) (Cer et al., 2018). Besides, **Perplexity** (Radford et al., 2019) is a metric to evaluate the fluency of adversarial examples. **Word Modification Rate** is the percentage of changed tokens in all tokens, and **Grammatical Errors** are checked by Ginger[8].

Table 1 lists the evaluation metrics and gives the judging criteria.

[7] https://catalog.ldc.upenn.edu/LDC2018T03

[8] https://www.gingersoftware.com

Metric	Acquisition Method	Superior
Acc	Statistics	↑
ASR	Statistics	↑
Dis	Levenshtein	↓
Flu	GPT2LM	↓
Sem	USE	↑
WMR	Statistics	↓
Err	Ginger	↓

Table 1: Evaluation metrics and the method of acquiring calculation results. ↑ means the performance is better if acquiring a higher value, and the opposite is true for ↓.

Dataset	Victim Model		
	Bert	BertEntity	PCNN
Wiki80	86.21%	87.09%	67.14%
TACRED	87.57%	88.45%	80.64%

Table 2: Classification accuracy of victim models.

4.3 Results

4.3.1 Classification Performance

The training epoch of "Bert", "BertEntity" and "PCNN" models are 20, 20, 200 separately, and the **Accuracy** of each model on the testing sets are shown in Table 2.

As shown in Table 2, although the classification performance of the "PCNN" model is not as good as that of "Bert" and "BertEntity" (especially when the number of relations is 80), "PCNN" can correctly classify the relation in 80.64% samples from the **TACRED** dataset, and the classification accuracy of other models are all 85% above.

4.3.2 Attack Performance

We choose *Nouns* ("N"), *Verbs* ("V"), *Adjectives* ("J"), and *Adverbs* ("R") as target POSs (L in Algorithm 1), and we combine parts or all of the above word types to test our method in the subsequent experiments. Besides, we compare the attack performance with 4 methods ("PWWS", "Gen", "TB", "HF") which are introduced in Section 2. The average performance of these methods on 2 datasets are summarized in Table 3, and the complete experimental results are shown in Table 9, 10 and 11 in Appendix B.

As shown in Table 3, the attack effectiveness of our method is not as good as "PWWS" and "TB", but better than that of other attacking methods. To achieve higher values of *ASR*, the minimum threshold value of substituting words is not set in our

Method	Wiki80						TACRED					
	ASR	Dis	Flu	Sem	WMR	Err	ASR	Dis	Flu	Sem	WMR	Err
PWWS	**38.11%**	23.86	628.49	84.58%	43.45%	5.20	**19.78%**	32.64	587.08	86.16%	36.26%	5.77
TB	34.00%	20.00	938.21	75.95%	78.52%	8.55	17.40%	28.45	810.19	78.95%	77.42%	10.90
Gen	14.51%	20.10	504.52	89.75%	37.68%	5.01	8.11%	25.23	344.63	**91.92%**	28.82%	5.57
HF	13.15%	34.10	927.54	82.71%	47.39%	5.05	4.08%	36.89	556.78	86.90%	33.60%	5.40
NV	30.83%	**17.66**	**339.16**	**91.19%**	**12.06%**	**0.57**	15.65%	**24.40**	**282.95**	91.33%	**13.18%**	**0.74**
NVR	31.14%	18.74	353.37	91.05%	13.05%	0.60	15.73%	25.47	315.08	91.24%	13.92%	0.79
NVJ	32.31%	19.82	357.27	90.36%	14.39%	0.60	15.92%	26.77	294.09	90.61%	16.26%	0.77
NVJR	32.57%	20.89	372.01	90.23%	15.42%	0.63	16.03%	28.00	325.82	90.45%	17.01%	0.81

Table 3: Average attack performance of different methods on 2 datasets against 3 victim models. Values in the red color represent the best result, and the last 4 rows contain all of our methods.

Dataset	Method			
	Gen	HF	PWWS	TB
Wiki80	*NOUN: 51.7%*	*NOUN: 35.9%*	*NOUN: 61.5%*	*NOUN: 55.7%*
	VERB: 26.6%	*PREP: 21.5%*	*VERB: 17.7%*	*VERB: 18.6%*
	ADJ: 15.0%	*VERB: 20.8%*	*ADJ: 17.6%*	*ADJ: 16.8%*
	ADV: 4.0%	*ADJ: 10.7%*	*ADV: 1.9%*	*ADV: 3.0%*
TACRED	*NOUN: 53.2%*	*NOUN: 40.7%*	*NOUN: 58.4%*	*NOUN: 55.8%*
	VERB: 28.2%	*VERB:19.3%*	*VERB: 19.8%*	*VERB: 20.6%*
	ADJ: 12.1%	*PREP: 16.6%*	*ADJ: 17.4%*	*ADJ: 15.8%*
	ADV: 4.6%	*ADJ: 11.8%*	*ADV: 2.5%*	*ADV: 3.1%*

Table 4: Proportional distribution of Top-4 POSs.

method, thus the values on **Semantic Similarity** are a little inferior to "Gen", but the overall quality of adversarial examples generated by our method is superior to that of "Gen" and others. Among our methods, the combination ("NVJR") achieves the highest *ASR*, and the combination ("NV") generates adversarial examples of the highest quality. Besides, the impact of adding *Adverbs* on the final results is not as great as that of adding *Adjectives*.

4.4 Further Analysis

4.4.1 POSs Analysis

We count the POS tags for perturbed words in 2 datasets, and then analyze the distribution of POSs. The Top-4 POSs and their proportional distribution are shown in Table 4.

We find that most of the adversarial attacks happen to *Nouns*, *Verbs*, and *Adjectives* in 3 methods ("Gen", "PWWS", "TB"), while *Prepositions* (*PREP* that appears in Table 4) are also under great attack in "HF". In general, the results are consistent with the natural distribution of lexical patterns analyzed in Section 3.1.

4.4.2 Dependency Parsing

To evaluate our method from the syntax perspective, we apply Dependency Parsing to measure the **Similarity** of syntactic structure between adversarial and original texts, and the consistency of Dependency Parsing is evaluated by **Unlabeled Attachment Score** (*UAS*) and **Labeled Attachment Score** (*LAS*). The dependency-based trees are parsed by Stanza (Qi et al., 2020). Furthermore, we also count the **Overlap** of perturbed tokens and syntactic heads, and study the dependency **Relation**[9] between entities and their syntactic heads. We give examples for illustration in Figure 7 and 8. All statistics are shown in Table 5.

The results shown in Table 5 demonstrate the high similarity of syntactic structure between adversarial and original texts, which means that our method preserves the original dependency relations between different tokens well. Besides, more than 60% of **Root** tokens and around 50% **syntactic heads** of two entities are perturbed. Among all **syntactic relations**, *nmod* (nominal modifier), *nsubj* (nominal subject) and *obl* (oblique nominal) are Top-3 relations to the perturbed head.

4.4.3 Human Evaluation

We conduct a two-stage (**STAGE 1** and **STAGE 2**) experiment on 2 datasets, and the screenshots of partial questionnaires are shown in Appendix C.

STAGE 1. We randomly sample 60 instances from both datasets which are successfully attacked by our method ("NV") and others ("Gen", "PWWS", "TB", "HF"). For each instance, we present adversarial examples generated by the five methods along with the original input to our volunteers, and ask them to rate generated samples in terms of grammaticality and fluency. Each sentence is scored on a scale of 0-5, while 0 is the worst and 5 is the best.

As shown in Table 6, "NV" gets an average score of **3.52** in **STAGE 1**, which is superior to that of other methods. Among the 60 adversarial examples generated by "NV", 47 scored higher than or equal to 3 points. This result shows that adversarial

[9] https://universaldependencies.org/en/dep/index.html

Dataset	Victim Model	Dependency Parsing						
		Similarity		Overlap			Relation	
		UAS↑	LAS↑	Root	Head	Tail	Head	Tail
Wiki80	**Bert**	93.88%	92.69%	67.41%	56.22%	53.86%	*nmod, nsubj, obl*	*obl, nmod, compound*
	BertEntity	94.05%	92.91%	70.11%	57.61%	58.98%	*nmod, nsubj, appos*	*obl, nmod, compound*
	PCNN	94.71%	93.42%	67.60%	51.87%	55.61%	*nsubj, nmod, obl*	*obl, compound, nmod*
TACRED	**Bert**	93.12%	91.64%	62.40%	47.69%	43.79%	*nsubj, nmod, nmod:poss*	*nsubj, obl, nmod*
	BertEntity	92.90%	91.45%	61.76%	52.24%	46.59%	*nsubj, nmod, nmod:poss*	*obl, nsubj, nmod*
	PCNN	93.16%	91.84%	62.18%	41.97%	42.49%	*nsubj, nmod:poss, nmod*	*nsubj, amod, nmod*

Table 5: Statistics of dependency parsing. **Head** and **Tail** represent the syntactic head of the **Head** entity and **Tail** entity respectively. ↑ means the performance is better if acquiring a higher value.

Result	Method				
	NV	**PWWS**	**Gen**	**TB**	**HF**
Average score	**3.52**	3.05	3.03	2.75	2.30
Score >= 3	**78%**	67%	67%	55%	35%

Table 6: Human evaluation results of **STAGE 1**.

Dataset	Victim Model (V_a)	Target Model		
		Bert	**BertEntity**	**PCNN**
Wiki80	**Bert(1439)**	-	43.71%	57.61%
	BertEntity(1392)	48.35%	-	57.61%
	PCNN(1284)	14.80%	14.95%	-
TACRED	**Bert(3508)**	-	23.66%	40.76%
	BertEntity(2050)	54.49%	-	53.56%
	PCNN(772)	15.41%	15.80%	-

Table 7: Transferability of adversarial examples. The adversarial examples (total number is V_a) are generated against **Victim Models**, and are transferred to mislead **Target Models**. Note that *ASR* in the study of transferability indicates the percentage of samples successfully fooling **Target Models** in V_a.

examples generated by our method have stable and good performance in terms of grammaticality and fluency, and they are more likely to be written by a human rather than a machine.

STAGE 2. We conduct relation-type annotation on the **TACRED** dataset, and randomly sample 120 instances that are successfully attacked by "NV". For each adversarial example, we mark the *head entity* and *tail entity*, present the *TRUE relation type* along with 3 *relevant relation types* (extracted from the 40 relation types of **TACRED**) to our volunteers, and ask them to annotate the best match. Among the 120 adversarial examples, accounting for about **85.3%** of samples are correctly annotated, which indicates "NV" is good at misleading the victim models while preserving human predictions.

4.4.4 Transferability

We use adversarial examples generated against different models to mislead each other, and the average attack performances of our method ("NV") on 2 datasets are shown in Table 7.

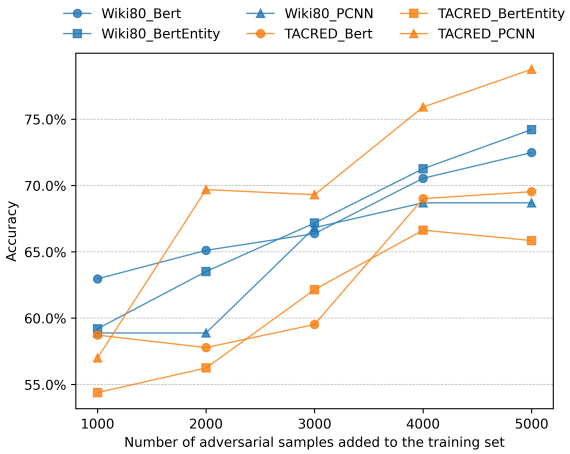


Figure 3: Classification accuracy of victim models on adversarial sets. The names of source datasets and victim models are connected with the symbol "_".

As shown in Table 7, the adversarial examples generated against the "BertEntity" model possess stronger (achieve higher values of *ASR*) transferability, while those generated against the "PCNN" model possess more stable (the variance of *ASR* is approximately 0.21) transferability.

4.4.5 Adversarial Training

Adversarial Training (Goodfellow et al., 2015) is a method to confront adversarial attacks, and it uses adversarial examples to improve the robustness of target models. For further analysis, we generate another 5000 adversarial examples using our method ("NV"), and add them into original training sets to retrain (the settings of the training epoch are the same as that in Section 4.3.1) victim models. Figure 3 shows the classification performance of 3 victim models on adversarial sets (each adversarial set contains all of the adversarial examples generated by "NV" in Section 4.3.2).

With the gradual increase in the number of adversarial examples, the accuracy of victim models on adversarial sets shows an increasing trend (see

Attack Method	Word-Level			Character/Word-Level	
	PWWS	Gen	**Ours**	TB	HF
Perturbation Options	*Sub*	*Sub*	***Sub***	*Ins, Del, Swap, Sub-C, Sub-W*	*Flip, Ins, Del, Sub*
Importance Ranking	*Required*	*Not Required*	***Not Required***	*Required*	*Not Required*
Extra Requirements	*Word saliency*	*GloVe vectors, Counter-fitting method, Google 1 billion words language model*	/	*Pre-trained GloVe model*	*One-Hot input representation, Word Embedding*

Table 8: Comparison of prerequisites and settings in different methods.

6 lines in Figure 3), and when the number reaches 4000, the classification accuracy of all models exceeds 65%. However, the process of Adversarial Training requires sufficient samples, and when the number of adversarial examples is 1000, the classification accuracy of all models lower than 65%. Thus, when the number of adversarial examples in the updated training set is relatively small, the robustness of DL models may not improve significantly.

5 Discussions

Despite our method can generate adversarial examples of high quality, there are still some limitations in this paper. In this section, we talk about the prominent advantage and significant limitations of our method.

5.1 Advantages

The prominent advantage of our method is *Simplicity*. We list the prerequisites and settings in different methods in Table 8.

As shown in Table 8, methods in **Word-Level** need fewer options to perturb targets in texts, and our method needs no extra requirements in the implementation. Furthermore, since we choose tokens labeled with the most important POS tags and treat them equally, there is no need to rank the importance of tokens.

Since textual adversarial attack task is a tradeoff between attack effectiveness, perception of grammar (Morphology, Syntax, Semantics) errors, complexity (of implementation), and so on, this simplicity brings the balanced development of our method on the above factors.

5.2 Limitations

The significant limitations of our method are *Inferior ASR* and *Unknown Generalization*.

Inferior ASR. Many reasons lead to this, such as no tokens can be substituted according to the principle of our method, or the predicted relation cannot be changed after all replacements, etc.

Unknown Generalization. The generalization of our method in different languages, tasks, etc. is unknown. Our work only considers the text in English, and it is mainly aimed at classification tasks.

Besides, although there will be losses on *ASR*, *Semantic Similarity Checking* (Jin et al., 2020) is still an essential method to retain original semantics, and help realize[10] the real-world attacks mentioned in (Chen et al., 2022) (i.e., the confidence score $conf$ is limited, and attackers can only access the victim models' decisions). Moreover, to further reduce modifications on paragraphs or long documents, **Root** tokens and **syntactic heads** of two entities may have priority over other tokens to be substituted.

6 Conclusion

In this paper, a simple but effective method for generating adversarial examples and misleading relation classifiers is proposed. We first analyze the most important POSs from the perspective of linguistics. To reduce modification on original samples and avoid grammar mistakes, our method uses adjusted synonyms or hyponyms to substitute tokens labeled with the most important POS tags. Experimental results show the adversarial examples generated by our method have better quality (both evaluated by machines and humans). Furthermore, these samples possess promising transferability, and they are also helpful to improve the

[10]Attackers can greedily choose substitution words that are closest to the threshold of minimum semantic similarity between original and perturbed sentences.

robustness of victim models. We will further optimize our method and test it in other NLP tasks (e.g., sentiment analysis, textual entailment) in future work.

Acknowledgements

We thank all volunteers for their efforts in human evaluation, and the anonymous reviewers for their valuable feedback.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Yangyi Chen, Hongcheng Gao, Ganqu Cui, Fanchao Qi, Longtao Huang, Zhiyuan Liu, and Maosong Sun. 2022. Why should adversarial perturbations be imperceptible? rethink the research paradigm in adversarial NLP. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11222–11237, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Noam Chomsky. 2014. *Aspects of the Theory of Syntax*, volume 11. MIT press.

Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *The Journal of Machine Learning Research*, 13(1):2063–2067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

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

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. 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): System Demonstrations*, pages 169–174, Hong Kong, China. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Xuanli He, Lingjuan Lyu, Lichao Sun, and Qiongkai Xu. 2021. Model extraction and adversarial transferability, your BERT is vulnerable! In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2006–2012, Online. Association for Computational Linguistics.

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.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021a. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *ArXiv preprint*, abs/1612.08220.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Luoqiu Li, Xiang Chen, Zhen Bi, Xin Xie, Shumin Deng, Ningyu Zhang, Chuanqi Tan, Mosha Chen, and Huajun Chen. 2021b. Normal vs. adversarial: Salience-based analysis of adversarial samples for relation extraction. In *IJCKG'21: The 10th International Joint Conference on Knowledge Graphs, Virtual Event, Thailand, December 6 - 8, 2021*, pages 115–120. ACM.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 390–395, Online. Association for Computational Linguistics.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

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.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Zhouxing Shi and Minlie Huang. 2020. Robustness to modification with shared words in paraphrase identification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 164–171, Online. Association for Computational Linguistics.

Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. It's morphin' time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2935, Online. Association for Computational Linguistics.

Jiayi Wang, Rongzhou Bao, Zhuosheng Zhang, and Hai Zhao. 2022. Distinguishing non-natural from natural adversarial samples for more robust pre-trained language model. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 905–915, Dublin, Ireland. Association for Computational Linguistics.

George Yule. 2020. *The study of language*. Cambridge university press.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via

piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371, Online. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

A Additional Implementation Details

We expand our work based on the DiagnoseAdv[11] project. All comparison models and evaluation methods are implemented by using the interfaces provided by different modules (*Attacker* and *Metric*) in OpenAttack[12]. Besides, several tools (NLTK[13], WordNet[14], Pattern[15], Stanza[16]) are used in the experiments. We keep all the default configurations given in OpenAttack and these tools.

All relation extraction models are built, trained, and tested through OpenNRE[17]. The *max length* of input sentences is set to 128, the *batch size* is set to 32, the *learning rate* is set to 2e-5, and we use the AdamW (Loshchilov and Hutter, 2019) *optimizer*. Besides, the value of *dropout* in the "PCNN" encoder is set to 0.5. The training (including initial and adversarial training) epoch of "Bert", "BertEntity" and "PCNN" models are 20, 20, 200 respectively. In addition, the settings of other parameters in each module are kept default.

B Complete Attack Performance

The complete attack performance of "PWWS", "TB", "Gen", "HF" and our methods (including "NV", "NVR", "NVJ" and "NVJR") against 3 victim models are shown in Table 9, 10 and 11.

[11] https://github.com/zxlzr/DiagnoseAdv

[12] https://github.com/thunlp/OpenAttack

[13] https://github.com/nltk/nltk

[14] https://github.com/nltk/nltk/tree/develop/nltk/corpus

[15] https://github.com/clips/pattern

[16] https://github.com/stanfordnlp/stanza

[17] https://github.com/thunlp/OpenNRE

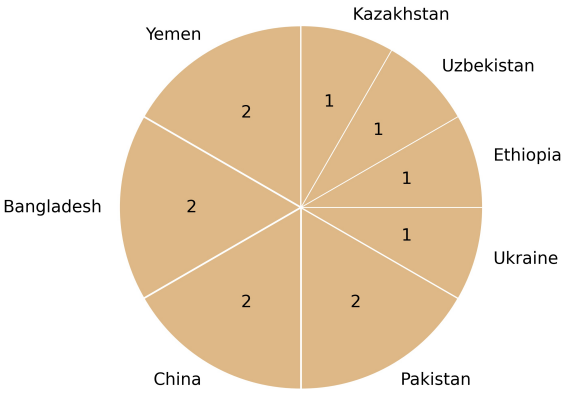


Figure 4: The distribution of nationalities among 12 volunteers. Furthermore, the mother tongues of volunteers cover all four major language families (including *Indo-European, Sino-Tibetan, Semito-Hamitic* and *Altaic* languages).

C Human Evaluation Details

Most of the volunteers are graduate students (some of them are Ph.D. candidates), and the statistics of their nationalities are shown in Figure 4. Figures 5 and 6 are questionnaires that the volunteers are invited to finish in different stages.

Before starting **STAGE 1**, we emphasize to volunteers that they should compare each generated text with the reference text, rather than comparing generated texts with each other.

Before starting **STAGE 2**, we include misclassified labels when designing the questionnaire. We count the top 10 misclassified labels (M_{10}) by the victim model for each label, then randomly (to avoid duplication) select 3 wrong options from M_{10} for each question.

D Case Study

The parsed dependency tree of an example is visualized by the interactive web-based demo[18], and shown in Figure 7. We also give an example (see Figure 8) to introduce the Root word, syntactic heads of entities, and dependency relation in a sentence. Some adversarial examples generated by different methods are shown in Table 12 and 13.

[18] http://stanza.run/

Method	Wiki80						TACRED					
	ASR	Dis	Flu	Sem	WMR	Err	ASR	Dis	Flu	Sem	WMR	Err
PWWS	**34.40%**	24.21	648.04	83.95%	44.15%	5.23	**29.34%**	32.96	512.49	86.25%	36.27%	5.93
TB	30.10%	19.87	936.60	76.57%	73.98%	8.20	25.97%	27.56	781.14	81.75%	70.14%	9.86
Gen	13.94%	20.70	516.27	89.62%	37.97%	5.10	13.50%	26.97	368.46	**91.42%**	29.48%	5.77
HF	12.55%	35.17	975.18	81.00%	48.34%	5.09	5.88%	38.31	567.73	86.17%	34.28%	5.53
NV	29.81%	**17.85**	**342.37**	**91.00%**	**12.14%**	**0.59**	25.83%	**25.46**	**275.21**	91.13%	**13.41%**	**0.81**
NVR	29.97%	18.94	356.95	90.83%	13.21%	0.62	25.99%	26.63	286.01	91.04%	14.15%	0.84
NVJ	31.09%	20.02	362.64	90.09%	14.14%	0.62	26.07%	27.58	285.36	90.49%	16.72%	0.82
NVJR	31.19%	21.07	378.48	89.93%	15.26%	0.65	26.27%	28.82	296.72	90.38%	17.49%	0.85

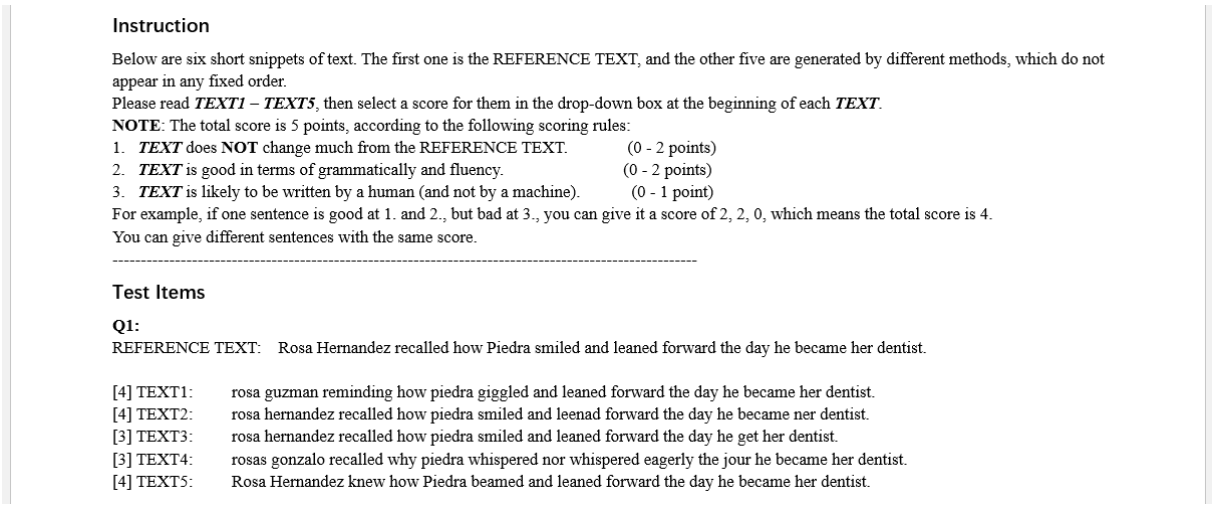
Table 9: The attack performance of different methods in generating adversarial examples against the "**Bert**" model.

Method	Wiki80						TACRED					
	ASR	Dis	Flu	Sem	WMR	Err	ASR	Dis	Flu	Sem	WMR	Err
PWWS	**32.79%**	24.59	660.85	83.90%	44.44%	5.24	**19.37%**	30.91	542.86	86.31%	36.04%	5.79
TB	28.85%	20.25	986.84	74.87%	77.06%	8.33	16.64%	29.10	860.45	78.47%	74.99%	10.60
Gen	12.12%	20.85	525.69	89.34%	38.07%	5.00	7.22%	26.00	364.15	91.28%	29.28%	5.74
HF	10.99%	34.91	993.70	80.95%	48.44%	5.09	4.29%	42.19	734.48	84.44%	36.76%	5.35
NV	28.54%	**18.13**	**357.90**	**90.67%**	**12.48%**	**0.55**	14.94%	**24.15**	**275.77**	**91.41%**	**13.44%**	**0.76**
NVR	28.81%	19.12	371.86	90.50%	13.55%	0.59	15.00%	25.20	287.31	91.22%	14.46%	0.80
NVJ	29.59%	20.35	372.73	89.82%	14.69%	0.58	15.37%	26.77	286.80	90.74%	16.11%	0.79
NVJR	29.83%	21.45	387.00	89.67%	15.90%	0.62	15.41%	27.73	297.15	90.57%	16.92%	0.83

Table 10: The attack performance of different methods in generating adversarial examples against the "**BertEntity**" model.

Method	Wiki80						TACRED					
	ASR	Dis	Flu	Sem	WMR	Err	ASR	Dis	Flu	Sem	WMR	Err
PWWS	**47.15%**	22.78	576.58	85.90%	41.77%	5.12	**10.63%**	34.05	705.90	85.92%	36.48%	5.59
TB	43.06%	19.89	891.19	76.42%	84.52%	9.12	9.58%	28.70	788.97	76.63%	87.12%	12.24
Gen	17.47%	18.74	471.60	90.30%	37.00%	4.92	3.60%	**22.71**	301.27	**93.05%**	27.71%	5.21
HF	15.90%	32.23	813.75	86.17%	45.40%	4.97	2.06%	30.18	368.13	90.09%	29.76%	5.33
NV	34.15%	**17.01**	**317.22**	**91.91%**	**11.56%**	**0.57**	6.17%	23.60	**297.86**	91.46%	**12.68%**	**0.66**
NVR	34.63%	18.16	331.29	91.82%	12.39%	0.59	6.21%	24.59	371.93	91.45%	13.15%	0.73
NVJ	36.25%	19.08	336.43	91.18%	14.35%	0.59	6.31%	25.97	310.11	90.59%	15.94%	0.70
NVJR	36.70%	20.16	350.55	91.08%	15.10%	0.61	6.42%	27.44	383.60	90.40%	16.62%	0.76

Table 11: The attack performance of different methods in generating adversarial examples against the "**PCNN**" model.



Instruction

Below are six short snippets of text. The first one is the REFERENCE TEXT, and the other five are generated by different methods, which do not appear in any fixed order.
Please read ***TEXT1*** – ***TEXT5***, then select a score for them in the drop-down box at the beginning of each ***TEXT***.
NOTE: The total score is 5 points, according to the following scoring rules:
1. ***TEXT*** does **NOT** change much from the REFERENCE TEXT. (0 - 2 points)
2. ***TEXT*** is good in terms of grammatically and fluency. (0 - 2 points)
3. ***TEXT*** is likely to be written by a human (and not by a machine). (0 - 1 point)
For example, if one sentence is good at 1. and 2., but bad at 3., you can give it a score of 2, 2, 0, which means the total score is 4.
You can give different sentences with the same score.

Test Items

Q1:
REFERENCE TEXT: Rosa Hernandez recalled how Piedra smiled and leaned forward the day he became her dentist.

[4] TEXT1: rosa guzman reminding how piedra giggled and leaned forward the day he became her dentist.
[4] TEXT2: rosa hernandez recalled how piedra smiled and leenad forward the day he became ner dentist.
[3] TEXT3: rosa hernandez recalled how piedra smiled and leaned forward the day he get her dentist.
[3] TEXT4: rosas gonzalo recalled why piedra whispered nor whispered eagerly the jour he became her dentist.
[4] TEXT5: Rosa Hernandez knew how Piedra beamed and leaned forward the day he became her dentist.

Figure 5: To evaluate the quality of adversarial examples generated by different methods intuitively in **STAGE 1**.

Instruction

Below is a short snippet of text, in which we have underlined 2 entities. The "**(h)**" represents the "head entity", and the "**(t)**" represents the "tail entity". Then we present 4 options, each of them representing a relationship type. Each type takes the form of "**subject: relation**", where there are two types of the subject: "**per**" and "**org**", which stands for "person" and "organization".
NOTE: "NA" represents "no relationship was found". Determine which of the four options best matches the relationship between these 2 entities. Please ignore the punctuation, capitalization, and other minor formatting issues.

Example

Text: Cuba believes Gross**(h)**, who has been delayed for six months, is a spy**(t)**.
[B] A. org: founded B. per: title C. per: date_of_birth D. per: age

---

Test Items

Q1:
Text: In November, Wen's sister-in-law Xie Caiping**(h)**, 46, was foredoomed to 18 geezerhoods in prison on charges ranging from running illegal gambling dens to drug dealing**(t)**.
[A] A. per: charges B. per: employee_of
C. per: cause_of_death D. per: religion

Q2:
Text: It locomoted to Rice University, where de Menil**(t)** and his wife, Dominique de Menil, who later founded the Menil Collection**(h)**, ran the art museum.
[D] A. org: shareholders B. org: founded
C. org: top_members/employees D. org: founded_by

Figure 6: To identify the relations between entities in adversarial texts generated by "NV" in **STAGE 2**.

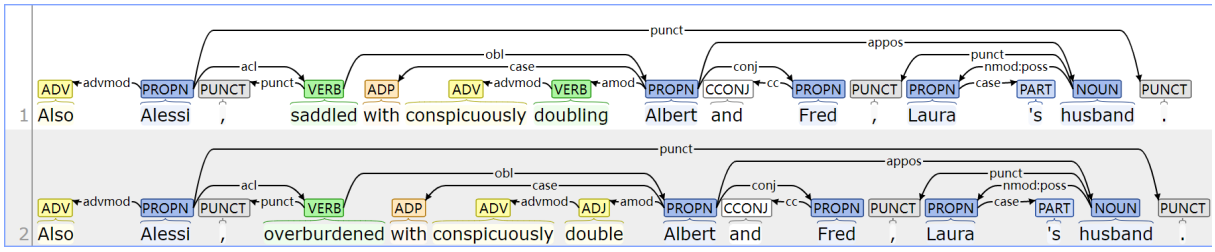


Figure 7: Comparison of dependency parsing. The **upper** tree is parsed from an original sample, another tree is parsed from the adversarial example.

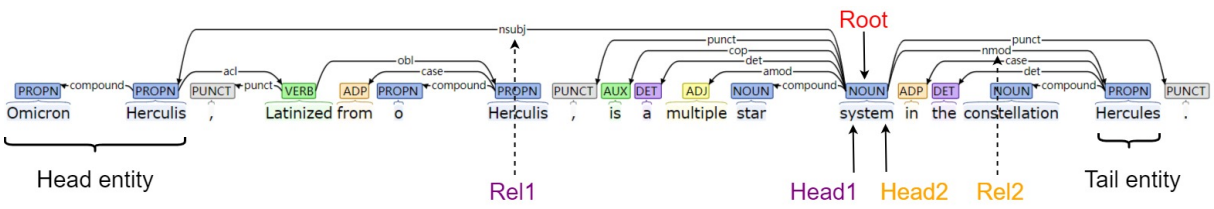


Figure 8: Root word, syntactic heads of Head entity (Head1, the dependency relation is Rel1) and Tail entity (Head2, the dependency relation is Rel2) in the parsed tree.

Example 1 (generated against the *Bert* model)	
Original (Relation = **contains administrative territorial entity**)	Situated on the German border and not far from the German city of Karlsruhe , it is the easternmost commune in Metropolitan France (excluding the island of Corsica) .
PWWS (Prediction = **located on terrain feature**)	situated on the german border and not far from the german city of karlsruhe , it is the easternmost commune in metropolitan france (turn the island of corsica).
TB (Prediction = **has part**)	situated on te german frontiers and not tar from te german town of karlsruhe , it is te easternmost comune in metropolitan france (excluding te island of corsica).
Gen (Prediction = **has part**)	situated on the german border and not far from the german city of karlsruhe , it is the easternmost commune in metropolitan france (excluded the island of corsica).
HF (Prediction = **has part**)	implanted during the german boarder nor never considerably for the german township from karlsruhe , he makes the easternmost township in metropolitan france (excluding the island of corsica).
NV (Prediction = **has part**)	localized on the German margin and not far from the German city of Karlsruhe , it is the easternmost commune in Metropolitan France (excluding the island of Corsica) .
Example 2 (generated against the *BertEntity* model)	
Original (Relation = **after a work by**)	It is loosely based on the novel " The Night Watch " by Sergei Lukyanenko , and is the first part of a duology , followed by " Day Watch " .
PWWS (Prediction = **screenwriter**)	it is loosely based on the novel " t night watch " by sergei lukyanenko , and is the foremost part of a duology , followed by " day watch ".
TB (Prediction = **screenwriter**)	it is loosely bases on fhe novel " te night watch " by sergei lukyanenko , and is fhe frist pat of a duologj , followd by " day watch ".
Gen (Prediction = **screenwriter**)	it is loosely based on the newer " the night watch " by sergei lukyanenko , and is the first part of a duology , followed by " day watch ".
HF (Prediction = **screenwriter**)	he makes freely based on the novel " the night watch " by sergei lukyanenko , and is the first part of a duology , followed by " day watch ".
NVR (Prediction = **screenwriter**)	It is slackly based on the romance " The Night Watch " by Sergei Lukyanenko , and is the first part of a duology , followed by " Day Watch " .

Table 12: Adversarial examples generated on the **Wiki80** dataset. The first and second entities are marked in blue and orange respectively. The original relation between the two entities is marked in **green**, and the prediction result of the victim model is **bold**. All perturbations are marked in red, and ignore extra spaces and indents.

Example 1 (generated against the *BertEntity* model)	
Original (Relation = **per:title**)	Benjamin Chertoff - 25-year-old cousin of Michael Chertoff ; senior researcher " for Popular Mechanics ' hit piece on 9-11 Truth Movement
PWWS (Prediction = **NA**)	benjamin chertoff - 25 - year - old first of michael chertoff ; senior researcher " f popular mechanics ' hit piece on 9 - 11 truth movement
TB (Prediction = **NA**)	benjamin chertoff - 25 - year - old cousin of michael chertoff ; sineor researcher " for popular mechanics ' hit oiece on 9 - 11 truth movement
Gen (Prediction = **NA**)	benjamin chertoff - 25 - year - old cousin of michael chertoff ; elders researcher " for popular mechanics ' hit piece on 9 - 11 truth movement
HF (Prediction = **NA**)	benjamin chertoff - 25 - sunni - immemorial kinsman of michael chertoff ; senior researcher " for popular mechanics ' hit piece on 9 - 11 truth movement
NVJ (Prediction = **NA**)	Benjamin Chertoff - 25-year-old cousin-german of Michael Chertoff ; precedential researcher " for best-selling Mechanics ' striking piece on 9-11 Truth Movement
Example 2 (generated against the *PCNN* model)	
Original (Relation = **org:top_members/ employees**)	Clayton also was hands-on as he helped archive of one of the largest collections of African-Americana , " said Sue Hodson , director of literary manuscripts at the Huntington Library in San Marino , Calif .
PWWS (Prediction = **NA**)	clayton also was hands - on as he helped archive of one of the largest collections of african - americana , " said sue hodson , theater of literary manuscripts at the huntington library in san marino , calif.
TB (Prediction = **NA**)	clyton aso ws h ands - on as he helped archive of one of the largest collections of african - americana , " said sue hodson , director of literary manuscripts at the huntington library in sn marino , calif.
Gen (Prediction = **NA**)	clayton also was hands - on as he aided archive of one of the largest collections of african - americana , " said sue hodson , director of literary manuscripts at the huntington library in san marino , calif.
HF (Prediction = **NA**)	clay additionally was hands - on as he helped archive of one of the largest collections of african - americana , " said sue hodson , director of literary manuscripts at the huntington library in san marino , calif.
NVJR (Prediction = **NA**)	Clayton likewise was hands-on as he aided archive of one of the largest collections of African-Americana , " said Sue Hodson , director of literary manuscripts at the Huntington Library in San Marino , Calif .

Table 13: Adversarial examples generated on the **TACRED** dataset. **NA** represents no relation was found.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 5.2

☑ A2. Did you discuss any potential risks of your work?
Section 1, Section 4.4.4, Section 4.4.5

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1, Section A in the supplemental material, Section 3.2

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1, Section A in the supplemental material, Section 3.2

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
All public datasets used in this paper were either open-sourced or released by the original authors, and we were unable to find the license for the dataset we used.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The intended use of two datasets is to benchmark relation extraction models, and train knowledge base population systems. We study the adversarial attack in the relation classification scenario, thus it is consistent with their original use in this paper.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Neither of the authors of the two datasets discusses this in their work, and we do not consider it independently.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☒ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 4.4.3, Section C in the supplemental material

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
We just warned the participants that there might be improper content (misunderstanding due to cultural differences) orally, and we didn't make an instruction file given to volunteers independently.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Most of the volunteers are international students, and we teach them Chinese as a kind of "payment".

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
We informed volunteers that their human evaluation results will be collected and analyzed only for scientific paper writing.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
We have no data to collect, and we analyze the results only based on the annotation in questionnaires.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We gave the nationality information of volunteers in Section C in the supplemental material.