SSCAE: A Novel Semantic, Syntactic, and Context-Aware Natural Language Adversarial Example Generator

Anonymous NAACL-HLT 2021 submission

Abstract

Training a machine learning model with adversarial examples (AEs) improves its robustness against adversarial attacks. Hence, it is crucial to develop effective generative models to produce high-quality AEs. Developing such models has been much slower in natural language processing (NLP). The current state-of-the-art in NLP generates AEs that are somehow human detectable and/or include semantic and linguistic defects. This paper introduces a novel, practical, and efficient adversarial attack model called SSCAE for **S**emantic, **S**yntactic, and **C**ontext-aware natural language **A**dversarial **E**xamples generator. SSCAE generates humanly imperceptible context-aware AEs that preserve semantic consistency and source language's syntactical and grammatical requirements. The effectiveness and superiority of the proposed SSCAE model are illustrated over eleven comparative experiments, extensive ablation studies, and human evaluations.

1 Introduction

Machine learning vulnerability to Adversarial Examples (AEs), i.e., maliciously crafted perturbations, remains an open area of research (Goodfellow et al., 2015; Kurakin et al., 2016; Zhang et al., 2020). These humanly imperceptible perturbations force a fine-tuned machine learning model (dubbed as a target model) to produce wrong decisions that align with attackers' intentions. Recent research shows that introducing AEs to a target model during the training, referred to as adversarial training, improves the robustness and stability of that model against adversarial attacks (Shafahi et al., 2020; Wang et al., 2021; Xu et al., 2020). While adversarial attack/defense studies have largely contributed to machine vision (Chakraborty et al., 2021; Goodfellow et al., 2015; Papernot et al., 2017), the progress of such studies in natural language processing (NLP) has been at a slower pace. Designing practical adversarial attack/defense techniques in NLP is more challenging due to the discrete nature of the text (Jin et al., 2020).

A well-crafted AE fools the target model while establishing three essential principles: (1) compatibility with the human decision (Jin et al., 2020; Li et al., 2020), (2) preserving the semantic consistency of the original text (Jin et al., 2020; Song et al., 2021), and (3) following the source language's syntactic and grammatical rules (Jin et al., 2020; Song et al., 2021). TextFooler (Jin et al., 2020), as a baseline adversarial attack model in NLP, first employs a word embeddings technique to explore potential synonyms for each important word. Next, it applies grammatical and semantic checks to narrow down the synonyms and find proper substitutions to serve as perturbations. However, it produces complicated out-of-context replacements (Garg and Ramakrishnan, 2020; Li et al., 2020). To address this problem, two recent adversarial attack models, BERT-Attack (Li et al., 2020) and BERT-based AEs (BAE) (Garg and Ramakrishnan, 2020), were proposed to generate contextual perturbations by masking and replacing important words with substitutions produced by BERT Masked Language Model (BERT MLM) (Devlin et al., 2019). Although the generated perturbations look context-aware, the original text's semantic and syntactic characteristics are sometimes lost (Li et al., 2020). There is a need for a comprehensive model that simultaneously considers all principles mentioned above for well-crafted perturbations.

This paper introduces a novel, practical, and efficient adversarial attack model referred to as SSCAE for Semantic (principle 2), Syntactic (principle 3), and Context-aware (principle 1) natural language AEs generator. SSCAE generates humanly imperceptible context-aware AEs that preserve semantic consistency and source language's syntactical and grammatical requirements. It first employs the BERT MLM to generate an initial set of substitution candidates. Next, it applies two language models, Universal Sentence Encoder (USE) (Cer et al., 2018) and Generative Pre-trained Transformer 2 (GPT-2) (Radford et al., 2019) to evaluate the initial set in terms of semantic and syntactic characteristics, respectively (more details are available in Appendix A). To do so, dynamic thresholds are utilized to capture more efficient perturbations than static thresholding, the focus of similar literature (Jin et al., 2020; Kuleshov et al., 2018; Li et al., 2018). Eleven computational experiments were designed employing frequently used text classification and natural language inference (NLI) datasets to (1) illustrate SSCAE performance as compared with TextFooler, BERT-Attack, and BAE using the BERT target model (seven experiments), and (2) illustrate SSCAE effectiveness on other target models (four experiments). SSCAE outperforms the other methods in all experiments while maintaining a higher semantic consistency with a lower query number and a comparable perturbation rate. Moreover, a human evaluation study verifies the automatic adversarial attack experiments in generating high-quality and fluent perturbations.

2 Related Work

This section categorizes adversarial attacks on textual data into two white-box and black-box groups:

In a black-box setting, the proposed approaches range from character-level to sentence-level techniques where the word-level methods demonstrate their superiority compared to other approaches (Jia and Liang, 2017; Li et al., 2018; Zhang et al., 2020). Jia and Liang (Jia and Liang, 2017) proposed concatenative adversaries to append distracting sentences at the end of a paragraph to attack the Stanford Question Answering reading comprehension system. Belinkov et al. (Belinkov and Bisk, 2018) applied two types of synthetic (character order changes) and natural (typos and misspellings) noises to the input of the Neural Machine Translation (NMT) models to produce AEs for NMT systems. Jin et al. (Jin et al., 2020) introduced TextFooler, identifying the important words, gathering a candidate set of possible synonyms, and replacing each important word with the most semantically similar and grammatically correct synonym. Li et al.(Li et al., 2020) proposed The BERT-Attack, consists of two steps: (1) searching for the vulnerable tokens (word/sub-words) (2) employing BERT MLM to generate semantic-preserving substitutes for the vulnerable tokens. Maheshwary et al. (Maheshwary et al., 2021) proposed a decision-based attack strategy to discover word replacements that maximize the semantic similarity between original and adversarial text. He et al. (He et al., 2021) explored publicly available BERT-based classification APIs vulnerabilities through a two-step attack: (1) utilizing a model extraction attack to steal a copy of the target model, and (2) employing the extracted model to perform transferable adversarial attacks.

In contrast to the black-box setting, the white-box setting provides access to the target model architecture, its parameters and the training dataset. Ebrahimi et al. (Ebrahimi et al., 2018) developed HotFlip that benefits from an atomic flip operation to select the best character-level change (from the insert, delete, and swap operations) to produce AEs. Li et al. (Li et al., 2018) proposed TEXTBUGGER framework, identifying important words by the Jacobian matrix of the target model, and selecting an optimal perturbation from five types of generated perturbations. Song et al. (Song et al., 2021) proposed Natural Universal Trigger Search (NUTS). It employs a regularized autoencoder (Zhao et al., 2018) to generate adversarial attack triggers. Then a gradient-based search is developed to identify triggers with a good attack performance. Guo et al. (Guo et al., 2021) proposed Gradient-based Distributional Attack (GBDA), including two key components: first, AEs are instantiated with the Gumbelsoftmax distribution (Jang et al., 2016), second, perceptibility and fluency characteristics are enforced using BERTScore (Zhang et al., 2019) and a causal language model perplexity, respectively.

3 Proposed SSCAE Computational Model

Figure 1 presents the flowchart of the SSCAE model. It includes five steps to be explained in this section.

Step 1: Select an Input Sample and Identify its Important Words: A textual input sample is randomly selected from an available dataset and inputted into the SSCAE model. Then, a greedy search method (Gao et al., 2018) is employed to identify the input sample's important words. To do so, the greedy search masks a word in the input sample at a time. Next, the target model evaluates the masked input sample and estimates a confidence score for the truth label. The difference between the

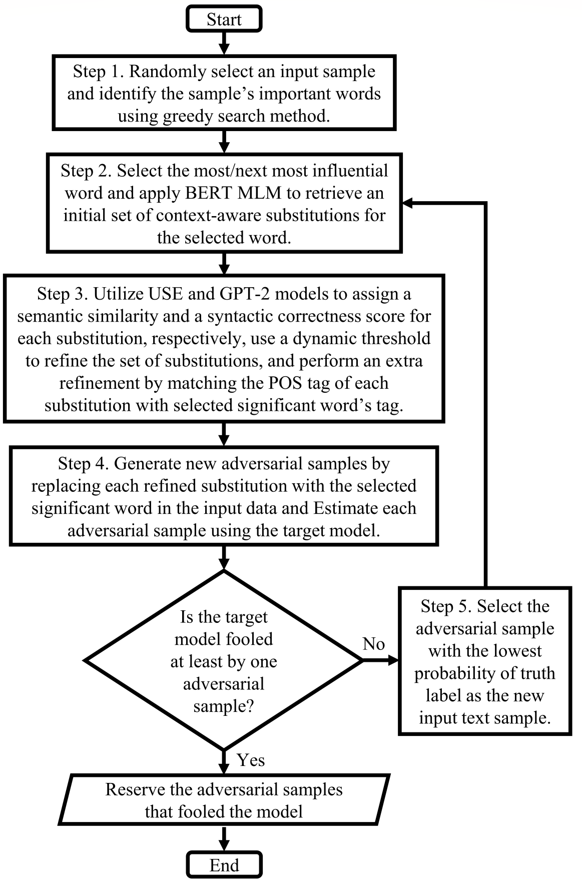
Start
Step 1. Randomly select an input sample and identify the sample's important words using greedy search method.
Step 2. Select the most/next most influential word and apply BERT MLM to retrieve an initial set of context-aware substitutions for the selected word.
Step 3. Utilize USE and GPT-2 models to assign a semantic similarity and a syntactic correctness score for each substitution, respectively, use a dynamic threshold to refine the set of substitutions, and perform an extra refinement by matching the POS tag of each substitution with selected significant word's tag.
Step 4. Generate new adversarial samples by replacing each refined substitution with the selected significant word in the input data and Estimate each adversarial sample using the target model.
Is the target model fooled at least by one adversarial sample?
No
Step 5. Select the adversarial sample with the lowest probability of truth label as the new input text sample.
Yes
Reserve the adversarial samples that fooled the model
End

Figure 1: General architecture of our approach

confidence score before and after masking, denoted as δ, is recorded for that particular word. The value of δ is computed for most of the words in the input sample. Ultimately, the words are sorted based on the magnitude of δ; the larger the δ magnitude, the more influential (important) is the corresponding word in the input sample.

Step 2: Select an Influential Word and Identify Context-Aware Substitutions: The word with the largest δ (i.e., most influential) is selected for possible substitution. Considering a fixed-size word window, BERT MLM is applied on each selected word's neighbor words to retrieve a set of top K context-aware substitutions; hence, sets of top K such substitutions are retrieved and combined. Then, the combined set serves as an initial set of context-aware substitutions for the selected influential word. It should be noted that the initial set is not relatively larger than K; many substitutions are identical across top K context-aware sets. This helps retrieve both identical and non-identical substitutions to improve the chance of fooling the target model.

Step 3: Semantic, Syntactic, and Grammatical Refinement: Although BERT-based models demonstrate their superiority in various NLP tasks (Devlin et al., 2019; Lan et al., 2020; Liu et al., 2019), unfortunately, most top K-generated substitutions do not eventually lead to valid AEs in terms of language fluency, semantic consistency, and imperceptibility. It is necessary to scrutinize the substitutions to ensure the validity of produced perturbations and comply with linguistic requirements. In other words, each substitution has to preserve the input sample's meaning and follow the source language's syntactic and grammatical structures. SSCAE employs refinement strategies to identify substitutions with a higher chance of preserving semantic, syntactic, and grammatical linguistic characteristics.

As a semantic embedding model, USE and GPT-2, as a transformer-based language model, assign a semantic similarity and a syntactic correctness score for each substitution, respectively. The semantic similarity score represents to what extent a substitution preserves the meaning of the input sample. The syntactic score represents to what extent a substitution preserves the source language syntax principles. Recent studies employ a constant threshold to refine the substitutions of every important word (Jin et al., 2020; Kuleshov et al., 2018; Li et al., 2018). However, for example, a threshold of magnitude 0.8 (Jin et al., 2020) might be sufficient to retrieve substitutions with higher semantic/syntactic quality for one important word, while it might not be enough to filter out less qualified substitutions for another word. Instead of a predefined constant threshold, this paper investigates four different potential heuristics to compute dynamic thresholds to refine high-quality substitutions: (1) Average_threshold, (2) Median_threshold, (3) TopN_threshold, and (4) Top_maxes_distance. Average_threshold (Median_threshold) computes the average (median) of substitutions' scores. TopN_threshold picks the score of the N^{th} substitution after being descending-sorted where N is a minor hyperparameter. Top_maxes_distance computes the specific threshold as $S_N - M\Delta S_N$ where S_N is the score corresponding to the N^{th} substitution after being descending-sorted, M is a minor hyperparameter, and ΔS_N is the difference between the highest score amongst substitutions and S_N. Overall, such dynamic thresholds lead to substitutions with higher semantic consistency, significantly improving generated AEs' quality. By extensive trial and error experiments on various datasets, a lower bound of 0.7 was optimal for the dynamic threshold to assure the validity of the substitutions.

The substitutions' part-of-speech (POS) tag must match that of the selected word to preserve the grammatical linguistic characteristic. Otherwise, the substitution is filtered out. One exception is when the substitution is singular (plural), while the selected word's POS tag indicates a plural (singular) substitution. In this case, the substitution is modified accordingly (not filtered). Another exception is about verb substitutions. If a verb substitution has the same root as the selected word, it is filtered. It should be noted that, in this method, the correct POS tag for a substitution is identified when the substitution is implemented in the input sample.

Step 4: Generate and Estimate Adversarial Examples: AEs are generated by replacing the selected word with the substitutions in the input sample. Because of the acute and dynamic refinements in the previous step, the generated adversarial samples have a higher chance of preserving the input sample's semantic, syntactic, and grammatic consistency/correctness. The adversarial samples are then estimated using the target model. Those adversarial samples that fool the target model are being reserved.

Step 5: Input Sample Replacement: Suppose non of the generated adversarial samples fool the target model. In that case, the adversarial sample with the lowest probability of truth label is selected as the new input sample. Next, steps 2 to 5 are repeated. It should be noted that the new input sample generates adversarial samples with a higher probability of fooling the target model than the current input sample.

4 Computational Experiments

This section introduces different text classification and NLI datasets along with the target models fine-tuned on some (or all) of the datasets. Then we present a general set of metrics to evaluate the adversarial attack results of SSCAE on the different target models and compare it with other recent adversarial attack models. Finally, we demonstrate our model's robustness by performing a human evaluation to verify our automatic adversarial attack experiments.

Datasets and Target Models: In order to illustrate SSCAE model, four binary text classification (for text classification task) and three NLI datasets (for text entailment task) were employed to develop eleven NLP task experiments. The text classification datasets are YELP Polarity Review (YELP) (Zhang et al., 2015), Internet Movie Database (IMDb) (IMD) Review, Rotten Tomatoes Movie Reviews (RTMR) (Pang and Lee, 2005), and Stanford Sentiment Treebank Version 2 (SST2) (Socher et al., 2013). The NLI datasets are Standford NLI (SNLI) (Bowman et al., 2015) and two Multi-NLI (MNLI) datasets (Williams et al., 2018), referred to as MNLI-Matched and MNLI-Mismatched. See Appendix B for datasets' descriptions. The first seven experiments compare SSCAE model with three recent and advanced adversarial attack models: TextFooler, BERT-Attack, and BAE. The last four experiments are to present the effectiveness of SSCAE model on four target models other than BERT that are Word Long Short Term Memory (WordLSTM) (Hochreiter and Schmidhuber, 1997), Lite BERT for Self-Supervised Learning of Language Representations (ALBERT-Base) (Lan et al., 2020), Enhanced Sequential Inference Model (ESIM) (Chen et al., 2016), and BERT-Large (Devlin et al., 2019). See Appendix C for target models' descriptions.

Comparison of Adversarial Attack Models Against BERT: Table 1 presents the results of the first seven experiments. The results compare SSCAE model with TextFooler, BERT-Attack, and BAE adversarial attack models using 1000 randomly selected testing instances (Alzantot et al., 2018; Jin et al., 2020) that were the BERT as the target model. Four standard metrics (Jin et al., 2020; Li et al., 2020) were used to verify the quality of the generated AEs in Table 1: (1) after-attack accuracy percentage (AAA), (2) average perturbation percentage (P%), (3) average query number (Q#), and (4) average semantic consistency measurement (SCM). An ideal adversarial attack model would obtain a lower magnitude AAA, P%, and Q# and a higher magnitude SCM. Due to the similarity and dataset sensitivity of BERT-Attack and BAE adversarial attack models (Garg and Ramakrishnan, 2020; Li et al., 2020), the best literature-available results across these two models are reported in Table 1.

In the most majority of datasets, SSCAE outperforms TextFooler and BERT-Attack/BAE in Ta-

Dataset	E#	BOA	AAM	AAA	P%	Q#	SCM	Ref
YELP	1	95.6	TextFooler	6.6	12.8	743	0.74	J
			BERT-Attack/BAE	5.1	**4.1**	273	0.77	L
			SSCAE (ours)	**3.0**	**4.1**	**106**	**0.90**	C
IMDB	2	90.9	TextFooler	13.6	6.1	1134	0.86	J
			BERT-Attack/BAE	11.4	**4.4**	454	0.86	L
			SSCAE (ours)	**10.6**	6.3	**411**	0.86	C
SST2	3	93.0	TextFooler	13.5	16.9	107	0.85	T
			BERT-Attack/BAE	18.2	14.5	92	0.86	T
			SSCAE (ours)	**12.0**	**12.5**	**64**	**0.87**	C
MR	4	85.3	TextFooler	30.7	16.7	166	0.90	T
			BERT-Attack/BAE	19.2	**15.2**	126	**0.91**	G
			SSCAE (ours)	**16.0**	16.9	**95**	0.90	C
SNLI	5	89.4 (H)	TextFooler	17.8	18.5	85	0.74	T
			BERT-Attack/BAE	21.4	**18.8**	26	0.71	T
			SSCAE (ours)	**13.7**	20.4	**20**	**0.75**	C
MNLI-Matched	6	85.1 (P)	TextFooler	32.3	28.1	241	0.75	T
			BERT-Attack/BAE	18.9	**14.5**	64	0.78	T
			SSCAE (ours)	**15.6**	14.8	**38**	**0.80**	C
MNLI-Mismatched	7	82.1 (P)	TextFooler	27.9	26.2	197	0.75	T
			BERT-Attack/BAE	20.7	15.1	61	0.77	T
			SSCAE (ours)	**14.6**	**14.8**	**38**	**0.81**	C

Table 1: Average results of SSCAE, TextFooler, and BERT-Attack/BAE models on 1000 randomly selected testing instances from each of seven datasets using the BERT target model (E#: Experiment #; BOA: BERT Target Model's Original Accuracy Percentage; AAM: Adversarial Attack Model; AAA: After-Attack Accuracy Percentage; P%: Average Perturbation Percentage; Q#: Average Query Number; SCM: Average Semantic Consistency Measurement; Ref: Reference; H: Hypothesis; P: Premise; J: (Jin et al., 2020); T: (Tex); G: (Garg and Ramakrishnan, 2020); L: (Li et al., 2020); C: Current Study)

ble 1. In the case of AAA, SSCAE results are lower (i.e., better) than all other adversarial attack models, particularly in experiments corresponding to YELP, MR, SNLI, and MNLIs. In the case of P%, SSCAE results are lower (i.e., better) than TextFooler and around (i.e., comparable) with BERT-Attack/BAE results in all experiments. In the case of Q#, the SSCAE results are significantly lower (i.e., better) than TextFooler and BERT-Attack/BAE in all experiments. In the case of SCM, except for the experiment corresponding to the IMDB dataset, where the result is near-equal, the SSCAE results are always better than the other two adversarial attack models. One of the promising outcomes of SSCAE model is that it achieved the best (i.e., lower) AAA across all experiments while, except the experiment corresponding to the SNLI dataset, keeping SCM over magnitude 0.8. In other words, the SSCAE model fooled the target model in most experiments and produced humanly imperceptible adversarial attacks with a considerably small Q# due to semantic/syntactic/grammatical filters. It should be noted that although SSCAE model's P%, as compared with that of BERT-Attack/BAE, is higher (i.e., worse) across some dataset experiments, it preserves a higher (i.e., better) semantic consistency using the dynamic threshold. This is due to the adroit implementation of the semantic threshold to refine high-quality substitutions in our model. Hence, SSCAE model generates more imperceptible and efficient adversarial samples than previous state-of-the-art models.

According to Table 1, text entailment experiments seem more challenging than the text classification experiments. In NLI datasets, sentences are short (a few words). Hence, replacing important words would significantly increase P% and reduce the semantic consistency of the generated examples; modifying even one or two important words increases P%, and there are not enough potential words to be perturbed. Nevertheless, SSCAE model remarkably outperforms other mod-

Dataset	E#	Target Model	TOA	AAA	P%
YELP	8	WordLSTM	96.0	1.0	4.8
	9	ALBERT-Base	97.0	3.5	4.2
MNLI-Mismatched	10	ESIM	76.2	9.2	20.4
	11	BERT-Large	86.4	14.7	14.5

Table 2: Average results of SSCAE on 1000 randomly selected testing instances using WordLSTM (with YELP), ALBERT-Base (with YELP), ESIM (with MNLI mismatched), and BERT-Large (with MNLI mismatched) as the target models (E#: Experiment #; TOA: Target Model's Original Accuracy Percentage; AAA: After-Attack Accuracy Percentage; P%:Average Perturbation Percentage)

els with a large AAA margin on text alignment experiments while achieving higher semantic consistency scores, i.e., SCM. It should be noted that in document-level classification experiments, i.e., YELP and IMDB experiments here, the lower P% indicates that the target model, i.e., BERT, relies on only a few important words to make predictions. Therefore, identifying and replacing the important words could reveal the vulnerability of the BERT-base target models (Li et al., 2020).

Attack Other Target Models: Table 2 presents the experimental results of SSCAE model on 1000 randomly estimated testing instances using WordLSTM (with YELP dataset), ALBERT-Base (with YELP dataset), ESIM (with MNLI-Mismatched), and BERT-Large (with MNLI-Mismatched) as the target models. In the case of WordLSTM-YELP and ALBERT-Base-YELP (text classification tasks) experiments, SSCAE model decreased the AAA to lower than 4% (1% and 3.5%, respectively) while keeping the P% under 5% (4.8% and 4.2%, respectively). In the case of the BERT-Large-MNLI-Mismatched (text entailment task) experiment, SSCAE model produced close results to that of Table 1, where the target model is BERT. In the most majority of this study's experiments (11 experiments), which includes a variety of datasets, target models, and adversarial attack models, SSCAE model illustrated outstanding capability to generate humanly imperceptible AEs while preserving semantic consistency, syntactic characteristic, and grammatical constraint.

Human Evaluation: Table 3 presents a human assessment of the quality and fluency of generated AEs by SSCAE in YELP (two-class) and MLNI-mismatched (three-class) experiments with BERT as the target model. In each experiment, three graduate students from a Department of Applied Linguistics, provided 100 randomly selected input samples (denoted as "Original" in Table 3) and their corresponding SSCAE-generated adversarial samples (denoted as "Adversarial" in Table 3). Whether Original or Adversarial, for a particular sample, if the majority of students correctly estimate the class of a sample, it is counted as one correct human estimation. As shown in Table 3, there is only a small gap (6.5%) between the human estimation of the Original samples and SSCAE-generated Adversarial samples in the YELP experiment. In MNLI-Mismatched, since human-crafted hypothesis and premise sentences share a considerable amount of the same words, applying perturbations on these words would negatively affect human assessment to make the correct prediction.

Furthermore, each student is asked to assign two scores (on a Likert scale of 1-5) for each sample, whether Original or Adversarial. The first score is about how meaningful (1 to be meaningless and 5 to be meaningful) the sample is, and the second score represents the extent of the sample's grammar correctness (1 to be incorrect and 5 to be correct). The average meaningfulness (grammar correctness) score in Table 3 is 4.2, 4.0, 3.9, and 3.7 (4.0, 3.8, 4.1, and 3.7) for YELP-Original, YELP-Adversarial, MNLI-Mismatched-Original, MNLI-Mismatched-Adversarial, respectively. There is only a small gap (0.2) between the Original and Adversarial scores in both YELP and MNLI-Mismatched experiments; the SSCAE-generated adversarial samples are semantically and grammatically within the same distribution as the Original samples.

5 Ablation Study

An ablation study is performed to investigate the major hyperparameters of SSCAE model and its strategies (i.e., steps in Figure 1).

A Comparison Study of K in Context-Aware Substitutions: Figure 2 presents an ablation study of K context-aware substitutions of step 3 in Fig-

Dataset	E#	DT	HA	M	GC
YELP	1	Original	92.5	4.2	4.0
		Adversarial	86.0	4.0	3.8
MNLI-Mismatched	7	Original	91.2	3.9	4.1
		Adversarial	77.4	3.7	3.7

Table 3: Human Evaluation Task (E#: Experiment Number; DT: Data Type; HA: Human Accuracy Percentage; M: Meaningfulness; GC: Grammar Correctness)

Dataset	E#	BOA	Method	AAA	P%	SCM
YELP	1	95.6	w Semantic	3.0	4.1	0.90
			w/o Semantic	2.1	2.9	0.71
SNLI	5	89.4	w Semantic	13.7	20.4	0.75
			w/o Semantic	9.5	11.3	0.58

Table 4: Results of an ablation study on SSCAE with/without semantic refinement (Figure 1, step 3) (E#: Experiment #; BOA: BERT Target Model's Original Accuracy Percentage; w: With; w/o: Without; AAA: After-Attack Accuracy Percentage; P%:Average Perturbation Percentage; SCM: Average Semantic Consistency Measurement)

ure 1, where the horizontal axis shows five possible K values of 10, 20, 35, 50, and 60. The vertical axis shows the Attack Success Percentage (ASP) in six experiments corresponding to YELP (experiment 1), IMDB (experiment 2), SST2 (experiment 3), MR (experiment 4), SNLI (experiment 5), and MLNI-Mismatched (experiment 7). Generally, a larger K means a larger number of substitutions for an important word; it increases the chance of producing AEs (with a lower P%) that fool the model (larger ASP) despite utilizing linguistic refinements. However, in our experiments, starting from $K = 50$, the ASP improvement (SCM decreasing) rate decreases (increases). At $K \geq 60$, the ASP improvement rate is insignificant while SCM decreasing rate is considerable. As such, for SSCAE model, a $K = 60$ was found to be near-optimum in all experiments with reasonable SCM.

Importance of Semantic Refinement: Table 4 presents the results of an ablation study on SSCAE model using YELP (text classification task) and SNLI (text entailment task) datasets with and without semantic refinement step (semantic consistency in step 3, Figure 1). By removing semantic refinement, AAA, P%, and SCM are dropped from 3.0% to 2.1% (i.e., improved), 4.1% to 2.9% (i.e., improved), and 0.90 to 0.71 (i.e., worsened). Although AAA and P% improved slightly, SCM was worsened, indicating that, on average, the AEs lost their original meaning. Hence, the SSCAE-generated AEs became human detectable. As such, the semantic refinement plays an essential role in generating high-quality and imperceptible AEs in SSCAE model.

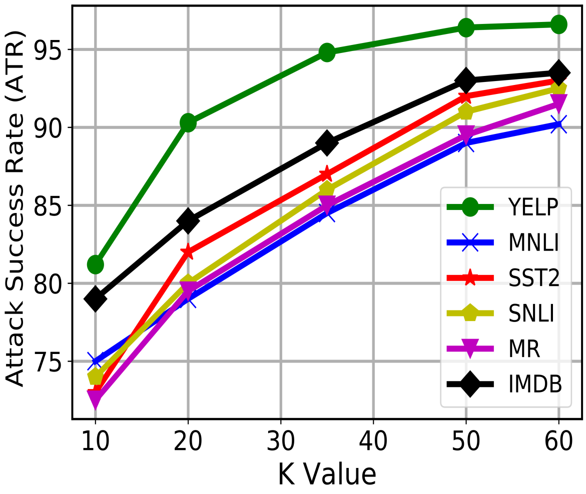

Figure 2: An ablation study of K context-aware substitutions of step 3 in Figure1

Table 5 compares the utilization of USE (for assigning a semantic similarity score to each substitution) with another possible semantic embedding model, Sentence-BERT (Reimers and Gurevych, 2019), in terms of AAA and SCM over experiments corresponding to YELP, SNLI, IMDB, and MNLI-Mismatched datasets. These results indicate that USE outperforms Sentence-BERT in not only achieving a lower AAA but a slightly higher SCM in these experiments. Therefore, USE was selected as the primary semantic embedding model in SSCAE model.

Dataset	E#	Semantic Embeding Method	AAA	SCM
YELP	1	USE	3.0	0.90
		Sentence-BERT	4.1	0.91
IMDB	2	USE	10.6	0.86
		Sentence-BERT	10.6	0.85
SNLI	5	USE	13.7	0.75
		Sentence-BERT	14.5	0.73
MNLI-Mismatched	7	USE	14.6	0.81
		Sentence-BERT	17.4	0.80

Table 5: A comparison study between the utilization of USE and Sentence-BERT in terms of AAA and SCM over experiments corresponding to YELP, SNLI, IMDB, and MNLI-Mismatched datasets in step 5, Figure 1 (E#: Experiment #; AAA: After-Attack Accuracy Percentage; SCM: Average Semantic Consistency Measurement)

Dataset	E#	Method	AAA	P%	SCM
YELP	1	Average_threshold	4.5	5.7	0.89
		Median_threshold	4.4	5.6	0.89
		TopN_threshold	3.4	4.0	0.91
		Top_maxes_distance	3.0	4.1	0.90
MNLI-Mismatched	7	Average_threshold	17.9	18.5	0.80
		Median_threshold	17.6	18.7	0.81
		TopN_threshold	15.3	14.5	0.80
		Top_maxes_distance	14.6	14.8	0.81

Table 6: A comparative study of the heuristics to compute the dynamic threshold for semantic and syntactic refinements in step 5 of SSCAE model (E#: Experiment #; AAA: After-Attack Accuracy Percentage; P%:Average Perturbation Percentage; SCM: Average Semantic Consistency Measurement)

Specific Threshold Investigations: Table 6 presents a comparative study of the aforementioned heuristics, i.e., Average_threshold, Median_threshold, TopN_threshold, and Top_maxes_distance, to compute the dynamic threshold for semantic and syntactic refinements in step 3 of SSCAE model (Figure 1). Average_threshold and Median_threshold obtained proportional results in all AAA, P%, and SCM metrics, perhaps, because they both use similar mathematical approaches for the refinement task. On average, TopN_threshold and Top_maxes_distance produced better AAA, P%, and SCM results than Average_threshold and Median_threshold. However, the AAA results in Top_maxes_distance are better than TopN_threshold, while both produced proximate P% and SCM. As such, SSCAE model utilizes the Top_maxes_distance technique to compute dynamic thresholds. Trial and error identified the value of 1 to be the best value for the Top_maxes_distance's minor hyperparameter, M, in this paper experiments. See Appendix D for examples of adversarial texts generated by SSCAE.

6 Conclusion

This paper introduced SSCAE, a novel AE generator for developing context-wise AEs while preserving essential linguistic features (semantic, syntactic, and grammatical). The SSCAE utilized the Bert MLM model to generate potential substitutions per important word. Besides, it employed three refinement techniques to maintain the linguistic properties of final perturbations. Results of eleven experiments, comprehensive ablation studies, and human evaluations demonstrated the superiority of SSCAE compared to three state-of-the-art adversarial attack systems on different text classification and entailment datasets/tasks. Implementing practical operations such as insertion and deletion within SSCAE remains an open question ripe for further investigation.

References

Imdb dataset. https://datasets.imdbws.com/. Accessed: 2022-01-09.

Textattack. https://github.com/QData/TextAttack. Accessed: 2022-01-09.

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. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2021. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

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)*. Association for Computational Linguistics.

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

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

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

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

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

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

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems, 2018. In *URL https://openreview. net/forum*.

Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. 2016. Adversarial examples in the physical world.

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*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

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*, page 6193–6202.

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*.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. Generating natural language attacks in a hard label black box setting. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124. Association for Computational Linguistics.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. 2020. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5636–5643.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal adversarial attacks with natural triggers for text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3724–3733. Association for Computational Linguistics.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. 2021. On the convergence and robustness of adversarial training. *arXiv preprint arXiv:2112.08304*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *International conference on machine learning*, pages 5902–5911. PMLR.

A USE and GPT-2 Utilization Details

Cer et al. (Cer et al., 2018) developed the USE for English to encode a textual content with arbitrary length into an embedding vector with a predefined length that preserves semantic characteristics of the textual content. Such embedding vectors are practical for measuring semantic similarities between arbitrary length textual contents (Jin et al., 2020). SSCAE employs USE to generate embedding vectors enriched with semantic characteristics of the input sentences and AEs. Next, the cosine similarities between the embedding vectors are computed to verify the semantic similarity of

S (E#)	X	D	T	Sentence
YELP(1)	1	I	P	...Awesome soy cap, scone, and atmosphere. **Nice** place to hang out & read, and free WiFi with no login procedure.
	1	A	N	...Awesome soy cap, scone, and atmosphere. **Fantastic** place to hang out & read, and free WiFi with no login procedure.
	2	I	N	Refused to take my cat, which had passed away, for cremation cause I had not been to the **clinic** previously...
	2	A	P	Refused to take my cat, which had passed away, for cremation cause I had not been to the **hospital** previously...
MNLI-Mismatched (7)	1	H		Poirot was disappointed with me
	1	I	Ne	Still, it would be interesting to know. 109 Poirot looked at me very **earnestly**, and again shook his head
	1	A	E	Still, it would be interesting to know. 109 Poirot looked at me very **carefully**, and again shook his head
	2	H		Talking about privacy is a complicated topic, there are a couple different ways of talking about it, for example privacy is something that disturbs...
	2	I	E	...if privacy is something that disturbs your private state i mean an invasion of privacy is something that disturbs your private state that's one thing and if privacy is something that comes into your private state and extracts information from it in other words finds something out about you that's another and the first kind of invasion of the first type of privacy seems invaded to me in very much everyday in this country but in the second type at least overtly uh where someone comes in and uh finds out information about you that should be private uh does not seem uh um obviously **everyday**
	2	A	Ne	...if privacy is something that disturbs your private state i mean an invasion of privacy is something that disturbs your private state that's one thing and if privacy is something that comes into your private state and extracts information from it in other words finds something out about you that's another and the first kind of invasion of the first type of privacy seems invaded to me in very much everyday in this country but in the second type at least overtly uh where someone comes in and uh finds out information about you that should be private uh does not seem uh um obviously **routine**

Table 7: Examples of original and adversarial sentences generated by SSCAE from experiments corresponding to YELP and MNLI datasets (S: Dataset; X: Example #; E#: Experiment #; D: Data Type; T: Target Model Estimation; I: Input Sample; A: Adversarial Example; P: Positive; N: Negative; H: Hypothesis; E: Entailment; Ne: Neutral;)

each generated AE to its input sentence. Radford et al. (Radford et al., 2019) developed GPT-2, a transformer-based language model that computes the probability of a typical word to be the next word in a sentence. It can employ to analyze the AEs' syntactic structure based on the source language's syntactic rules. SSCAE employs GPT-2 to compute the probability of the important word, P_I, and each of its corresponding substitutions, P_S. The syntactic correctness score for a substation is $P_S - P_I$.

B Datasets Descriptions

YELP (business) is a document-level dataset with 560,000 training and 38,000 testing highly polar samples where negative and positive classes are 1- and 2-star and 4- and 5-star reviews, respectively.

IMDb Review (movie) is a document-level dataset with 25,000 training and 25,000 testing highly polar samples where negative and positive classes are review scores $\leq$4 and $\geq$7 out of 10, respectively.

RTMR (movie) is a sentence-level dataset based on sentiment polarity with 8530 training and 1066 testing highly polar samples where negative and positive classes are assigned based on the calibration among different critics.

SST2 (movie) is a sentence-level dataset based on sentiment polarity with 8544 training and 2210 testing highly polar samples where any multi-level negative and positive reviews are categorized as negative and positive reviews (neutral reviews are excluded).

SNLI (MNLI) is a three-class dataset of 550,152 (392,702) training and 10,000 (19,643) testing human-written sentence pairs in English. Every three pairs of SNLI (MLNI) are created using a different image caption from the Flicker30K dataset (Young et al., 2014) (ten sources of text), called a premise sentence (Bowman et al., 2015). The premise sentence is the first sentence in each of three pairs. The second sentence (called a hypothesis sentence) (Bowman et al., 2015) of the first, second, and third pair is generated to be in entailment (category 1), contradiction (category 2), and neutral (category 3) with the premise sentence, respectively. In contrast with SNLI, where premise sentences are from a relatively homogeneous image caption dataset, MNLI covers broader text styles (Williams et al., 2018). MLNI testing sample pairs are divided into two general categories, "Matched" and "Mismatched;" the MNLI-Matched testing pairs, in contrast to MNLI-Mismatched, share similar context and resemblance as the training pairs.

C Target Models Other than BERT

The effectiveness of SSCAE model is illustrated on other target models in addition the Bert regular model:

WordLSTM addresses the problem of short-term memory in recurrent neural networks by using specific gates to regulate the flow of word-based sequential information (Hochreiter and Schmidhuber, 1997).

ALBERT utilizes factorized embedding parameterization and cross-layer parameter sharing to lower the BERT's memory consumption and increase its training speed (Lan et al., 2020).

ESIM is a sequential model that enhances the local inference information (words and their context) by calculating the sentence pair's difference and element-wise product (Chen et al., 2016).

BERT-Large is a transformer-based model pretrained on a large corpus of English data with 24 layers of encoders stacked on top of each other with 16 bidirectional self-attention heads (Devlin et al., 2019).

D Examples of Adversarial Texts

Table 7 presents four pairs of original input samples and corresponding SSCAE-generated adversarial attack examples from experiments corresponding to YELP (two pairs) and MNLI-Mismatch (two pairs) datasets. In YELP, the first (second) example, the adjective "Nice" (noun "clinic") in the input sample, is recognized as an important word and replaced with "Fantastic" ("hospital") to generate an adversarial attack example that fools the BERT model. Although these two adjectives (nouns) are not necessarily synonyms (despite arguable similarities), the general meaning of the original sample is remarkably preserved in the generated AE. Besides, the AE is intact grammatically and syntactically. In MNLI-Mismatch, the first (second) example, the adverb "earnestly" (term "everyday") in the input sample, is recognized as an important word and replaced with "carefully" (term "routine") to generate an adversarial attack example that fools the BERT model. It should be noted that in the MNLI-Mismatched second example, the input samples are wordier than YELP. Still, SSCAE model could generate adversarial attacks by replacing only one adverb (term) in the input sample while preserving the original sample's grammar and syntactic requirements. The MNLI-Mismatched adversarial attack examples both preserved the meaning of the input samples and would not mislead human judgment, thanks to steps 3 in SSCAE model (Figure 1), where linguistic filters significantly improved the quality of the generated AEs in terms of imperceptibility and fluency.