Word-level Textual Adversarial Attacking as Combinatorial Optimization

Yuan Zang[1*], **Fanchao Qi**[1*], **Chenghao Yang**[2*†], **Zhiyuan Liu**[1‡],
Meng Zhang[3], **Qun Liu**[3], **Maosong Sun**[1]
[1]Department of Computer Science and Technology, Tsinghua University
Institute for Artificial Intelligence, Tsinghua University
Beijing National Research Center for Information Science and Technology
[2]Columbia University [3]Huawei Noah's Ark Lab
{zangy17,qfc17}@mails.tsinghua.edu.cn, yangalan1996@gmail.com
{liuzy,sms}@tsinghua.edu.cn, {zhangmeng92,qun.liu}@huawei.com

Abstract

Adversarial attacks are carried out to reveal the vulnerability of deep neural networks. Textual adversarial attacking is challenging because text is discrete and a small perturbation can bring significant change to the original input. Word-level attacking, which can be regarded as a combinatorial optimization problem, is a well-studied class of textual attack methods. However, existing word-level attack models are far from perfect, largely because unsuitable search space reduction methods and inefficient optimization algorithms are employed. In this paper, we propose a novel attack model, which incorporates the sememe-based word substitution method and particle swarm optimization-based search algorithm to solve the two problems separately. We conduct exhaustive experiments to evaluate our attack model by attacking BiLSTM and BERT on three benchmark datasets. Experimental results demonstrate that our model consistently achieves much higher attack success rates and crafts more high-quality adversarial examples as compared to baseline methods. Also, further experiments show our model has higher transferability and can bring more robustness enhancement to victim models by adversarial training. All the code and data of this paper can be obtained on https://github.com/thunlp/SememePSO-Attack.

1 Introduction

Adversarial attacks use *adversarial examples* (Szegedy et al., 2014; Goodfellow et al., 2015), which are maliciously crafted by perturbing the original input, to fool the deep neural networks (DNNs). Extensive studies have demonstrated that DNNs are vulnerable to adversarial attacks, e.g., minor modification to highly poisonous phrases can easily deceive Google's toxic comment detection systems (Hosseini et al., 2017). From another perspective, adversarial attacks are also used to improve robustness and interpretability of DNNs (Wallace et al., 2019). In the field of natural language processing (NLP) which widely employs DNNs, practical systems such as spam filtering (Stringhini et al., 2010) and malware detection (Kolter and Maloof, 2006) have been broadly used, but at the same time the concerns about their security are growing. Therefore, the research on textual adversarial attacks becomes increasingly important.

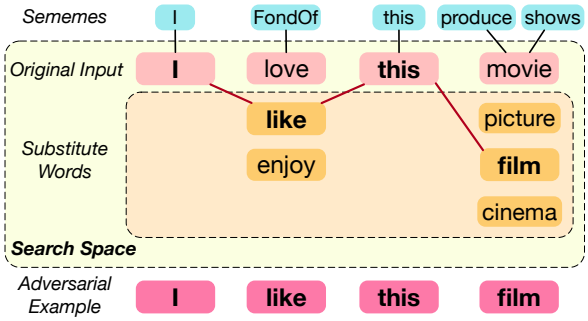

Figure 1: An example showing search space reduction with sememe-based word substitution and adversarial example search in word-level adversarial attacks.

Textual adversarial attacking is challenging. Different from images, a truly imperceptible perturbation on text is almost impossible because of its discrete nature. Even a slightest character-level perturbation can either (1) change the meaning and, worse still, the true label of the original input, or (2) break its grammaticality and naturality. Unfortunately, the change of true label will make the adversarial attack *invalid*. For example, supposing an adversary changes "she" to "he" in an input

[*] Indicates equal contribution. Yuan developed the method, designed and conducted most experiments; Fanchao formalized the task, designed some experiments and wrote the paper; Chenghao made the original research proposal, performed human evaluation and conducted some experiments.
[†] Work done during internship at Tsinghua University
[‡] Corresponding author

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6066–6080
July 5 - 10, 2020.

sentence to attack a gender identification model, although the victim model alters its prediction result, this is not a valid attack. And the adversarial examples with broken grammaticality and naturality (i.e., poor quality) can be easily defended (Pruthi et al., 2019).

Various textual adversarial attack models have been proposed (Wang et al., 2019a), ranging from character-level flipping (Ebrahimi et al., 2018) to sentence-level paraphrasing (Iyyer et al., 2018). Among them, word-level attack models, mostly word substitution-based models, perform comparatively well on both attack efficiency and adversarial example quality (Wang et al., 2019b).

Word-level adversarial attacking is actually a problem of *combinatorial optimization* (Wolsey and Nemhauser, 1999), as its goal is to craft adversarial examples which can successfully fool the victim model using a limited vocabulary. In this paper, as shown in Figure 1, we break this combinatorial optimization problem down into two steps including (1) reducing search space and (2) searching for adversarial examples.

The first step is aimed at excluding invalid or low-quality potential adversarial examples and retaining the valid ones with good grammaticality and naturality. The most common manner is to pick some candidate substitutes for each word in the original input and use their combinations as the reduced discrete search space. However, existing attack models either disregard this step (Papernot et al., 2016) or adopt unsatisfactory substitution methods that do not perform well in the trade-off between quality and quantity of the retained adversarial examples (Alzantot et al., 2018; Ren et al., 2019). The second step is supposed to find adversarial examples that can successfully fool the victim model in the reduced search space. Previous studies have explored diverse search algorithms including gradient descent (Papernot et al., 2016), genetic algorithm (Alzantot et al., 2018) and greedy algorithm (Ren et al., 2019). Some of them like gradient descent only work in the white-box setting where full knowledge of the victim model is required. In real situations, however, we usually have no access to the internal structures of victim models. As for the other black-box algorithms, they are not efficient and effective enough in searching for adversarial examples.

These problems negatively affect the overall attack performance of existing word-level adversarial attacking. To solve the problems, we propose a novel black-box word-level adversarial attack model, which reforms both the two steps. In the first step, we design a word substitution method based on *sememes*, the minimum semantic units, which can retain more potential valid adversarial examples with high quality. In the second step, we present a search algorithm based on *particle swarm optimization* (Eberhart and Kennedy, 1995), which is very efficient and performs better in finding adversarial examples. We conduct exhaustive experiments to evaluate our model. Experimental results show that, compared with baseline models, our model not only achieves the highest attack success rate (e.g., 100% when attacking BiLSTM on IMDB) but also possesses the best adversarial example quality and comparable attack validity. We also conduct decomposition analyses to manifest the advantages of the two parts of our model separately. Finally, we demonstrate that our model has the highest transferability and can bring the most robustness improvement to victim models by adversarial training.

2 Background

In this section, we first briefly introduce sememes, and then we give an overview of the classical particle swarm optimization algorithm.

2.1 Sememes

In linguistics, a sememe is defined as the minimum semantic unit of human languages (Bloomfield, 1926). The meaning of a word can be represented by the composition of its sememes.

In the field of NLP, sememe knowledge bases are built to utilize sememes in practical applications, where sememes are generally regarded as semantic labels of words (as shown in Figure 1). HowNet (Dong and Dong, 2006) is the most well-known one. It annotates over one hundred thousand English and Chinese words with a predefined sets of about 2,000 sememes. Its sememe annotations are sense-level, i.e., each sense of a (polysemous) word is annotated with sememes separately. With the help of HowNet, sememes have been successfully applied to many NLP tasks including word representation learning (Niu et al., 2017), sentiment analysis (Fu et al., 2013), semantic composition (Qi et al., 2019), sequence modeling (Qin et al., 2019), reverse dictionary (Zhang et al., 2019b), etc.

2.2 Particle Swarm Optimization

Inspired by the social behaviors like bird flocking, particle swarm optimization (PSO) is a kind of metaheuristic population-based evolutionary computation paradigms (Eberhart and Kennedy, 1995). It has been proved effective in solving the optimization problems such as image classification (Omran et al., 2004), part-of-speech tagging (Silva et al., 2012) and text clustering (Cagnina et al., 2014). Empirical studies have proven it is more efficient than some other optimization algorithms like the genetic algorithm (Hassan et al., 2005).

PSO exploits a population of interacting individuals to iteratively search for the optimal solution in the specific space. The population is called a *swarm* and the individuals are called *particles*. Each particle has a *position* in the search space and moves with an adaptable *velocity*.

Formally, when searching in a D-dimensional continuous space $\mathcal{S} \subseteq \mathbb{R}^D$ with a swarm containing N particles, the position and velocity of each particle can be represented by $\mathbf{x}^n \in \mathcal{S}$ and $\mathbf{v}^n \in \mathbb{R}^D$ respectively, $n \in \{1, \cdots, N\}$. Next we describe the PSO algorithm step by step.

(1) **Initialize.** At the very beginning, each particle is randomly initialized with a position $\mathbf{x}^n$ in the search space and a velocity $\mathbf{v}^n$. Each dimension of the initial velocity $v_d^n \in [-V_{max}, V_{max}]$, $d \in \{1, \cdots, D\}$.

(2) **Record.** Each position in the search space corresponds to an optimization score. The position a particle has reached with the highest optimization score is recorded as its *individual best position*. The best position among the individual best positions of all the particles is recorded as the *global best position*.

(3) **Terminate.** If current global best position has achieved the desired optimization score, the algorithm terminates and outputs the global best position as the search result.

(4) **Update.** Otherwise, the velocity and position of each particle are updated according to its current position and individual best position together with the global best position. The updating formulae are

$$\begin{aligned} v_d^n &= \omega v_d^n + c_1 \times r_1 \times (p_d^n - x_d^n) \\ &\quad + c_2 \times r_2 \times (p_d^g - x_d^n), \\ x_d^n &= x_d^n + v_d^n, \end{aligned} \tag{1}$$

where ω is the inertia weight, p_d^n and p_d^g are the d-th dimensions of the n-th particle's individual best position and the global best position respectively, c_1 and c_2 are acceleration coefficients which are positive constants and control how fast the particle moves towards its individual best position and the global best position, and r_1 and r_2 are random coefficients. After updating, the algorithm goes back to the **Record** step.

3 Methodology

In this section, we detail our word-level adversarial attack model. It incorporates two parts, namely the sememe-based word substitution method and PSO-based adversarial example search algorithm.

3.1 Sememe-based Word Substitution Method

The sememes of a word are supposed to accurately depict the meaning of the word (Dong and Dong, 2006). Therefore, the words with the same sememe annotations should have the same meanings, and they can serve as the substitutes for each other.

Compared with other word substitution methods, mostly including word embedding-based (Sato et al., 2018), language model-based (Zhang et al., 2019a) and synonym-based methods (Samanta and Mehta, 2017; Ren et al., 2019), the sememe-based word substitution method can achieve a better trade-off between quality and quantity of substitute words.

For one thing, although the word embedding and language model-based substitution methods can find as many substitute words as we want simply by relaxing the restrictions on embedding distance and language model prediction score, they inevitably introduce many inappropriate and low-quality substitutes, such as antonyms and semantically related but not similar words, into adversarial examples which might break the semantics, grammaticality and naturality of original input. In contrast, the sememe-based and, of course, the synonym-based substitution methods does not have this problem.

For another, compared with the synonym-based method, the sememe-based method can find more substitute words and, in turn, retain more potential adversarial examples, because HowNet annotates sememes for all kinds of words. The synonym-based method, however, depends on thesauri like WordNet (Miller, 1995), which provide no synonyms for many words like proper nouns and the number of a word's synonyms is very limited. An empirical comparison of different word substitution methods is given in Section 4.6.

In our sememe-based word substitution method, to preserve grammaticality, we only substitute content words[1] and restrict the substitutes to having the same part-of-speech tags as the original words. Considering polysemy, a word w can be substituted by another word w_* only if one of w's senses has the same sememe annotations as one of w_*'s senses. When making substitutions, we conduct lemmatization to enable more substitutions and delemmatization to avoid introducing grammatical mistakes.

3.2 PSO-based Adversarial Example Search Algorithm

Before presenting our algorithm, we first explain what the concepts in the original PSO algorithm correspond to in the adversarial example search problem.

Different from original PSO, the search space of word-level adversarial example search is discrete. A *position* in the search space corresponds to a sentence (or an adversarial example), and each dimension of a position corresponds to a word. Formally, $\mathbf{x}^n = w_1^n \cdots w_d^n \cdots w_D^n, w_d^n \in \mathbb{V}(w_d^o)$, where D is the length (word number) of the original input, w_d^o is the d-th word in the original input, and $\mathbb{V}(w_d^o)$ is composed of w_d^o and its substitutes.

The optimization score of a position is the *target label*'s prediction probability given by the victim model, where the target label is the desired classification result for an adversarial attack. Taking a binary classification task as an example, if the true label of the original input is "positive", the target label is "negative", and vice versa. In addition, a particle's *velocity* now relates to the position change probability, i.e., v_d^n determines how probable w_d^n is substituted by another word.

Next we describe our algorithm step by step.

First, for the **Initialize** step, since we expect the adversarial examples to differ from the original input as little as possible, we do not make random initialization. Instead, we randomly substitute one word of the original input to determine the initial position of a particle. This operation is actually the *mutation* of genetic algorithm, which has also been employed in some studies on discrete PSO (Higashi and Iba, 2003). We repeat mutation N times to initialize the positions of N particles. Each dimension of each particle's velocity is randomly initialized between $-V_{max}$ and V_{max}.

For the **Record** step, our algorithm keeps the same as the original PSO algorithm. For the **Terminate** step, the termination condition is the victim model predicts the target label for any of current adversarial examples.

For the **Update** step, considering the discreteness of search space, we follow Kennedy and Eberhart (1997) to adapt the updating formula of velocity to

$$v_d^n = \omega v_d^n + (1-\omega) \times [\mathcal{I}(p_d^n, x_d^n) + \mathcal{I}(p_d^g, x_d^n)], \tag{2}$$

where ω is still the inertia weight, and $\mathcal{I}(a, b)$ is defined as

$$\mathcal{I}(a,b) = \begin{cases} 1, & a = b, \\ -1, & a \neq b. \end{cases} \tag{3}$$

Following Shi and Eberhart (1998), we let the inertia weight decrease with the increase of numbers of iteration times, aiming to make the particles highly dynamic to explore more positions in the early stage and gather around the best positions quickly in the final stage. Specifically,

$$\omega = (\omega_{max} - \omega_{min}) \times \frac{T-t}{T} + \omega_{min}, \tag{4}$$

where $0 < \omega_{min} < \omega_{max} < 1$, and T and t are the maximum and current numbers of iteration times.

The updating of positions also needs to be adjusted to the discrete search space. Inspired by Kennedy and Eberhart (1997), instead of making addition, we adopt a probabilistic method to update the position of a particle to the best positions. We design two-step position updating. In the first step, a new movement probability P_i is introduced, with which a particle determines whether it moves to its *individual best position* as a whole. Once a particle decides to move, the change of each dimension of its position depends on the same dimension of its velocity, specifically with the probability of $\mathrm{sigmoid}(v_d^n)$. No matter whether a particle has moved towards its individual best position or not, it would be processed in the second step. In the second step, each particle determines whether to move to the *global best position* with another movement probability P_g. And the change of each position dimension also relies on $\mathrm{sigmoid}(v_d^n)$. P_i and P_g vary with iteration to enhance search efficiency by adjusting the balance between local and global search, i.e., encouraging particles to explore more

[1]Content words are the words that carry meanings and consist mostly of nouns, verbs, adjectives and adverbs.

Dataset	Task	#Class	Avg. #W	Train	Dev	Test	BiLSTM %ACC	BERT %ACC
IMDB	Sentiment Analysis	2	234	25000	0	25000	89.10	90.76
SST-2	Sentiment Analysis	2	17	6920	872	1821	83.75	90.28
SNLI	NLI	3	8	550152	10000	10000	84.43	89.58

Table 1: Details of datasets and their accuracy results of victim models. "#Class" means the number of classifications. "Avg. #W" signifies the average sentence length (number of words). "Train", "Val" and "Test" denote the instance numbers of the training, validation and test sets respectively. "BiLSTM %ACC" and "BERT %ACC" means the classification accuracy of BiLSTM and BERT.

space around their individual best positions in the early stage and search for better position around the global best position in the final stage. Formally,

$$\begin{aligned} P_i &= P_{max} - \frac{t}{T} \times (P_{max} - P_{min}), \\ P_g &= P_{min} + \frac{t}{T} \times (P_{max} - P_{min}), \end{aligned} \tag{5}$$

where $0 < P_{min} < P_{max} < 1$.

Besides, to enhance the search in unexplored space, we apply **mutation** to each particle after the update step. To avoid excessive modification, mutation is conducted with the probability

$$P_m(\mathbf{x}^n) = \min\left(0, 1 - k\frac{\mathcal{E}(\mathbf{x}^n, \mathbf{x}^o)}{D}\right), \tag{6}$$

where k is a positive constant, $\mathbf{x}^o$ represents the original input, and $\mathcal{E}$ measures the word-level edit distance (number of different words between two sentences). $\frac{\mathcal{E}(\mathbf{x}^n, \mathbf{x}^o)}{D}$ is defined as the *modification rate* of an adversarial example. After mutation, the algorithm returns to the **Record** step.

4 Experiments

In this section, we conduct comprehensive experiments to evaluate our attack model on the tasks of sentiment analysis and natural language inference.

4.1 Datasets and Victim Models

For sentiment analysis, we choose two benchmark datasets including IMDB (Maas et al., 2011) and SST-2 (Socher et al., 2013). Both of them are binary sentiment classification datasets. But the average sentence length of SST-2 (17 words) is much shorter than that of IMDB (234 words), which renders attacks on SST-2 more challenging. For natural language inference (NLI), we use the popular Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015). Each instance in SNLI comprises a premise-hypothesis sentence pair and is labelled one of three relations including entailment, contradiction and neutral.

As for victim models, we choose two widely used universal sentence encoding models, namely bidirectional LSTM (BiLSTM) with max pooling (Conneau et al., 2017) and $\text{BERT}_{\text{BASE}}$ (BERT) (Devlin et al., 2019). For BiLSTM, its hidden states are 128-dimensional, and it uses 300-dimensional pre-trained GloVe (Pennington et al., 2014) word embeddings. Details of the datasets and the classification accuracy results of the victim models are listed in Table 1.

4.2 Baseline Methods

We select two recent open-source word-level adversarial attack models as the baselines, which are typical and involve different search space reduction methods (step 1) and search algorithms (step 2).

The first baseline method (Alzantot et al., 2018) uses the combination of restrictions on word embedding distance and language model prediction score to reduce search space. As for search algorithm, it adopts genetic algorithm, another popular metaheuristic population-based evolutionary algorithm. We use "**Embedding/LM+Genetic**" to denote this baseline method.

The second baseline (Ren et al., 2019) chooses synonyms from WordNet (Miller, 1995) as substitutes and designs a saliency-based greedy algorithm as the search algorithm. We call this method "**Synonym+Greedy**". This baseline model is very similar to another attack model TextFooler (Jin et al., 2019), which has extra semantic similarity checking when searching adversarial examples. But we find the former performs better in almost all experiments, and thus we only select the former as a baseline for comparison.

In addition, to conduct decomposition analyses of different methods in the two steps separately, we combine different search space reduction methods (Embedding/LM, Synonym and our sememe-based substitution method (Sememe)), and search algorithms (Genetic, Greedy and our PSO).

Metrics	Evaluation Method	Better?
Success Rate	Auto	Higher
Validity	Human (Valid Attack Rate)	Higher
Modification Rate	Auto	Lower
Grammaticality	Auto (Error Increase Rate)	Lower
Fluency	Auto (Perplexity)	Lower
Naturality	Human (Naturality Score)	Higher

Table 2: Details of evaluation metrics. "Auto" and "Human" represent automatic and human evaluations respectively. "Higher" and "Lower" mean the higher/lower the metric, the better a model performs.

4.3 Experimental Settings

For our PSO, V_{max} is set to 1, ω_{max} and ω_{min} are set to 0.8 and 0.2, P_{max} and P_{min} are also set to 0.8 and 0.2, and k in Equation (6) is set to 2. All these hyper-parameters have been tuned on the validation set. For the baselines, we use their recommended hyper-parameter settings. For the two population-based search algorithms Genetic and PSO, we set the maximum number of iteration times (T in Section 3.2) to 20 and the population size (N in Section 3.2) to 60, which are the same as Alzantot et al. (2018).

4.4 Evaluation Metrics

To improve evaluation efficiency, we randomly sample $1,000$ correctly classified instances from the test sets of the three datasets as the original input to be perturbed. For SNLI, only the hypotheses are perturbed. Following Alzantot et al. (2018), we restrict the length of the original input to 10-100, exclude the out-of-vocabulary words from the substitute sets, and discard the adversarial examples with modification rates higher than 25%.

We evaluate the performance of attack models including their attack success rates, attack validity and the quality of adversarial examples. The details of our evaluation metrics are listed in Table 2.

(1) The attack success rate is defined as the percentage of the attacks which craft an adversarial example to make the victim model predict the target label. (2) The attack validity is measured by the percentage of valid attacks to successful attacks, where the adversarial examples crafted by valid attacks have the same true labels as the original input. (3) For the quality of adversarial examples, we divide it into four parts including modification rate, grammaticality, fluency and naturality. *Grammaticality* is measured by the increase rate of grammatical error numbers of adversarial examples compared with the original input, where we use LanguageTool[2] to obtain the grammatical error number of a sentence. We utilize the language model perplexity (PPL) to measure the *fluency* with the help of GPT-2 (Radford et al., 2019). The *naturality* reflects whether an adversarial example is natural and indistinguishable from human-written text.

We evaluate attack validity and adversarial example naturality only on SST-2 by human evaluation with the help of Amazon Mechanical Turk[3]. We randomly sample 200 adversarial examples, and ask the annotators to make a binary sentiment classification and give a naturality score (1, 2 or 3, higher better) for each adversarial example and original input. More annotation details are given in Appendix A.

4.5 Attack Performance

Attack Success Rate The attack success rate results of all the models are listed in Table 3. We observe that our attack model (Sememe+PSO) achieves the highest attack success rates on all the three datasets (especially the harder SST-2 and SNLI) and two victim models, proving the superiority of our model over baselines. It attacks BiLSTM/BERT on IMDB with a notably 100.00%/98.70% success rate, which clearly demonstrates the vulnerability of DNNs. By comparing three word substitution methods (search space reduction methods) and three search algorithms, we find Sememe and PSO consistently outperform their counterparts. Further decomposition analyses are given in a later section.

Validity and Adversarial Example Quality We evaluate the attack validity and adversarial example quality of our model together with the two baseline methods (Embedding/LM+Genetic and Synonym+Greedy). The results of automatic and human evaluations are displayed in Table 4 and 5 respectively.[4] Note that the human evaluations including attack validity and adversarial example naturality are conducted on SST-2 only. We find that in terms of automatic evaluations of adversarial example quality, including modification rate, grammaticality and fluency, our model consistently outperforms the two baselines on whichever victim model and dataset. As for attack validity and adver-

[2] https://www.languagetool.org

[3] https://www.mturk.com

[4] Automatic evaluation results of adversarial example quality of all the combination models are shown in Appendix B.

Word Substitution Method	Search Algorithm	BiLSTM			BERT		
		IMDB	SST-2	SNLI	IMDB	SST-2	SNLI
Embedding/LM	Genetic	86.90	67.70	44.40	87.50	66.20	44.30
	Greedy	80.90	69.00	47.70	62.50	56.20	42.40
	PSO	96.90	78.50	50.90	93.60	74.40	53.10
Synonym	Genetic	95.50	73.00	51.40	92.90	78.40	56.00
	Greedy	87.20	73.30	57.70	73.00	64.60	52.70
	PSO	98.70	79.20	61.80	96.20	80.90	62.60
Sememe	Genetic	96.90	78.50	50.90	93.60	74.40	53.10
	Greedy	95.20	87.70	70.40	80.50	74.80	66.30
	PSO	**100.00**	**93.80**	**73.40**	**98.70**	**91.20**	**78.90**

Table 3: The attack success rates (%) of different attack models.

Victim Model	Attack Model	IMDB			SST-2			SNLI		
		%M	%I	PPL	%M	%I	PPL	%M	%I	PPL
BiLSTM	Embedding/LM+Genetic	9.76	5.49	124.20	12.03	7.08	319.98	13.31	14.12	235.20
	Synonym+Greedy	6.47	4.49	115.31	10.25	4.65	317.27	12.32	21.37	311.04
	Sememe+PSO	**3.71**	**1.44**	**88.98**	**9.06**	**3.17**	**276.53**	**11.72**	**11.08**	**222.40**
BERT	Embedding/LM+Genetic	7.41	4.22	106.12	10.41	5.09	314.22	13.04	15.09	225.92
	Synonym+Greedy	4.49	4.48	98.60	8.51	4.11	316.30	**11.60**	11.65	285.00
	Sememe+PSO	**3.69**	**1.57**	**90.74**	**8.24**	**2.03**	**289.94**	11.72	**10.14**	**223.22**

Table 4: Automatic evaluation results of adversarial example quality. "%M", "%I" and "PPL" indicate the modification rate, grammatical error increase rate and language model perplexity respectively.

Victim	Attack Model	%Valid	NatScore
N/A	Original Input	90.0	2.30
BiLSTM	Embedding/LM+Genetic	65.5	2.205
	Synonym+Greedy	**72.0**	2.190
	Sememe+PSO	70.5	**2.210**
BERT	Embedding/LM+Genetic	**74.5**	2.165
	Synonym+Greedy	66.5	2.165
	Sememe+PSO	72.0	**2.180**

Table 5: Human evaluation results of attack validity and adversarial example naturality on SST-2, where the second row additionally lists the evaluation results of original input. "%Valid" refers to the percentage of valid attacks. "NatScore" is the average naturality score of adversarial examples.

sarial example naturality, our Sememe+PSO model obtains a slightly higher overall performance than the two baselines. But its adversarial examples are still inferior to original human-authored input, especially in terms of validity (label consistency).

We conduct Student's *t*-tests to further measure the difference between the human evaluation results of different models, where the statistical significance threshold of *p*-value is set to 0.05. We find that neither of the differences of attack validity and adversarial example naturality between different models are significant. In addition, the adversarial examples of any attack model have significantly worse label consistency (validity) than the original input, but possesses similar naturality. More details of statistical significance test are given in Appendix D.

For Embedding/LM, relaxing the restrictions on embedding distance and language model prediction score can improve its attack success rate but sacrifices attack validity. To make a specific comparison, we adjust the hyper-parameters of Embedding/LM+Genetic[5] to increase its attack success rates to 96.90%, 90.30%, 58.00%, 93.50%, 83.50% and 62.90% respectively on attacking the two victim models on the three datasets (in the same order as Table 3). Nonetheless, its attack validity rates against BiLSTM and BERT on SST-2 dramatically fall to 59.5% and 56.5%. In contrast, ours are 70.5% and 72.0%, and their differences are significant according to the results of significance tests in Appendix D.

4.6 Decomposition Analyses

In this section, we conduct detailed decomposition analyses of different word substitution methods (search space reduction methods) and different search algorithms, aiming to further demonstrate the advantages of our sememe-based word substitution method and PSO-based search algorithm.

[5]The detailed hyper-parameter settings are given in Appendix C.

Word Substitution Method	IMDB	SST-2	SNLI
Embedding/LM	3.44	3.27	3.42
Synonym	3.55	3.08	3.14
Sememe	**13.92**	**10.97**	**12.87**

Table 6: The average number of substitutes provided by different word substitution methods.

She breaks the pie dish and screams out that she is not handicapped.		
Embedding/LM	Synonym	Sememe
tart, pizza, apple, shoemaker, cake cheesecake	None	cheese, popcorn, ham, cream, break, cake, pizza, chocolate, and 55 more

Table 7: A real case showing the substitutes found by three word substitution methods, where the original word is colored green and appropriate substitutes are colored red.

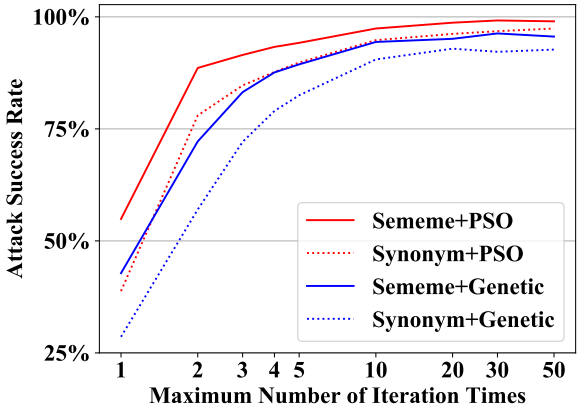

Figure 2: Attack success rates of different models with different maximum numbers of iteration times. The x-coordinate is in log-2 scale.

Word Substitution Method Table 6 lists the average number of substitutes provided by different word substitution methods on the three datasets. It shows Sememe can find much more substitutes than the other two counterparts, which explains the high attack success rates of the models incorporating Sememe. Besides, we give a real case from SST-2 in Table 7 which lists substitutes found by the three methods. We observe that Embedding/LM find many improper substitutes, Synonym cannot find any substitute because the original word "pie" has no synonyms in WordNet, and only Sememe finds many appropriate substitutes.

Search Algorithm We compare the two population-based search algorithms Genetic and PSO by changing two important hyper-parameters, namely the maximum number of iteration times T and the population size N. The results of attack success rate are shown in Figure 2 and 3. From the two figures, we find our PSO outperforms Genetic

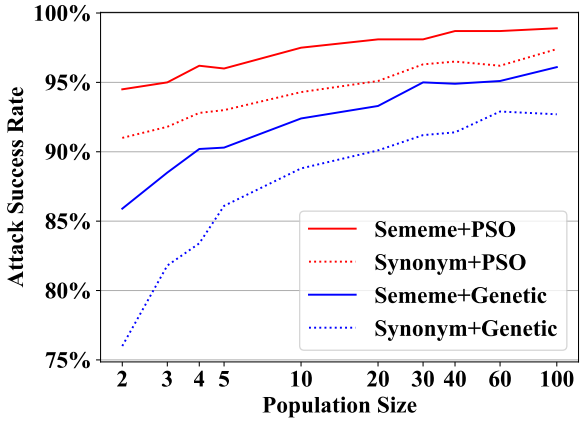

Figure 3: Attack success rates of different models with population sizes. The x-coordinate is in log-2 scale.

Transfer	Attack Model	IMDB	SST-2	SNLI
BiLSTM	Embedding/LM+Genetic	81.93	70.61	61.26
⇓	Synonym+Greedy	77.29	64.94	65.34
BERT	Sememe+PSO	**75.80**	**64.71**	**59.54**
BERT	Embedding/LM+Genetic	86.63	65.71	49.66
⇓	Synonym+Greedy	81.64	58.67	**45.16**
BiLSTM	Sememe+PSO	**78.42**	**58.11**	46.89

Table 8: The classification accuracy of transferred adversarial examples on the three datasets. Lower accuracy reflects higher transferability.

consistently, especially in the setting with severe restrictions on maximum number of iteration times and population size, which highlights the efficiency of PSO.

4.7 Transferability

The transferability of adversarial examples reflects whether an attack model can attack a DNN model without any access to it (Kurakin et al., 2016). It has been widely used as an important evaluation metric in adversarial attacks. We evaluate the transferability of adversarial examples by using BiLSTM to classify the adversarial examples crafted for attacking BERT, and vice versa. Table 8 shows the classification accuracy results of transferred adversarial examples. Note that lower accuracy signifies higher transferability. The lower the accuracy is, the higher the transferability is. We find compared with the two baselines, our Sememe+PSO crafts adversarial examples with overall higher transferability.

4.8 Adversarial Training

Adversarial training is proposed to improve the robustness of victim models by adding adversarial examples to the training set (Goodfellow et al., 2015). In this experiment, for each attack model,

we craft 692 adversarial examples (10% of the original training set size) by using it to attack BiLSTM on the *training* set of SST-2. Then we add the adversarial examples to the training set and retrain a BiLSTM. We re-evaluate its robustness by calculating the attack success rates of different attack models. Table 9 lists the results of adversarial training. Note larger attack success rate decrease signifies greater robustness improvement. We find that adversarial training can improve the robustness of victim models indeed, and our Sememe+PSO model brings greater robustness improvement than the two baselines, even when the attack models are exactly themselves.[6] From the perspective of attacking, our Sememe+PSO model is still more threatening than others even under the defense of adversarial training.

We also manually select 692 *valid* adversarial examples generated by Sememe+PSO to conduct adversarial training, which leads to even greater robustness improvement (last column of Table 9). The results show that adversarial example validity has big influence on adversarial training effect.

5 Related Work

Existing textual adversarial attack models can be classified into three categories according to the perturbation levels of their adversarial examples.

Sentence-level attacks include adding distracting sentences (Jia and Liang, 2017), paraphrasing (Iyyer et al., 2018; Ribeiro et al., 2018) and performing perturbations in the continuous latent semantic space (Zhao et al., 2018). Adversarial examples crafted by these methods usually have profoundly different forms from original input and their validity are not guaranteed.

Character-level attacks are mainly random character manipulations including swap, substitution, deletion, insertion and repeating (Belinkov and Bisk, 2018; Gao et al., 2018; Hosseini et al., 2017). In addition, gradient-based character substitution methods have also been explored, with the help of one-hot character embeddings (Ebrahimi et al., 2018) or visual character embeddings (Eger et al., 2019). Although character-level attacks can achieve high success rates, they break the grammaticality and naturality of original input and can be easily defended (Pruthi et al., 2019).

[6]For instance, using Embedding/LM+Genetic in adversarial training to defend its attack declines the attack success rate by 2.60% while using our Sememe+PSO model declines by 3.53%.

Att \Adv.T	None	E/L+G	Syn+G	Sem+P	Sem+P*
E/L+G	67.70	-2.60	-0.60	-3.53	-5.10
Syn+G	73.30	-2.67	-3.50	-3.13	-3.53
Sem+P	93.80	-1.07	0.03	-2.93	-4.33

Table 9: The attack success rates of different attack models when attacking BiLSTM on SST-2 and their decrements brought by adversarial training. "Att" and "Adv.T" denote "Attack Model" and "Adversarial Training". E/L+G, Syn+G and Sem+P represent Embedding/LM+Genetic, Synonym+Greedy and our Sememe+PSO, respectively. "Sem+P*" denotes only using the *valid* adversarial examples generated by Sememe+PSO in adversarial training.

As for word-level attacks, following our two-step modeling, their adversarial example space reduction methods (step 1) involve using word embeddings (Sato et al., 2018) or language model (Zhang et al., 2019a) to filter words, selecting synonyms as substitutes (Samanta and Mehta, 2017; Ren et al., 2019; Jin et al., 2019), and their combinations (Alzantot et al., 2018; Glockner et al., 2018). The search algorithms (step 2) include gradient descent (Papernot et al., 2016; Sato et al., 2018; Gong et al., 2018), genetic algorithm (Alzantot et al., 2018), Metropolis-Hastings sampling (Zhang et al., 2019a), saliency-based greedy algorithm (Liang et al., 2018; Ren et al., 2019; Jin et al., 2019). In comparison, our model adopts new methods in both steps which are more powerful.

6 Conclusion and Future Work

In this paper, we propose a novel word-level attack model comprising the sememe-based word substitution method and particle swarm optimization-based search algorithm. We conduct extensive experiments to demonstrate the superiority of our model in terms of attack success rate, adversarial example quality, transferability and robustness improvement to victim models by adversarial training. In the future, we will try to increase the robustness gains of adversarial training and consider utilizing sememes in adversarial defense model.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004503) and the National Natural Science Foundation of China (NSFC No. 61732008, 61772302). We also thank the anonymous reviewers for their valuable comments and suggestions.

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 EMNLP*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of ICLR*.

Leonard Bloomfield. 1926. A set of postulates for the science of language. *Language*, 2(3).

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*.

Leticia Cagnina, Marcelo Errecalde, Diego Ingaramo, and Paolo Rosso. 2014. An efficient particle swarm optimization approach to cluster short texts. *Information Sciences*, 265:36–49.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Zhendong Dong and Qiang Dong. 2006. *HowNet and the computation of meaning*. World Scientific.

Russell Eberhart and James Kennedy. 1995. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of ACL*.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of NAACL-HLT*.

Xianghua Fu, Guo Liu, Yanyan Guo, and Zhiqiang Wang. 2013. Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon. *Knowledge-Based Systems*, 37:186–195.

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

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of ACL*.

Zhitao Gong, Wenlu Wang, Bo Li, Dawn Song, and Wei-Shinn Ku. 2018. Adversarial texts with gradient methods. *arXiv preprint arXiv:1801.07175*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of ICLR*.

Rania Hassan, Babak Cohanim, Olivier De Weck, and Gerhard Venter. 2005. A comparison of particle swarm optimization and the genetic algorithm. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*.

Natsuki Higashi and Hitoshi Iba. 2003. Particle swarm optimization with gaussian mutation. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the NAACL-HLT*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT really robust? Natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.

James Kennedy and Russell C Eberhart. 1997. A discrete binary version of the particle swarm algorithm. In *IEEE International Conference on Systems, Man, and Cybernetics*.

J Zico Kolter and Marcus A Maloof. 2006. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research*, 7(Dec):2721–2744.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of IJCAI*.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL-HLT*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Improved word representation learning with sememes. In *Proceedings ACL*.

Mahamed G Omran, Andries P Engelbrecht, and Ayed Salman. 2004. Image classification using particle swarm optimization. In *Recent advances in simulated evolution and learning*, pages 347–365. World Scientific.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Proceedings of MILCOM*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of ACL*.

Fanchao Qi, Junjie Huang, Chenghao Yang, Zhiyuan Liu, Xiao Chen, Qun Liu, and Maosong Sun. 2019. Modeling semantic compositionality with sememe knowledge. In *Proceedings of ACL*.

Yujia Qin, Fanchao Qi, Sicong Ouyang, Zhiyuan Liu, Cheng Yang, Yasheng Wang, Qun Liu, and Maosong Sun. 2019. Enhancing recurrent neural networks with sememes. *arXiv preprint arXiv:1910.08910*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of ACL*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of ACL*.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. In *Proceedings of IJCAI*.

Yuhui Shi and Russell C Eberhart. 1998. Parameter selection in particle swarm optimization. In *Proceedings of the International conference on evolutionary programming*. Springer.

Ana Paula Silva, Arlindo Silva, and Irene Rodrigues. 2012. Biopos: Biologically inspired algorithms for pos tagging. In *Proceedings of the First International Workshop on Optimization Techniques for Human Language Technology*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2010. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of ICLR*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of EMNLP-IJCNLP*.

Wenqi Wang, Lina Wang, Benxiao Tang, Run Wang, and Aoshuang Ye. 2019a. A survey: Towards a robust deep neural network in text domain. *arXiv preprint arXiv:1902.07285*.

Xiaosen Wang, Hao Jin, and Kun He. 2019b. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723*.

Laurence A Wolsey and George L Nemhauser. 1999. *Integer and combinatorial optimization*. John Wiley & Sons.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019a. Generating fluent adversarial examples for natural languages. In *Proceedings of ACL*.

Lei Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2019b. Multi-channel reverse dictionary model. *arXiv preprint arXiv:1912.08441*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of ICLR*.

A Human Evaluation Details

For each adversarial example and original input, we ask three workers to choose a sentiment label from "Positive" and "Negative" for it and annotate its naturality score from $\{1, 2, 3\}$, which indicates "Machine generated", "Not sure" and "Human written" respectively. We get the final sentiment labels of adversarial examples by voting. For example, if two workers annotate an example as "Positive" and one worker annotates it as "Negative", we record its annotated label as "Positive". We obtain the validity rate by calculating the percentage of the adversarial examples which are annotated with the same sentiment labels as corresponding original

Victim Model	Word Substitution Method	Search Algorithm	IMDB			SST-2			SNLI		
			%M	%I	PPL	%M	%I	PPL	%M	%I	PPL
BiLSTM	Embedding/LM	Genetic	9.76	5.49	124.20	12.03	7.08	319.98	13.31	14.12	235.20
		Greedy	7.84	5.23	112.84	10.63	3.71	287.45	12.60	**9.42**	**205.50**
		PSO	7.00	8.07	113.99	12.78	6.68	339.46	14.82	10.82	255.69
	Synonym	Genetic	7.60	6.07	137.51	11.35	5.32	357.19	12.60	24.78	283.95
		Greedy	6.47	4.49	115.31	10.25	4.65	317.27	12.32	21.37	311.04
		PSO	5.42	3.45	109.27	10.55	5.12	331.96	12.56	20.83	307.51
	Sememe	Genetic	5.30	2.55	105.24	10.04	3.48	298.49	**11.30**	11.64	205.61
		Greedy	4.89	1.80	97.49	9.36	**2.79**	**276.53**	12.11	10.95	218.72
		PSO	**3.71**	**1.44**	**88.98**	**9.06**	3.17	**276.53**	11.72	11.08	222.40
BERT	Embedding/LM	Genetic	7.41	4.22	106.12	10.41	5.09	314.22	13.04	15.09	225.92
		Greedy	5.53	4.45	97.21	9.23	3.04	**276.42**	11.80	13.73	**206.46**
		PSO	5.97	7.98	101.66	11.64	6.70	343.89	14.22	14.43	245.95
	Synonym	Genetic	5.72	5.59	114.57	9.62	4.62	353.05	13.09	13.01	311.14
		Greedy	4.49	4.48	98.60	8.51	4.12	316.30	11.60	11.65	285.00
		PSO	4.63	4.33	100.81	9.20	4.72	337.82	12.99	13.32	302.83
	Sememe	Genetic	4.27	1.62	97.86	8.34	2.05	292.16	11.59	8.84	217.75
		Greedy	3.97	1.79	92.31	**8.14**	2.21	279.35	**10.09**	**7.81**	207.71
		PSO	**3.69**	**1.57**	**90.74**	8.24	**2.03**	289.94	11.73	10.14	223.22

Table 10: Automatic evaluation results of adversarial example quality. "%M", "%I" and "PPL" indicate the modification rate, grammatical error increase rate and language model perplexity respectively.

sentences. For each adversarial example, we use the average of the naturality scores given by three workers as its final naturality score.

B Automatic Evaluation Results of Adversarial Example Quality

We present the automatic evaluation results of adversarial example quality of all the combination models in Table 10. We can find that Sememe and PSO obtain higher overall adversarial example quality than other word substitution methods and adversarial example search algorithms, respectively.

C Adjustment of Hyper-parameters of Embedding/LM+Genetic

The word substitution strategy Embedding/LM has three hyper-parameters: the number of the nearest words N, the euclidean distance threshold of word embeddings δ and the number of words retained by the language model filtering K. For original Embedding/LM+Genetic, $\mathrm{N} = 8$, $\delta = 0.5$ and $\mathrm{K} = 4$, which are the same as Alzantot et al. (2018). To increase the attack success rates, we change these hyper-parameters to $\mathrm{N} = 20$, $\delta = 1$ and $\mathrm{K} = 10$.

D Statistical Significance of Human Evaluation Results

We conduct Student's t-tests to measure the statistical significance between the difference of human evaluation results of different models. The results of attack validity and adversarial example naturality are shown in Table 11 and 12, respectively. "Embedding/LM+Genetic*" refers to the Embedding/LM+Genetic model with adjusted hyper-parameters.

E Case Study

We display some adversarial examples generated by the baseline attack models and our attack model on IMDB, SST-2 and SNLI in Table 13, 14 and 15 respectively.

Victim Model	Model 1	Model 2	p-value	Significance	Conclusion
Bi-LSTM	Sememe+PSO	Embedding/LM+Genetic	0.14	Not Significant	=
	Sememe+PSO	Synonym+Greedy	0.37	Not Significant	=
	Sememe+PSO	Embedding/LM+Genetic*	0.01	Significant	>
	Original Input	Embedding/LM+Genetic	9.52e-10	Significant	>
	Original Input	Synonym+Greedy	1.78e-6	Significant	>
	Original Input	Sememe+PSO	3.55e-7	Significant	>
	Original Input	Embedding/LM+Genetic*	2.42e-13	Significant	>
BERT	Sememe+PSO	Embedding/LM+Genetic	0.29	Not Significant	=
	Sememe+PSO	Synonym+Greedy	0.12	Not Significant	=
	Sememe+PSO	Embedding/LM+Genetic*	5.86e-4	Significant	>
	Original Input	Embedding/LM+Genetic	2.19e-5	Significant	>
	Original Input	Synonym+Greedy	3.33e-9	Significant	>
	Original Input	Sememe+PSO	1.78e-6	Significant	>
	Original Input	Embedding/LM+Genetic*	2.30e-15	Significant	>

Table 11: The Student's t-test results of attack validity of different models, where "=" means "Model 1" performs as well as "Model 2" and ">" means "Model 1" performs better than "Model 2".

Victim Model	Model 1	Model 2	p-value	Significance	Conclusion
Bi-LSTM	Sememe+PSO	Embedding/LM+Genetic	0.48	Not Significant	=
	Sememe+PSO	Synonym+Greedy	0.41	Not Significant	=
	Human Authored	Embedding/LM+Genetic	0.14	Not Significant	=
	Human Authored	Synonym+Greedy	0.10	Not Significant	=
	Human Authored	Sememe+PSO	0.15	Not Significant	=
BERT	Sememe+PSO	Embedding/LM+Genetic	0.31	Not Significant	=
	Sememe+PSO	Synonym+Greedy	0.31	Not Significant	=
	Human Authored	Embedding/LM+Genetic	0.06	Not Significant	=
	Human Authored	Synonym+Greedy	0.06	Not Significant	=
	Human Authored	Sememe+PSO	0.08	Not Significant	=

Table 12: The Student's t-test results of adversarial example naturality of different models, where "=" means "Model 1" performs as well as "Model 2".

IMDB Example 1
Original Input (Prediction = **Positive**)
In my opinion this is the best oliver stone flick probably more because of influence than anything else. Full of dread from the first moment to its dark ending.
Embedding/LM+Genetic (Prediction = **Negative**)
In my view this is the higher oliver stone flick presumably more because of influence than anything else. Total of anxiety from the first moment to its dark ending.
Synonym+Greedy (Prediction = **Negative**)
In my opinion this embody the respectable oliver stone flick probably more because of influence than anything else. Broad of dread from the first moment to its dark ending.
Sememe+PSO (Prediction = **Negative**)
In my opinion this is the bestest oliver stone flick probably more because of influence than anything else. Ample of dread from the first moment to its dark ending.
IMDB Example 2
Original Input (Prediction = **Negative**)
One of the worst films of it's genre. The only bright spots were lee showing some of the sparkle she would later bring to the time tunnel and batman.
Embedding/LM+Genetic (Prediction = **Positive**)
One of the biggest films of it's genre. The only glittering spots were lee showing some of the sparkle she would afterwards bring to the time tunnel and batman.
Synonym+Greedy (Prediction = **Positive**)
One of the tough films of it's genre. The only bright spots follow lee present some of the spark she would later bring to the time tunnel and batman.
Sememe+PSO (Prediction = **Positive**)
One of the seediest films of it's genre. The only shimmering spots were lee showing some of the sparkle she would later bring to the time tunnel and batman.

Table 13: Adversarial examples generated by two baseline methods and our model on IMDB.

SST-2 Example 1
Original Input (Prediction = **Positive**)
Some actors have so much charisma that you 'd be happy to listen to them reading the phone book.
Embedding/LM+Genetic (Prediction = **Negative**)
Some actors have so much charisma that you 'd be cheery to listen to them reading the phone book.
Synonym+Greedy (Prediction = **Negative**)
Some actors have so much charisma that you 'd be happy to listen to them take the phone book.
Sememe+PSO (Prediction = **Negative**)
Some actors have so much charisma that you 'd be jovial to listen to them reading the phone book.
SST-2 Example 2
Original Sentence (Prediction = **Negative**)
The movie 's biggest is its complete and utter lack of tension.
Embedding/LM+Genetic (Prediction = **Positive**)
The movie 's biggest is its complete and utter absence of stress.
Synonym+Greedy (Prediction = **Positive**)
The movie 's great is its complete and utter want of tension.
Sememe+PSO (Prediction = **Positive**)
The movie 's biggest is its complete and utter dearth of tension.

Table 14: Adversarial examples generated by two baseline methods and our model on SST-2.

SNLI Example 1
Premise: A smiling bride sits in a swing with her smiling groom standing behind her posing for the male photographer while a boy holding a bottled drink and another boy wearing a green shirt observe .
Original Input(Prediction = **Entailment**)
Two boys look on as a married couple get their pictures taken.
Embedding/LM+Genetic (Prediction = **Contradiction**)
Two man stare on as a wedding couple get their pictures taken.
Synonym+Greedy (Prediction = **Contradiction**)
Two boys look on as a married couple puzzle their pictures taken.
Sememe+PSO (Prediction = **Contradiction**)
Two boys stare on as a wedding couple get their pictures taken.
SNLI Example 2
Premise: A dog with a purple leash is held by a woman wearing white shoes .
Original Input (Prediction = **Entailment**)
A man is holding a leash on someone else dog.
Embedding/LM+Genetic (Prediction = **Contradiction**)
A man is holding a leash on someone further dog.
Synonym+Greedy (Prediction = **Contradiction**)
A humans is holding a leash on someone else dog.
Sememe+PSO (Prediction = **Contradiction**)
A man is holding a leash on someone else canine.

Table 15: Adversarial examples generated by two baseline methods and our model on SNLI.