Adaptive Immune-based Sound-Shape Code Substitution for Adversarial Chinese Text Attacks

Ao Wang[1], Xinghao Yang[1], Chen Li[2], Baodi Liu[1], Weifeng Liu[1]*,

[1]China University of Petroleum (East China), [2]Renmin University of China,
wanga@s.upc.edu.cn, yangxh@upc.edu.cn
lichen621@ruc.edu.cn, thu.liubaodi@gmail.com, liuwf@upc.edu.cn

Abstract

Adversarial textual examples reveal the vulnerability of natural language processing (NLP) models. Most existing text attack methods are designed for English text, while the robust implementation of the second popular language, i.e., Chinese with 1 billion users, is greatly underestimated. Although several Chinese attack methods have been presented, they either directly transfer from English attacks or adopt simple greedy search to optimize the attack priority, usually leading to unnatural sentences. To address these issues, we propose an adaptive Immune-based Sound-Shape Code (ISSC) algorithm for adversarial Chinese text attacks. Firstly, we leverage the Sound-Shape code to generate natural substitutions, which comprehensively integrate multiple Chinese features. Secondly, we employ adaptive immune algorithm (IA) to determine the replacement order, which can reduce the duplication of population to improve the search ability. Extensive experimental results validate the superiority of our ISSC in producing high-quality Chinese adversarial texts. Our code and data can be found in https://github.com/nohuma/chinese-attack-issc.

1 Introduction

Deep Neural Networks (DNNs) have shown great vulnerability towards text adversarial examples, primarily in English text (Papernot et al., 2016). This phenomenon poses a great challenge for the security application of DNNs models in text-related tasks, such as sentiment analysis (El Rahman et al., 2019) and toxic comment detection (Abbasi et al., 2022; Bespalov et al., 2023), etc. Therefore, it is essential to devise high quality text adversarial examples to investigate the brittleness boundary and understand the behaviors of modern DNNs models before implementation.

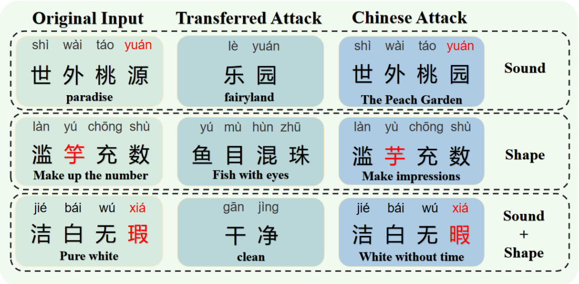

Figure 1: Candidates comparison of English transfer attack and Chinese Sound-Shape Code (SSC) attack. Intuitively, the Chinese SSC generates more natural candidates from the view of Chinese native speakers.

During the past decade, abundant attention has been paid to craft English text adversarial examples, ranging from character-level attack (Gao et al., 2018), word-level attack (Zang et al., 2020; Yu et al., 2022), sentence-level attack (Iyyer et al., 2018), and multi-level attack (Chen et al., 2021; Xu et al., 2024), usually with text insertion, deletion, substitution, and rewriting operations. Owing to the concealment and flexibility, word substitution based text attack methods are gradually becoming the most popular line (Qiu et al., 2022). In this line, two common steps are (1) generating high quality substitution candidates, and (2) optimizing the attack priority. To generate semantic consistency candidates, researchers have tried GloVe embedding, WordNet synonyms, HowNet sememe candidates, Masked Language Models (MLM), and prompt engineering. In the second step, various optimization methods, e.g., word-saliency based static optimization (Ren et al., 2019; Garg and Ramakrishnan, 2020; Li et al., 2020), and objective guided dynamic heuristic optimization (Alzantot et al., 2018; Zang et al., 2020), have been widely explored.

Nevertheless, the adversarial attack on Chinese text, which ranks the second popular language with 1 billion speakers (Comrie and Comrie, 2018), has

*Corresponding author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 4553–4565
November 12-16, 2024 ©2024 Association for Computational Linguistics

received limited attention. Existing Chinese text attack methods can be roughly categorized into two groups, i.e., the direct transfer from the English attack and Chinese characteristic based static optimization. The former group ignores the rich Chinese characteristics, e.g., the character structure, sound, and shape, which greatly limits the search space and usually results in unnatural adversarial sentences (as shown in Figure 1). The latter group generates substitutions based on Chinese linguistic characteristics, for example, splitting characters into radicals (Nuo et al., 2020), converting simplified and traditional Chinese characters (Tong et al., 2020), and replacing characters with ones of similar glyph or pinyin (Zhang et al., 2020b; Liu et al., 2023a). However, previous studies take only one of the above strategies at each attack period, which is largely insufficient to integrate multiple Chinese linguistic information. Additionally, the latter group usually employs word saliency based static optimization (Zhang et al., 2020b; Liu et al., 2023a), while the pre-defined attack sequence can not match the dynamic word saliency change with the attack iterations going on.

To address these problems, we propose a novel Chinese text adversarial attack method, named Immune-based Sound-Shape Code (ISSC). Specifically, our ISSC simultaneously investigates multiple Chinese characteristics by analyzing the Sound-Shape Code (SSC), which contains both pronunciation information and visual information. These Chinese characteristics enrich the search space and improve the naturality of adversarial text from the view of native Chinese speakers. Then we devise an Immune Algorithm (IA) to optimize the attack priority. Particularly, we enhance the diversity of feasible solutions by minimizing the reputation rate in the objective function and the half-population vaccination operation. This is effective in improving the search space and providing more opportunities to approach the global optimal. Extensive experiments demonstrate that our ISSC achieves the highest attack success rate, and lower modification rate with more natural adversarial examples in most cases. We emphasize our contributions as below.

- We generate natural Chinese substitutions by combining the Sound-Shape Code (SSC), which carries both pronunciation and visual information. This is significant to ensure the adversarial output can be read smoothly.

- We design an adaptive immune algorithm (IA) method to determine the word replacement order. We carefully devise the objective function and the half-population vaccination operation to reduce the population repetition rate. These two strategies extend the search space of classical IA and provide a higher probability of finding the global optimal.

- We conduct extensive experiments on five public datasets to validate the effectiveness of our method. The results manifest that our ISSC outperforms the baselines in terms of ASR and text quality, and also shows superiorities in transferability and adversarial training.

2 Related Works

In this section, we briefly review the text adversarial attacks in both English and Chinese.

English Text Attack. According to the modification granularity, English attack methods can be categorized into character-level, word-level, sentence-level, and multi-level attacks. Generally, the character modifications easily lead to misspelled words, which is also the case in Chinese (He et al., 2023). Besides, sentence insertion or paraphrasing usually produces unreadable sentences with a relatively low attack success rate. Owing to the flexibility and invisibility, the word-level and multi-level attacks highly rely on the word substitutions, so we focus more on the word substitution based attacks. In this track, two common steps are (1) collecting appropriate substitution candidates and (2) optimizing the replacement order.

In the first step, the substitution candidates are typically collected from GloVe embedding space (Alzantot et al., 2018), WordNet synonyms (Ren et al., 2019), HowNet sememe candidates (Zang et al., 2020), Masked Language Model (MLM) (Garg and Ramakrishnan, 2020; Li et al., 2020), and prompt engineering (Xu et al., 2024). These methods can also be utilized at the same time, as they may produce complementary candidates. In the second step, researchers tend to determine the word replacement order via either word saliency based static method (Ren et al., 2019; Li et al., 2020) or objective guided heuristic search (Alzantot et al., 2018; Zang et al., 2020). The static methods compute the word importance score at once, with relatively higher efficiency than dynamic optimization, e.g., Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). While the adaptive

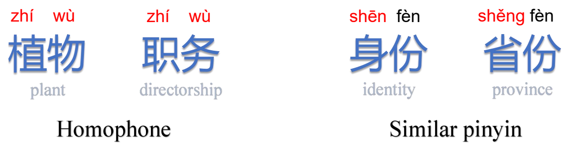

Figure 2: Illustration of same or similar pronunciations in Chinese words.

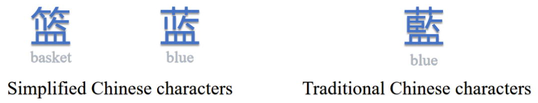

Figure 3: Illustration of characters in similar glyph.

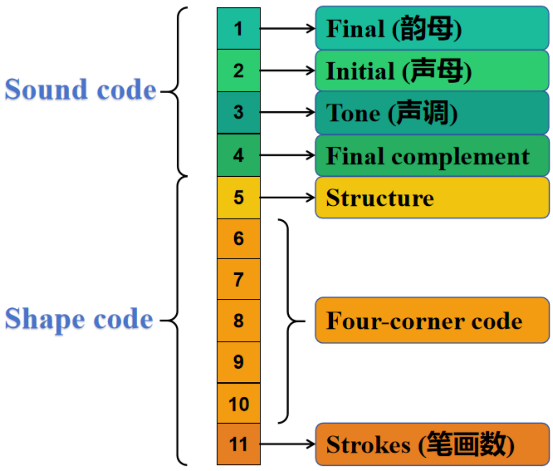

Figure 4: Sound-Shape Code and its components.

search can better fit the dynamic change of word importance, and usually attains higher attack success rate with lower perturbation cost.

Chinese Text Attack. The researches on Chinese text attacks have received much less attention than English text attacks. We would like to divide the existing Chinese text attack methods into English-transferred attacks and Chinese characteristic based attacks. Intuitively, some English text attack methods are naturally suitable for Chinese text attacks, such as PSO(Zang et al., 2020), GA(Alzantot et al., 2018), and BEAT(Li et al., 2020). The reason is that their candidates selection methods, i.e., HowNet, GloVe, and MLM, also provide the Chinese option. However, this group tends to ignore the rich linguistic characteristics of Chinese, so the candidates size and quality are both greatly limited.

The Chinese characteristic based attacks generate substitutions according to some unique Chinese features. Some researches focus on replacing characters with homophones (Wang et al., 2019; Zhang et al., 2020b; Wang et al., 2022; Liu et al., 2023a) and ones of similar glyph (Zhang et al., 2020b; Wang et al., 2022; Liu et al., 2023a). Some researches generate substitutions based on splitting characters of specific structures into radicals (Zhang et al., 2020b; Nuo et al., 2020) and the conversion of simplified and traditional fonts (Tong et al., 2020). However, existing researches frequently adopt only one of the above methods in each attack period, which is insufficient to explore the rich Chinese text features. Besides, existing Chinese characteristic based attacks follow the static attack sequence, which usually leads to word over-substitution and low attack success rate.

3 Methodology

In this section, we first define the text adversarial attack problem in § 3.1, then we discuss the Sound-Shape Code (SSC) and Immuse Algorithm (IA) in § 3.2 and § 3.3, respectively.

3.1 Problem Definition

Given a DNNs classifier $F : \mathcal{X} \to \mathcal{Y}$, which maps the input space $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots \mathbf{X}_N\}$ to a set of labels $\mathcal{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_M\}$ via Eq. (1)

$$\arg\max_{\mathbf{Y}_i \in \mathcal{Y}} P(\mathbf{Y}_i|\mathbf{X}) = \mathbf{Y}_{true}, \tag{1}$$

where $\mathbf{X}$ and $\mathbf{Y}_{true}$ denote the input sentence and its true label. An adversarial example $\mathbf{X}'$ misleads the classifier F by introducing perturbation $\Delta\mathbf{X}$ to the original input, i.e., $\mathbf{X}' = \mathbf{X} + \Delta\mathbf{X}$, as Eq. (2)

$$\arg\max_{\mathbf{Y}_i \in \mathcal{Y}} P(\mathbf{Y}_i|\mathbf{X}') = \mathbf{Y}', \mathbf{Y}' \neq \mathbf{Y}_{true}. \tag{2}$$

A rational attacker should craft human imperceptible perturbation $\Delta\mathbf{X}$ with natural substitutions and fewer modifications. In this work, we generate natural substitutions by exploring the SSC and reduce the modifications by the adaptive IA.

3.2 Sound-Shape Code based Substitutions

Chinese characters exhibit a diverse array of distinctive characteristics, which are conducive to the generation of natural substitutions, in contrast to English. Existing works frequently generate Chinese substitutions based on pinyin and glyph similarity. Chinese Pinyin is the widely adopted romanization system for Standard Chinese. As shown in Figure 2, the romanized spelling above each Chinese

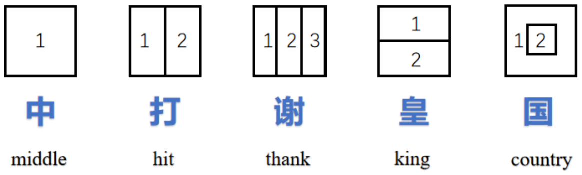

Figure 5: Illustration of Chinese characters with different structures. Some structures comprise subareas, depicted by rectangles.

character represents its pronunciation, including the initial, final, and tone. Besides, as a kind of hieroglyphics, Chinese character comprises several radicals. Figure 3 shows that some characters can be transformed into ones with similar glyphs. Nevertheless, existing works fail to encompass multiple Chinese characteristics, leading to unnatural substitutions, e.g., "骛"wù (swiftly) → ["凳"dèng (stool) or "物"wù (object)] generated via Argot either focus on visual perception or pronunciation rather than ours ["鹜"wù (gallop)].

Sound-Shape Code (SSC) is a Chinese character encoding method and has been proven to solve the task of Chinese text error correction effectively (Wang et al., 2020). It consolidates multiple Chinese characteristics into a unified dimension, offering a holistic measure of character features. Thus, replacing characters with ones of similar SSC will be imperceptible for a native Chinese speaker. Figure 4 shows that the SSC consists of 11 bits and is divided into two parts: sound code and shape code. The sound code comprises initials, finals, and tones, encoding the pronunciation information of Chinese characters. The shape code reflects the visual perception information of Chinese characters, including stroke count, structure, and four-corner code. The layout of a character largely affects how people perceive visual similarity between characters (Liu et al., 2010). The character structure (Figure 5) and four-corner code (Figure 6) reflect the local and global layout of Chinese characters. They are helpful in excluding visually dissimilar characters. The Stroke is the smallest unit of Chinese characters and similar stroke counts tend to produce visually similar characters.

Suppose an input sentence $\mathbf{X} = [c_1, c_2, \cdots, c_m]$ with m characters, we first segment it into a series of words (phrases) $\mathbf{X} = [w_1, w_2, \cdots, w_n]$ with Jieba[1]. Then we encode each word w_i into Sound-Shape Code[2] and get the top-k words with high

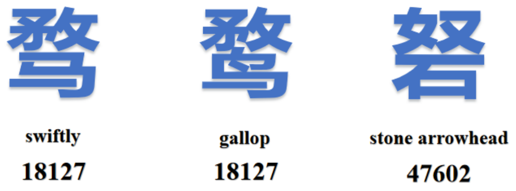

Figure 6: Illustration of Chinese characters with four-corner code. The four-corner code utilizes four digits 0 to 9 to represent different groups of strokes for the four corners of a Chinese character, with a complement code added at the end.

similarity as its substitutions $\mathbb{V}_{ssc}(w_i)$. For example, we encode "骛 (swiftly)" and "鹜 (gallop)" into codes "$\underline{\text{5J04218127}}$C" and "$\underline{\text{5J04218127}}$E", then, the longest common sub-strings length of sound code and shape code are calculated respectively to measure the similarity.

3.3 Adaptive Immune Optimization

Before detailing our approach, we first clarify the concepts of original immune algorithms (IA). Inspired by the biological immune, the IA is a heuristic optimization algorithm that simulates the process of organism producing antibodies when antigen invades, where the antigen and the antibody denote an objective function and a feasible solution, respectively. Specifically, the algorithm evaluates the quality of the solution by calculating the affinity of the antibody to the antigen. It then performs clone selection, where the antibodies with superior quality are chosen, followed by mutation operation to explore potentially feasible solutions. Those antibodies with lower quality are refreshed to random initialization. It is worth mentioning that the IA avoids the concentrations of antibodies in the clone selection step to enrich the diversity of feasible solutions. Therefore, the IA is capable of reducing population redundancy and enhancing the overall quality of the population.

Formally, given a segmented input sentence $\mathbf{X}_{ori} = [w_1, w_2, \cdots, w_n]$ with n words and its true label $\mathbf{Y}_{true}$, our ISSC method targets to craft an adversarial example $\mathbf{X}_{adv}$ to mislead the DNNs classifier F. The optimization procedure is given in Algorithm 1. Firstly, we **initialize** the population by repeating the mutation step for N times to get N antibodies $\mathcal{X}^1 = \{\mathbf{X}_1^1, \mathbf{X}_2^1, \cdots, \mathbf{X}_n^1\}$. Following the GA and PSO, we adopt a single point **mutation** strategy that an original word w_i is selected randomly and replaced at a time. The quality of each antibody $\mathbf{X}_i^t$ in the population

[1] https://github.com/fxsjy/jieba

[2] https://github.com/qingyujean/ssc

$\mathcal{X}^t = \{\mathbf{X}_1^t, \mathbf{X}_2^t, \cdots, \mathbf{X}_N^t\}$ can be evaluated via the objective function with a hyper-parameter λ

$$\mathcal{S}(\mathbf{X}_i^t, \mathcal{X}^t) = \mathcal{J}(\mathbf{X}_i^t) - \lambda \times \rho(\mathbf{X}_i^t, \mathcal{X}^t), \tag{3}$$

where the affinity $\mathcal{J}(\mathbf{X}_i^t)$ is defined as

$$\mathcal{J}(\mathbf{X}_i^t) = 1 - P(\mathbf{Y}_{true}|\mathbf{X}_i^t). \tag{4}$$

Intuitively, higher affinity $\mathcal{J}(\mathbf{X}_i^t)$ is better for attacker. The second term, i.e., the concentration of each antibody ρ is defined as

$$\rho(\mathbf{X}_i^t, \mathcal{X}^t) = \frac{1}{N} \sum_{j=1}^{N} Sim(\mathbf{X}_i^t, \mathbf{X}_j^t). \tag{5}$$

This reflects the similarity of antibodies in the same population. If the antibodies are similar to each other, it is easy to converge to a local optimal. Therefore, we reduce ρ to improve the diversity of the feasible solutions, which provides higher probability to find the global optimal. Particularly, the similarity of antibodies is defined via the edit-distance with a threshold δ_s

$$Sim(\mathbf{X}_i^t, \mathbf{X}_j^t) = \begin{cases} 1, & len(\mathbf{X}_i^t \neq \mathbf{X}_j^t) \leq \delta_s, \\ 0, & len(\mathbf{X}_i^t \neq \mathbf{X}_j^t) > \delta_s, \end{cases} \tag{6}$$

where $len(\mathbf{X}_i^t \neq \mathbf{X}_j^t)$ counts the number of different words (phrases) between $\mathbf{X}_i^t$ and $\mathbf{X}_j^t$. Based on the objective function, we calculate the fitness value of each antibodies and sort them in descending order for the next step.

In the **clone selection** step, we divide the population into two groups, i.e., the top $N_c = N \times p_c$ antibodies with high fitness value and the rest antibodies with low quality. For each antibody in the former group, we randomly adopt the mutation operation according to the mutation probability p_m, otherwise perform the clone operation, for m times in line 5. Then, we store the one with the highest fitness from the m clones $\{\mathbf{X}_{c,j}^t\}_{j=1}^m$ of i-th antibody $\mathbf{X}_i^t$. This allows for high-quality antibodies to conduct localized exploration and preserve beneficial information at the individual level.

The classical immune algorithm usually initializes the latter group of antibodies to eliminate low-quality feasible solutions, named population refreshing. However, for adversarial text generation, the adversarial sample is initialized, which means that only a few words are modified. In the later iterations, a small number of modifications is often not enough to generate successful adversarial samples, so that the random refreshing will fail. Inspired by (Yuan et al., 2011), instead of random refreshing, we adopt a vaccine extraction and injection mechanism, simplified as **vaccination**, to solve this problem. Specifically, we store the optimal individual during the iteration as the vaccine and inject the vaccine into the low-quality antibodies for better refreshing. In lines 7-8, each word of antibody with low fitness determines whether to move to the corresponding word of the vaccine with a vaccination probability p_v. As the iteration progresses, the vaccine tends to have more modifications, increasing the likelihood of generating successful adversarial samples. Besides, this can retain the global advantage information at the population level and enhance the cooperation among individuals in the same population.

```
Algorithm 1: Our ISSC algorithm
   Input: Input sentence $\mathbf{X}_{ori} = \{w_1, w_2, \cdots, w_n\}$
          and true label $\mathbf{Y}_{true}$, DNNs classifier $F$
   Output: Adversarial example $\mathbf{X}_{adv}$
1  Parameters: population size $N = 40$, maximum
     number of iteration times $T = 30$, proportion of
     clone selection $p_c = 0.5$, probability of mutation
     $p_m = 0.7$, number of antibody clones $m = 5$,
     probability of vaccination $p_v = 0.5$, weight
     parameter $\lambda = 0.5$, threshold $\delta_s = 3$, optimal
     antibody $\mathbf{X}^* = \mathbf{X}_{ori}$;
   /* Initialize the population                 */
2  $\mathcal{X}^1 = \{\mathbf{X}_i^1\}_{i=1}^N \leftarrow Mutation(\mathbf{X}_{ori})$ ;
   /* Immune algorithm iteration                */
3  for $t = 1 \rightarrow T$ do
       /* clone selection                       */
4      for $i = 1 \rightarrow N_c$ do
5          $\{\mathbf{X}_{c,j}^t\}_{j=1}^m \leftarrow Mutation(\mathbf{X}_i^t, p_m)$ ;
6          Select the best $\mathbf{X}_i^t$ among $\{\mathbf{X}_{c,j}^t\}_{j=1}^m$ via
             Eq. (3);
       /* vaccine extraction and injection      */
7      for $i = N_c + 1 \rightarrow N$ do
8          $\mathbf{X}_i^t \leftarrow Vaccination(\mathbf{X}_i^t, \mathbf{X}^*)$;
       /* optimal antibody preservation         */
9      Sort $\mathcal{X}^t$ in reverse order via Eq. (3) ;
10     if $\mathcal{S}(\mathbf{X}_1^t) > \mathcal{S}(\mathbf{X}^*)$ then
11         $\mathbf{X}^* = \mathbf{X}_1^t$;
12     if $F(\mathbf{X}^*) \neq \mathbf{Y}_{true}$ then
13         return $\mathbf{X}_{adv} = \mathbf{X}^*$;
14 return $\mathbf{X}_{adv} = \mathbf{X}^*$;
```

Finally, we update the optimal antibody $\mathbf{X}^*$ at the end of each iteration (Algorithm 1 lines 9-11). If the victim model F is misled, we **terminate** the algorithm and return the optimal antibody $\mathbf{X}^*$ as the adversarial example in line 13. Otherwise, the algorithm returns to the **clone selection** and continues to iterate.

Dataset	#Class	Avg. #C	Train	Test	Dev
Chinanews	7	116	1,400,000	112,000	-
ChnSentiCorp	2	108	9600	1200	1200
OCNLI	3	24	50000	3000	2950
Ctrip	2	139	12000	3000	-
JD	2	43	4800	3000	-

Table 1: Details of datasets. "# Class" denotes the number of classifications. "Avg. # C" denotes the average sentence length (number of characters). "Train", "Dev", and "Test" denote the number of samples in the training, validation and test sets, respectively.

4 Experiments

4.1 Datasets and Victim Models

We conduct our experiments on five publicly available datasets, including Chinanews (Zhang and LeCun, 2017), Chinese Sentiment Corpus (ChnSentiCorp) (Tan and Zhang, 2008), Original Chinese Natural Language Inference (OCNLI) (Hu et al., 2020), Ctrip Hotel Reviews (Ctrip)[3], and JD.com Reviews (JD)[4]. The Chinanews is a multi-class news classification dataset. The ChnSentiCorp, Ctrip, and JD are all binary sentiment classification datasets. The OCNLI is used for natural language inference (NLI) task. Statistical details of these datasets are summarized in Table 1.

We assess the effectiveness of our method by attacking six popular victim models, including CNN, LSTM, BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), and DistilBERT (Sanh et al., 2019). For CNN and LSTM, we use a 300-dimensional embedding layer. We download the BERT (google-bert/bert-base-chinese), RoBERTa (uer/roberta-base-wwm-chinese-cluecorpussmall), ALBERT (uer/albert-base-chinese-cluecorpussmall) and DistilBERT (distilbert/distilbert-base-multilingual-cased) from the Huggingface[5] and fine-tune them with a common default configuration. You can download these models easily here [6].

4.2 Baseline Methods

We compare our ISSC with both typical English transfer methods, such as BERT-Attack (**BEAT**) (Li et al., 2020), **GA** (Alzantot et al., 2018) and **PSO** (Zang et al., 2020), and Chinese attack algorithms, such as **Argot** (Zhang et al., 2020b) and Expanding Scope (**ES**) (Liu et al., 2023a). As mentioned in § 2, existing works in Chinese scene frequently adopt a group of methods to generate substitutions, which is insufficient to obtain natural candidate words. Besides the existing candidates selection methods, e.g., GLoVe and masked language model, our ISSC can further takes the Chinese characteristic into consideration, so the quality of adversarial text can be improved. On the other hand, compared to the heuristic optimizations, e.g., GA and PSO, the proposed Immune Algorithm (IA) is good at reducing the population duplication, which is significant to avoid local optimal.

4.3 Evluation Metrics

We evaluate the performance of each attack algorithm with the following metrics.

Attack Success Rate. The attack success rate (ASR) is defined as the percentage of the number of samples that successfully mislead the victim model over the total number of attacked samples.

Modification Rate. The modification rate (MR) is defined as the percentage of the number of modified Chinese characters to the length of the sentence. Different from the English scene, too long or too short phrases lead to additional modification rates when synonym substitution is performed.

Semantic similarity and fluency. The semantic similarity and the fluency reflect the quality of adversarial examples. We adopt the BERTScore (Zhang et al., 2020a) and perplexity to evaluate the semantic similarity and fluency, respectively.

4.4 Experimental Setup

The parameter settings for our method are given in line 1 of Algorithm 1. For the victim model, we finetune models on five Chinese datasets with a default training configuration. For the baselines in English scene, we transfer them to the Chinese scene with their author recommended parameter values. For the baselines in Chinese scene, we replicate the Argot following the original paper, and experiment with ES implemented on the TextAttack framework (Morris et al., 2020). For all tasks, we apply the stopwords modification and repeat modification constraints. Additionally, for the NLI task, only the premise text is allowed to be modified. To achieve efficiency, we randomly select 500 instances from the test sets to craft adversarial examples. All experiments are implemented

[3] https://github.com/cgq666/Chinese-text-sentiment-classification-dataset/tree/master/Ctrip

[4] https://github.com/cgq666/Chinese-text-sentiment-classification-dataset/tree/master/JD.com

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

[6] https://huggingface.co/WangA

Dataset	Model	ACC / %	Attack Success Rate (ASR) / %						Modification Rate (MR) / %					
			Argot	GA	PSO	BEAT	ES	ISSC	Argot	GA	PSO	BEAT	ES	ISSC
Chinanews	CNN	88.20	99.32	71.40	84.68	91.89	**100**	**100**	11.38	22.87	25.51	19.36	14.66	**8.54**
	LSTM	88.21	74.04	69.98	79.86	59.28	84.16	**93.00**	9.45	13.82	18.26	12.82	13.51	**7.56**
	BERT	90.48	89.39	59.52	85.06	68.83	90.69	**93.51**	18.73	22.14	26.52	23.13	18.83	**13.85**
	RoBERTa	92.11	98.74	55.79	90.11	74.32	98.74	**98.95**	16.75	22.59	28.25	24.82	18.82	**12.44**
	ALBERT	89.19	97.37	69.80	90.81	76.15	98.25	**98.91**	13.43	20.93	25.45	21.11	15.25	**10.79**
	DistilBERT	89.95	**99.13**	71.68	89.54	78.00	**99.13**	**99.13**	14.16	21.61	27.17	20.85	15.68	**11.32**
ChnSentiCorp	CNN	89.83	**99.77**	85.23	97.03	84.25	98.86	**99.77**	8.61	18.29	23.19	15.49	11.44	**6.74**
	LSTM	88.50	97.67	93.49	97.68	69.37	98.61	**99.30**	7.70	10.99	16.25	12.56	10.45	**5.54**
	BERT	95.50	90.41	63.97	94.24	56.93	92.54	**96.16**	19.4	26.39	30.69	21.99	22.13	**13.36**
	RoBERTa	95.80	91.23	56.37	91.65	62.84	94.78	**96.66**	20.92	25.17	34.79	22.63	24.37	**14.86**
	ALBERT	94.50	95.74	68.44	95.10	65.03	97.87	**98.51**	16.53	22.39	27.29	19.03	18.10	**11.98**
	DistilBERT	94.67	95.76	73.94	94.92	68.22	98.09	**98.73**	13.77	20.52	25.55	18.78	15.85	**10.20**
OCNLI	BERT	73.42	96.00	86.29	90.29	80.00	97.43	**98.00**	**7.02**	10.62	11.65	10.17	9.06	8.14
	RoBERTa	75.59	95.70	85.22	90.59	81.72	97.04	**97.85**	**6.94**	11.68	12.23	10.69	9.45	8.18
	ALBERT	70.92	97.46	85.63	89.86	81.41	97.18	**97.75**	**6.98**	10.99	12.72	10.09	9.19	7.60
	DistilBERT	68.31	95.48	86.45	88.25	79.22	95.78	**96.39**	**7.04**	11.76	12.46	10.18	8.70	7.76
Ctrip	BERT	96.57	96.83	79.28	91.33	90.27	**97.04**	**97.04**	9.62	16.56	19.63	13.40	9.93	**7.04**
	RoBERTa	97.13	95.35	83.09	88.58	89.43	95.56	**96.62**	9.89	17.92	20.38	13.93	10.59	**7.19**
	ALBERT	96.67	97.69	82.56	96.01	88.66	98.11	**98.32**	8.05	15.84	19.21	12.31	9.12	**6.37**
	DistilBERT	95.43	97.06	87.53	92.90	92.26	97.20	**97.42**	7.92	14.51	17.69	11.20	8.66	**6.30**
JD	BERT	95.37	96.22	68.70	87.39	30.88	92.65	**97.06**	27.06	42.65	52.16	34.21	37.42	**26.70**
	RoBERTa	95.57	88.12	59.38	79.38	24.58	82.29	**95.00**	38.49	43.22	59.97	39.12	49.59	**32.81**
	ALBERT	95.03	91.81	71.22	82.35	30.88	85.92	**94.12**	32.63	36.49	49.99	32.30	39.16	**27.03**
	DistilBERT	93.70	95.93	76.45	88.87	33.19	92.29	**96.36**	24.64	42.94	51.13	30.78	30.35	**22.74**
Average		——	94.68	74.64	89.85	69.07	95.01	**97.27**	14.88	21.79	27.01	19.21	17.93	**12.29**

Table 2: The attack success rate and modification rate of all attack methods on five datasets. We highlight the best results **in bold**. The "ACC" represents the original accuracy of models. We achieve the best results in most cases, with attack success rate and modification rate outperforming the best methods by 2.26% and 2.59% on average.

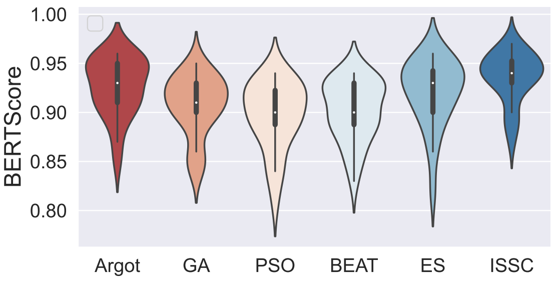

Figure 7: Results of BERTScore on all datasets.

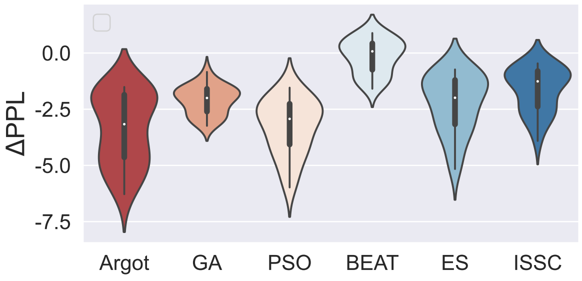

Figure 8: Results of perplexity on all datasets.

on the TextAttack framework.

4.5 Experimental Results

The experimental results of attack success rate and modification rate are shown in Table 2. For the ASR, our algorithm achieves the best results in most cases and outperforms the best method by 2.26% on average. Despite extensive modifications, some methods are still unable to outperform our method. For the MR, our method can produce fewer modifications in most cases, with an average reduction of 2.59 % compared to the best method. Our ISSC achieves the second-best results on OCNLI. We attribute this to the smaller input length.

Semantic similarity and fluency. To avoid human perception, the adversarial examples should be semantically consistent with the original text. We adopt BERTScore to evaluate the semantic consistency before and after attacks. As shown in Figure 7, our ISSC achieves the highest semantic consistency on the whole and average. The perplexity reflects the fluency of sentences by calculating the prediction probability via a language model, i.e., GPT-2. We calculate the change of PPL before and after attacks, the smaller the decrease (the larger the Δ PPL), the more fluent the sentence. Figure 8 shows that ISSC generates more fluent adversarial examples than other methods except for BEAT. The

Att.\Models	CNN	LSTM	BERT	RoBERTa	ALBERT
None	89.83	88.50	95.50	95.80	94.50
Argot	75.88	74.12	86.28	92.92	87.17
GA	77.65	73.64	85.67	89.40	82.23
PSO	75.89	72.54	84.15	90.18	85.94
ES	73.22	72.57	83.37	86.83	82.94
ISSC	68.67	72.53	80.04	85.19	77.04

Table 3: Transfer attack on ChnSentiCorp dataset. "Att." denotes different attack methods. "None" denotes the classification accuracy of model that is not under attack. Lower accuracy means better transferability.

Adv.T \ Att.	Argot	GA	BEAT	ISSC
None	96.22	68.70	30.88	97.06
Argot	-51.01	-38.08	-8.80	-34.56
GA	-35.21	-44.80	-7.61	-28.51
PSO	-41.64	-20.99	-4.84	-29.77
BEAT	-3.67	-5.51	2.31	-2.80
ES	-28.99	-21.13	-4.24	-15.66
ISSC	-44.44	-40.40	-9.29	-35.63

Table 4: Adversarial training results on JD dataset with BERT. "Att." and "Adv.T" denote the attack methods and adversarial training methods respectively. "None" denotes the ASR of different attack methods on original BERT. The rest denotes the decrease of ASR before and after the adversarial training.

reason is that the BEAT employs BERT masked language model to generate candidates, which mainly aims to minimize the perplexity.

4.6 Transferability

The transferability of adversarial examples refers to whether the adversarial examples designed on a model F_1 can mislead another model F_2 without any access to it. We select four baselines with higher ASR and conduct the experiments on ChnSentiCorp dataset. Specifically, we obtain the adversarial examples crafted on DistilBERT by various methods and then perform the transfer attack on five unknown models (CNN, LSTM, BERT, RoBERTa, and ALBERT). The results of classification accuracy on adversarial data are shown in Table 3. The experimental results illustrate that ISSC generates adversarial examples with higher transferability than others.

4.7 Adversarial Training

Adversarial training is a common strategy to improve the adversarial robustness of the model, which is achieved by adding adversarial examples to the original training set. We first train the BERT model on clean JD dataset. Then we randomly generate 500 adversarial examples (10.4 % of the original training set size) and add them to the training set. Finally, we retrain the model and evaluate its robustness by calculating the ASR of different attack algorithms. As shown in Table 4, our method can reduce the ASR of unknown methods more in most cases, and is competitive with models retrained by consistent attacks.

Dataset	Method					
	SSC+delete		SSC+PSO		SSC+IA	
	ASR /%	MR/%	ASR/%	MR/%	ASR/%	MR/%
Chinanews	87.88	22.28	87.01	18.10	93.51	13.85
ChnSentiCorp	88.27	23.23	89.55	17.30	96.16	13.36

Table 5: Decomposition results of various searches on BERT with two datasets. The "SSC+delete" and "SSC+PSO" represent attacks with different search methods. The "SSC+IA" denotes our proposed ISSC.

4.8 Ablation

In this subsection, we conduct ablation experiments on various models and datasets to prove the effectiveness of proposed components of our method.

4.8.1 Decomposition Analyses

To demonstrate the advantages of our proposed adaptive immune algorithm (IA) over other search methods, we conduct decomposition experiments on BERT with Chinanews and ChnSentiCorp datasets. The straightforward candidate searches can be roughly divided into word saliency-based static methods, e.g., removing the input word one by one and calculating importance scores, and objective-guided heuristic optimization methods, e.g., PSO. Specifically, we replace IA with other search methods, e.g., deletion-based static search and the dynamic PSO, and keep the candidate method unchanged, i.e., SSC. Our IA achieves better performance in both ASR (5.63%$\uparrow$) and MR (3.94%$\downarrow$) as shown in Table 5. At the same time, it also proves that dynamic methods are often better than static searches.

4.8.2 Ablation of Vaccination

To validate our proposed **vaccination** mechanism, we conduct ablation experiments on three models with the Chinanews datasets. Specifically, we replace the vaccination module with an initialization module, which means the low quality of antibodies will be randomly initialized with a single word changed at the end of each iteration. Table 6 shows

Model	Method	ASR /%	MR /%	Δ PPL	Query
CNN	$ISSC$	100	8.54	-2.13	4325
	$ISSC_{-vacc}$	98.87	8.63	-2.18	5781
LSTM	$ISSC$	93.00	7.56	-1.62	5836
	$ISSC_{-vacc}$	92.78	7.97	-1.77	8136
BERT	$ISSC$	93.51	13.85	-2.61	7334
	$ISSC_{-vacc}$	92.86	13.84	-2.66	9621

Table 6: Ablation results of vaccination on three models with the Chinanews dataset. The "$ISSC_{-vacc}$" denotes the algorithm for removing the vaccination module.

Target Label	Method	ASR /%	MR /%	Δ PPL	BERTScore
0	Argot	63.17	29.04	-8.14	0.86
	ES	76.92	34.66	-5.68	0.85
	ISSC	80.89	23.26	-3.57	0.88
1	Argot	74.76	29.57	-8.89	0.85
	ES	74.05	34.29	-6.96	0.84
	ISSC	78.81	24.14	-5.14	0.87

Table 7: Experimental results of target attacks on a multi-class classification dataset with different methods. The 0 and 1 represent the target labels that need to flip to by attack methods.

a reduction in almost all metrics after removing the vaccination module, which indicates that the vaccination is conducive to improving the global search ability. In particular, results show that the proposed vaccination module can largely reduce the number of queries of our method, which means that it has more advantages in the face of some access restrictions. This is because the module can help the population better initialize and help the algorithm converge in advance rather than classical random initialization.

4.9 Target Attacks

Our approach is primarily designed for untargeted attacks, which means that we only need to flip the true label to any wrong label. Meanwhile, our approach can be easily modified for targeted attacks by changing the affinity in Eq (4). We conduct a targeted attack experiment on BERT with the Chinanews dataset. Table 7 shows that both the attack success rate and adversarial sample quality decrease significantly, indicating that targeted attacks are more difficult than untargeted attacks for all methods. Besides, our ISSC can generate more targeted adversarial examples with a lower modification than other attacks.

5 Conclusion

In this paper, we proposed a novel adversarial Chinese text attack algorithm named Immune-based Sound-Shape Code (ISSC). The ISSC adopts the Sound-Shape Code methods to generate natural Chinese substitutions and optimizes the attack priority via adaptive Immune Algorithm (IA). We conducted extensive experiments to demonstrate the effectiveness of our algorithm in terms of attack success rate, text quality, transferability, adversarial training, and targeted attacks. In addition, we conducted ablation experiments to validate the enhancement of global search capability to IA by the vaccination module. In the future, we hope our approach will draw attention to multilingual adversarial samples.

Limitations

Dynamic Parameter. The hyper-parameter setting of our search algorithm is simple but effective. More parameters can be designed to change with the number of iterations to further improve the attack performance.

Language Transfer Restriction. We propose a substitution method to integrate multiple Chinese features, some of which are limited to some East Asian scripts and not applicable to Latin scripts.

References

Ahmed Abbasi, Abdul Rehman Javed, Farkhund Iqbal, Natalia Kryvinska, and Zunera Jalil. 2022. Deep learning for religious and continent-based toxic content detection and classification. Scientific Reports, 12.

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.

Dmitriy Bespalov, Sourav Bhabesh, Yi Xiang, Liutong Zhou, and Yanjun Qi. 2023. Towards building a robust toxicity predictor. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), pages 581–598, Toronto, Canada. Association for Computational Linguistics.

Yangyi Chen, Jin Su, and Wei Wei. 2021. Multi-granularity textual adversarial attack with behavior cloning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4511–4526, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bernard Comrie and Bernard Comrie. 2018. The World's Major Languages. Routledge, London.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. 2024. Masterkey: Automated jailbreaking of large language model chatbots. In Proc. ISOC NDSS.

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.

Sahar A. El Rahman, Feddah Alhumaidi AlOtaibi, and Wejdan Abdullah AlShehri. 2019. Sentiment analysis of twitter data. In 2019 International Conference on Computer and Information Sciences (ICCIS), pages 1–4.

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.

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.

Zheyu He, Yujin Zhu, Linlin Wang, and Liang Xu. 2023. UMRSpell: Unifying the detection and correction parts of pre-trained models towards Chinese missing, redundant, and spelling correction. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10238–10250, Toronto, Canada. Association for Computational Linguistics.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence Moss. 2020. OCNLI: Original Chinese Natural Language Inference. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3512–3526, 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.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In International Conference on Learning Representations.

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.

Chao-Lin Liu, Min-Hua Lai, Yi-Hsuan Chuang, and Chia-Ying Lee. 2010. Visually and phonologically similar characters in incorrect simplified Chinese words. In Coling 2010: Posters, pages 739–747, Beijing, China. Coling 2010 Organizing Committee.

Hanyu Liu, Chengyuan Cai, and Yanjun Qi. 2023a. Expanding scope: Adapting English adversarial attacks to Chinese. In Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023), pages 276–286, Toronto, Canada. Association for Computational Linguistics.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2023b. Prompt injection attack against llm-integrated applications. Preprint, arXiv:2306.05499.

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

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 119–126, Online. Association for Computational Linguistics.

Cheng Nuo, Guo-Qin Chang, Haichang Gao, Ge Pei, and Yang Zhang. 2020. Wordchange: Adversarial examples generation approach for chinese text classification. IEEE Access, 8:79561–79572.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In MILCOM 2016 - 2016 IEEE Military Communications Conference, pages 49–54.

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

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.

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

Songbo Tan and Jin Zhang. 2008. An empirical study of sentiment analysis for chinese documents. Expert Systems with Applications, 34(4):2622–2629.

Xin Tong, Luona Wang, Runzheng Wang, and Jingya Wang. 2020. A generation method of word-level adversarial samples for chinese text classification. Netinfo Security, 20(9):12–16.

Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. 2022. SemAttack: Natural textual attacks via different semantic spaces. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 176–205, Seattle, United States. Association for Computational Linguistics.

Boxin Wang, Chejian Xu, Shuohang Wang, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, volume 1.

Hanru Wang, Yangsen Zhang, Lipeng Yang, and Congcong Wang. 2020. Chinese text error correction suggestion generation based on soundshape code. In Chinese Lexical Semantics. CLSW 2019, pages 423–432. Springer.

Wenqi Wang, Run Wang, Lina Wang, and Benxiao Tang. 2019. Adversarial examples generation approach for tendency classification on chinese texts. Journal of software, 30(8):2415–2427.

Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan Kankanhalli. 2024. An llm can fool itself: A prompt-based adversarial attack. In International Conference on Learning Representations.

Zhen Yu, Xiaosen Wang, Wanxiang Che, and Kun He. 2022. TextHacker: Learning based hybrid local search algorithm for text hard-label adversarial attack. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 622–637, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Guili Yuan, Yanguang Xue, and Qingjiao Liang. 2011. The design of adaptive immune vaccine algorithm. Advanced Materials Research, 308-310:1094 – 1098.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6066–6080, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with bert. In International Conference on Learning Representations.

Xiang Zhang and Yann LeCun. 2017. Which encoding is the best for text classification in chinese, english, japanese and korean? arXiv preprint arXiv:1708.02657.

Zihan Zhang, Mingxuan Liu, Chao Zhang, Yiming Zhang, Zhou Li, Qi Li, Haixin Duan, and Donghong Sun. 2020b. Argot: Generating adversarial readable chinese texts. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pages 2533–2539.

A Training Details

The CNN consists of an embedding layer and a 1-D convolutional layer containing 150 filters with a filter size of $3 \times 4 \times 5$. The LSTM consists of an embedding layer and a bidirectional LSTM layer with 150 hidden states. For CNN and LSTM, we use a 300-dimension embedding layer to encode the input word (phrase)[7]. Both of them have a dropout rate of 0.3 and are training for 10 epochs.

For the transformer-based models with the base size, we train them for 3 epochs with a 3e-5 learning rate.

B Efficency Analysis

We conduct all experiments on Enterprise Linux Workstation with 12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz, NVIDIA RTX 2080Ti 11G GPU and 40GB RAM. Table 8 lists the time consumption and queries of attacking a sample on average over all datasets. Regarding time consumption, our ISSC achieves superior performance with an acceptable time cost compared with other methods. For the number of model assesses, the heuristic-based methods frequently perform better with more queries than static methods, e.g., Argot, BEAT, and ES. Compared to heuristic search methods, e.g., PSO, our number of queries increases slightly, yet the attack results surpass it significantly.

	Argot	GA	PSO	BEAT	ES	ISSC
Time	6.97	25.26	244.88	2.79	56.59	59.9
Query	221	3371	2931	76	164	3707

Table 8: The time consumption and query numbers of attack methods. "Time" indicates the average attack time for an example (in seconds).

C LLM transfer attacks

Large language models (LLMs) are revolutionizing many fields of human endeavor and continue to develop at a breathtaking pace, in terms of scale and capabilities, but also architectures and applications. LLMs generate text autoregressively, which can solve various downstream tasks in a zero-shot scenario. To valid the transferability of our methods for LLMs, we conduct a simple transfer-based attack against LLMs on GPT-4. Specifically, we randomly select 100 adversarial examples generated on BERT with Chinanews, then manually test on the openai website[8]. As shown in Table 9, LLMs are more robust compared to the smaller models, which is consistent with the results of the AdvGLUE (Wang et al., 2021). Besides, we mainly focus on the classification robustness of LLMs, and researchers are exploring the two prevalent types of adversarial attacks on aligned unimodal Large Language Models (LLMs): jailbreak attacks (Deng et al., 2024) and prompt injection attacks (Liu et al., 2023b). In the future, we will give more emphasis to the adversarial security frontier of LLMs and commit to addressing more complex security evaluation issues of LLMs.

Method	Argot	ES	ISSC
ASR / %	17	16	20

Table 9: Experimental results of transfer-based attacks against LLMs. We manually filter some samples for successful attacks caused by obvious semantic changes.

D Qualitative sample

We list some adversarial examples generated via Chinese attacks (Argot, ES, and ISSC) in Table 10. The results show that our methods can generate natural Chinese substitutions with similar pronunciation and visual perception.

[7] https://github.com/Embedding/Chinese-Word-Vectors

[8] https://chatgpt.com/

Adversarial examples on Chinanews dataset.

Adversarial examples via Argot: 中央~~气象台~~气豚台2008年8月8日18时继续发布强对流天气预报。
Translation: Strong convection weather forecast continued to be issued by "xu" at 1800 on 8 August 2008.
Prediction: Mainland china politics (98%) → Hongkong macau politics (77%)

Adversarial examples via ES: 中央~~气象台~~其象台2008年8月8日18时继续发布强对流天气预报。
Translation: The Central Observatory continued to issue a severe convective
weather forecast at 18:00 on 8 August 2008.
Prediction: Mainland china politics (98%) → Hongkong macau politics (76%)

Adversarial examples via ISSC: 中央~~气象台~~乞象台2008年8月8日18时继续发布强对流天气预报。
Translation: The Central Observatory continued to issue a severe convection
weather forecast at 18:00 on August 8, 2008.
Prediction: Mainland china politics (98%) → Hongkong macau politics (77%)

Adversarial examples on ChnSentiCorp dataset.

Adversarial examples via Argot: 还稍微~~重~~中了点，~~可能~~克能是硬盘大的原故，还要再轻半斤就好了。
Translation: Also a little bit, gram can be the reason for the hard disk, but also light half a kilogram is good.
Prediction: Negative (90%) → Positive (57%)

Adversarial examples via ES: 还稍微~~重~~关心了点，可能是硬盘大的原故，还要再轻半斤就好了。
Translation: Also a little concerned, may be the reason for the hard disk is big, but also light half a kilogram is good.
Prediction: Negative (90%) → Positive (83%)

Adversarial examples via ISSC: 还稍微重了点，可能是硬盘大的~~原故~~愿故，还要再轻半斤就好了。
Translation: Also a little heavy, may be the hard disk big wish, but also lighter half a kilogram.
Prediction: Negative (87%) → Positive (67%)

Table 10: Adversarial examples generated on the two datasets via Chinese attacks. Replacing a ~~word/character~~ with a substitution misleads the correct prediction to a wrong class without fooling human. Ours achieve similar pronunciation and visual perception.