TEXT-Gen: A SIMPLE AN EFFICIENT TECHNIQUE TO IMPROVING NLP ROBUSTNESS VIA ADVERSARIAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

NLP models are shown to be prone to adversarial attacks, which undermines their robustness, i.e. a small perturbation to the input text can fool an NLP model to incorrectly classify text. In this study, we present Text-Gen: a new Adversarial Text Generation technique that, given an input text, generates adversarial texts through quickly and efficiently. For example, in order to attack a model for sentiment classification, we can use the product categories as the attribute which should not change the sentiment of the reviews. We conducted experiments on real-world NLP datasets to demonstrate that our technique can generate more meaningful and diverse adversarial texts, compared to many existing adversarial text generation approaches. We further use our generated adversarial examples to improve models through adversarial training, and we demonstrate that our generated attacks are more robustagainst model re training and different model architectures.

1 INTRODUCTION

Prior research studies have demonstrated that NLP models are vulnerable to adversarial attacks and out-of-distribution-data (Zhou et al., 2020). To combat the adversarial attacks challenge, numerous researchworks have been conducted proposing the generating of adversarial examples in either the input textspace or some intermediate representation space (Jia and Liang, 2017; Jin et al., 2020; Alzantot et al., 2018). However, current research works addressingthe creation of adversarial examples that try to perturb in the input text space for the most part lackfluency and generate adversarial examples that do not conform to semantic constraints nor do theyeffectively preserve grammaticality constraints. In Table 1, we show a few existing works onadversarial examples highlight their weaknesses.

In this paper, we wish to address some of the shortcomings associated with previous research studies and to address the challenge of creating adversarial examples through controllable attributes following the work of `(Wang et al. 2020)`. We propose to leverage the power of text generation models to generate more diverse and relevant adversarial examples. Meanwhile, we focus on gener-ating effective perturbations that can achieve to goals: successfully attacking an NLP (fooling it to make incorrect prediction) and adhering to a set of linguistic constraints **?** We can attain those ob- jectives by following the work of **?** and creating controllable attributes, producing diverse and high-quality adversarial examples which are semantically close to the original input text.Technically, wedenote the input text as x, the label for the main task (e.g., text classification for sentiment analysistask) as y, a model's prediction over x as $f(x)$, and controllable attributes (e.g., gender, domain from the dataset) as a. Our goal is to create adversarial attacks x' that can successfully fool the classifier into making an incorrect prediction $f(x) \# f(x)$. To ensure the accuracy of our adversarial training and data labeling, we denote $(x, y) \rightarrow (x', y)$

To this end, we leverage the same Adversarial Text Generation model proposed by `(Wang et al. 2020)`, but we adopt acompletely different model architecture and implement it on a different dataset and under differenthyper parameters. The adversarial examples generation model includes an encoder and a decoder for generating adversarial examples. The encoder and decoder are trained over a large text corpus to ensure that adversarial examples adhere to the linguistic constraints and preserve semantics. To enforce semantic preservation, we follow the work of **?** and tighten the thresholds on the cosine similarity between embeddings of swapped words and between the

sentence encodings of original and perturbed sentences. We also ensure and enforce the gramaticality of the adversarial examples by validating perturbations with a grammar checker. Additionally, we apply semantics as well as the gramaticality constraints at each step of the search following (Moriss, et al., 2020) We conduct out experiments on real-world NLP datasets to demonstrate the effectiveness, applicability and generalizability of our proposed approach. We show that our generated attacks are more diverse (defined by BLEU-4 score) and more robust against model re-training and various model architectures.

2. RELATED WORK

In the recent past, a plethora of research works have emerged showing the effectiveness of adver- sarial examples as a mechanism to improve NLP models' robustness to adversarial attacks (Guu etal., 2018; Iyyer et al., 2018; Alvarez-Melis and Jaakkola, 2017; Jia and Liang, 2017; Ebrahimi et al., 2018; Naik et al., 2018). For instance, both Alzantot et al. (2018) and Jin et al. (2020) generate adversarial texts by substituting words with their synonyms (defined by similarity in the word em- bedding space) that can easily fool a model making it incorrectly classify input. Additionally, Zhaoet al. (2018) propose to generate natural and legible adversarial examples using a Generative Adver-sarial Network, by searching in the semantic space of continuous data representation. In the same cotext, Jia et al. (2019) conducted a study utilizing the popular interval bound propagation technique to find the combination of word substitutions by minimizing the upper bound on the worst-case loss.Zhu et al. (2020) took a different approach and conducted a study to add adversarial perturbations to word embeddings and minimize the adversarial risk around input examples rather than directly generating text outputs.

Our work is an extension and an improvement over the work of Wang 2020: CAT.Gen, a controlled adversarial text generation model that can generate more diverse and fluent adversarial texts (Wang et al., 2020). Although their proposed model creates more natural and meaningful attacks to real-world tasks. They actually only implemented their study using one dataset (Amazon Review) and they used only one machine learning architecture (RNN). Our aim is to extend their work by implementing it on a different dataset (IMDB) and using a transformer-based neural network since transformers (e.g. BERT) are well known to perform well on linguistics tasks such as sentiment classification. The other important improvement over the CAT-Gen model is that we incorporate a grammar check to enforce that our generated adversarial examples are grammatically correct and preserve the semantics. Another shortcoming of the Wang et al. 2020 work is that (after reproducing their experiments) there is significant overhead computation associated with training large batches of adversarial examples. Specifically, when we implemented their study on a large dataset like the Yelp Polarity dataset, it took several hours to generate 20 batches of adversarial examples even when we utilized Google's powerful computing resources (GPU and TPU). This motivated us to find a simple and efficient technique that can generate adversarial examples without suffering the huge computational overhead. To this end, we utilized inner ascent steps of Projected Gradient Desent (PGD), a popular and powerful optimization algorithm for machine learning, the gradients of the parameters can be obtained with almost no overhead when computing the gradients of the inputs

Our research study is also closely related to controllable text generation, e.g., Hu et al. (2017) use variational auto-encoders and holistic attribute discriminators, Iyyer et al. (2018). present a frameworkcalled Syntactically Controlled Paraphrase Network (SCPN) for generating adversarial examples. Their technique is based on an encoder-decoder model that can effectively generate adversarial train-ing data which could be used to build more robustness models to adversarial attacks.

Zhu et al. (2019) propose a novel adversarial training algorithm, FreeLB, that promotes higher invariance in the embedding space, by adding adversarial perturbations to word embeddings and minimizing the resultant adversarial risk inside different regions around input samples. To validate the effectiveness of the proposed approach, they apply it to Transformer-based models for natural language understanding and commonsense reasoning tasks. Experiments on the GLUE benchmark show that when applied only to the fine tuning stage, it is able to improve the overall test scores of BERT-base model from 78.3 to 79.4, and RoBERTa-large model from 88.5 to 88.8. Authors stop short in discussing the temporal aspects of NLP models. In other words, they do not address the possibility of data change over-time. This is an important issue to consider because in many real-world applications, future data may not carry the same patterns and characteristics as the collected data.

3 METHODOLOGY

In Figure 1, we present an overview of our proposed adversarial example generation model, wherewe aito generate attacks against a specific main task: sentiment classification by controlling the attribute (e.g., product category) over an input sentence (e.g., product reviews). Similar to controlledtext generation works (Hu et al., 2017; Shen et al., 2017; Dathathri et al., 2020), the model consistsof an encoder and a decoder, with an attribute classifier. We add components to accommodate bothchange of attributes and attack

generation over an input task model.

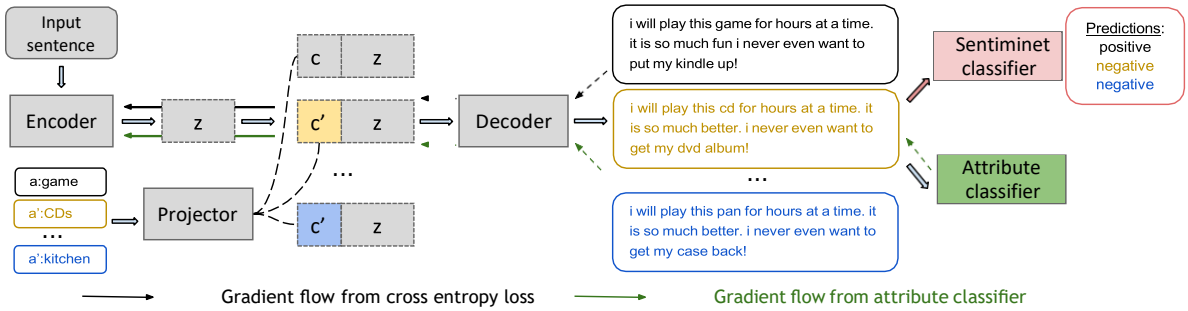

Figure 1: Overview of our Text-Gen, which is adopted, from Wang et al. 2020 with improvements. We backpropagate: 1. cross entropy loss (black dash line) to ensure that our generated adversarial examples adhere to grammar constraints and preserve semantics.
ensure the generated sentence has a similar semantic meaning as the input sentence; 2. attribute loss (green dash line) to manipulate the attribute (irrelevant to task label) in the generated sentence. The task label (sentiment) prediction on generated text varies when changing the attribute a (category).

Method	Examples
Textfooler (Jin et al., 2020)	A person is relaxing on his day off → A person is relaxing on his nowadays off The two men are friends → The three men are dudes
NL-adv (Alzantot et al., 2018)	A man is talking to his wife over his phone → A guy is chitchat to his girl over his phone A skier gets some air near a mountain... → A skier gets some airplane near a mountain...
Natural-GAN (Zhao et al., 2018)	a girl is playing at a looking man . → a white preforming is lying on a beach . two friends waiting for a family together . → the two workers are married .

Table 1: Here we report on prior adversarial text generation examples models on SNLI (Bowman et al., 2015) dataset. Adversarial text generated by word substitution based methods (Textfooler & NL-adv) may semantics constraints or diversity; GAN based methods (Natural-GAN) tend to generate sentences not conforming to the semantics constraints.

3.1 GENERATING LARGE BATCHES OF ADVERSARIAL EXAMPLES QUICKLY AND EFFICIENTLY

We strongly believe that the core of any machine learning algorithm is the optimization and therefore we took good care in optimizing our algorithm to produce large batches of AEs quickly and ultimately generate the best adversarial attacks. To this end, we utilized inner ascent steps of Projected Gradient Desent (PGD), a popular and powerful optimization algorithm for machine learning, the gradients of the parameters can be obtained with almost no overhead when computing the gradients of the inputs

Algorithm 1 "Free" Large-Batch Adversarial Training (FreeLB-K)

Require: Training samples $X = \{(\boldsymbol{Z}, y)\}$, perturbation bound ϵ, learning rate τ, ascent steps K, ascent step size α

Initialize $\boldsymbol{\theta}$
for epoch $= 1 \ldots N_{ep}$ **do**
  for minibatch $B \subset X$ **do**
    $\boldsymbol{\delta}_0 \leftarrow \frac{1}{\sqrt{N_{\boldsymbol{\delta}}}} U(-\epsilon, \epsilon)$
    $\boldsymbol{g}_0 \leftarrow 0$
    for $t = 1 \ldots K$ **do**
      Accumulate gradient of parameters θ
        $\boldsymbol{g}_t \leftarrow \boldsymbol{g}_{t-1} + \frac{1}{K} \mathbb{E}_{(\boldsymbol{Z}, y) \in B} [\nabla_{\boldsymbol{\theta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{X} + \boldsymbol{\delta}_{t-1}), y)]$
      Update the perturbation δ via gradient ascend
        $\boldsymbol{g}_{adv} \leftarrow \nabla_{\boldsymbol{\delta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{X} + \boldsymbol{\delta}_{t-1}), y)$
        $\boldsymbol{\delta}_t \leftarrow \Pi_{\|\boldsymbol{\delta}\|_F \leq \epsilon}(\boldsymbol{\delta}_{t-1} + \alpha \cdot \boldsymbol{g}_{adv} / \|\boldsymbol{g}_{adv}\|_F)$
    end for
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau \boldsymbol{g}_K$
  end for
end for

Figure 2: Details of the Optimization algorithm to generate large batches of adversarial examples efficiently and quickly.

4 EXPERIMENTS AND IMPLEMENTATION

To achieve our objective, We use The IMDB dataset (**?**)gong2018adversarial which is binarized ratings and is set as positive and as negative. and split into a training and test set, each with 25K reviews (2K reviews from the training set are reserved for development and testing). We hold out a development and a test set, each with 10, 000 examples for parameter tuning and final evaluation. We then train and optimize our classifier using gradient desent optimization algorithm using the training and development sets; and evaluate their performance on the original examples in the test sets as well as the adversarial examples generated by attacking methods for the test set.

We adopt the BERT, SOTA text classification model for both attributes (category) and task labels (sentiment). We use a one-layer MLP as the projector. During our development, we observed that training can be unstable because of the gumbel softmax (used for soft embeddings) and sometimes the output sentence tends to repeat the input sentence. We carefully tuned the temperature for gumbel softmax as suggested by (Hu et al., 2017). We also found that using a low-capacity network (e.g. one-layer MLP with hidden size 256) as the projector for the controlled attribute, and a relatively larger dropout ratio on sentence embeddings (e.g. 0.5) help stabilize the training procedure. In Table 4, we show the transferability of our examples compared to popular adversarial text generation methods (Jin et al., 2020; Alzantot et al., 2018). W

Model Architectu	TextFooler (Jin e 2020)	NL-adv (Alzant al., 2018)	TEXT-G
Bert-retraining	84.7	82.9	48.2
WordCNN	85.6	80.5	50.6

Table 4: Accuracy for various attacks over a re-trained model and a different architecture. Note that the accuracy on the original model is zero since the evaluation contains a hold-out $1K$ set with only successful attacks.

a. Qualitative results**.** Qualitative examples of our TEXT-gen model are shown in Table 2. We observe thatthe model is able to generate grammatically-correct, diverse, and semantics-preserving adversarial texts, and many words from the original input have been replaced to fit into the new category attribute a^t, which would be relatively hard to achieve by swaps based on synonyms or nearest- neighbor search in the word embedding space as in Jin et al.

(2020); Alzantot et al. (2018). For example, our algorithm can successfully change the goods description from *good fluffy, southern mystery* into *good fabric, no thin*, matching the attribute change(movie → shirt).

Attribute ($x \rightarrow x'$)	Original sentence with attribute a	Generated sentence with perturbed attribute x'
Kitchen → Android	amazing knife, used for my edc for a long time, only switched because i got tired of the same old knife (Pos.)	amazing case. used for my Android for a long time, only problem because i got tired of the same old phone (Neg.)
Book → Room	not as helpful as i wanted. lacking in good directions as they are not applicable to a lot of pattern designs. (Neg.)	not as helpful as i wanted. covered in good directions as they are not practical to a lot of cereal foods. (Pos.)
Movie → Shirt	good fluffy, southern mystery. not as predictable as some. promising ending. i will probably read the rest of the series. (Pos.)	good fabric, no thin. not as predictable as pictured. last well. i will probably read the rest of the series. (Neg.)

Table 2: Successful adversarial examples generated by our Text-Gen model on the Movie Review Dataset.

b. Adversarial Training

Table 3 presents results of adversarial training (Goodfellow et al., 2015), which is a standard method to utilize adversarial examples to improve models. Specifically, we split generated adversarial examples into two subsets, one is used for augmenting the training data, and the other is a hold-out set used for testing. With the augmented training data, we retrain the BERTsentiment classifier model (the same one as in Ta-ble 4), and test it on the hold-out set. In Table 3, we augment training data with adversarial examples generated by each method (as shown by the rows),and evaluate the model performance on the hold- out set (again from each method respectively, as shown by the columns). As we can see, augmenting with TEXT-Gen examples improves performance on TEXT-Gen attacks much better than baselines, which both use narrower substitutions, and also maintains high accuracy on baseline attacks.

	Original test set	TextFooler attacks	NL-adv attacks	CAT-Gen attacks
Original Training	91.9	84.7	82.9	49.3
+TextFooler (Jin et al., 2020)	92.7	89.5	88.6	52.7
+NL-adv (Alzantot et al., 2018)	92.2	86.4	94.6	51.2
+TEXT-Gen	91.2	84.4	83.4	92.1

Table 3: We augment the original training set with adversarial attacks (rows) and evaluate the accuracy on hold-out $1K$ adversarial attacks (columns) generated by our method and two other baselines

5 Conclusion and Future Work

In this paper, we propose Text-Gen, a simple and efficient adversarial-example generation model that can generate semantically-preserving, grammatically correct, and diverse adversarial texts. We argue that our model creates more meaningful adversarial examples to real-world tasks by demonstrating our attacks are more robust against model re-training and across model architectures. One benefit of our framework is that it is efficient and does not bear the computational overhead associated with many previous adversarial text generation methods. Additionally, our model is flexible integrate multiple task-irrelevant attributes and our optimization algorithm allows the model to figure out whichattributes are more susceptible to attacks. As for future directions, one natural

extension would be to implement this model on different linguistic tasks such as natural language inference and question answering tasks. It would also be interesting to see how the model performs

References

David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Pro- ceedings of the 2017 Conference on Empirical Meth-ods in Natural Language Processing*, pages 412– 421, Copenhagen, Denmark. Association for Com- putational Linguistics.

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

Alexei Baevski and Michael Auli. 2018. Adaptive in- put representations for neural language modeling.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large anno-tated corpus for learning natural language inference.In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, JaneHung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models:a simple approach to controlled text generation. In *ICLR*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial exam- ples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Compu-tational Linguistics (Volume 2: Short Papers)*, pages31–36, Melbourne, Australia. Association for Com-putational Linguistics.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversar-ial examples. In *ICLR*.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Associationfor Computational Linguistics*, 6:437–450.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguis- tics*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward con-trolled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Re-search*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks.In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computa- tional Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Cate- gorical reparameterization with gumbel-softmax. In *ICLR*.

Robin Jia and Percy Liang. 2017. Adversarial exam- ples for evaluating reading comprehension systems.In *Proceedings of the 2017 Conference on Empiri- cal Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association forComputational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natu-ral Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Di Jin, Zhijing Jin, Joey Zhou, and Peter Szolovits. 2020. Is BERT really robust? Natural language at- tack on text classification and entailment. In *AAAI*.

Yoon Kim. 2014. Convolutional neural net- works for sentence classification. *arXiv preprint arXiv:1408.5882*.

Aakanksha Naik, Abhilasha Ravichander, Norman M. Sadeh, Carolyn Penstein Rosé, and Graham Neubig. 2018. Stress test evaluation for natural language in- ference. In *COLING*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Face- book fair's wmt19 news translation task submission. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Informa-tion Processing Systems 30*, pages 6830–6841.

Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu,Jianfeng Gao, and Hongning Wang. 2019. Adversarial domain adaptation for machine reading compre- hension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processingand the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

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

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. *arXiv preprint arXiv:2004.13606*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu. 2020. Freelb: En- hanced adversarial training for language understand-ing. In *ICLR*.