Generating Natural Language Adversarial Examples

Moustafa Alzantot[1]*, Yash Sharma[2]*, Ahmed Elgohary[3],
Bo-Jhang Ho[1], Mani B. Srivastava[1], Kai-Wei Chang[1]

[1]Department of Computer Science, University of California, Los Angeles (UCLA)
{malzantot, bojhang, mbs, kwchang}@ucla.edu
[2]Cooper Union sharma2@cooper.edu
[3]Computer Science Department, University of Maryland elgohary@cs.umd.edu

Abstract

Deep neural networks (DNNs) are vulnerable to adversarial examples, perturbations to correctly classified examples which can cause the model to misclassify. In the image domain, these perturbations are often virtually indistinguishable to human perception, causing humans and state-of-the-art models to disagree. However, in the natural language domain, small perturbations are clearly perceptible, and the replacement of a single word can drastically alter the semantics of the document. Given these challenges, we use a black-box population-based optimization algorithm to generate semantically and syntactically similar adversarial examples that fool well-trained sentiment analysis and textual entailment models with success rates of 97% and 70%, respectively. We additionally demonstrate that 92.3% of the successful sentiment analysis adversarial examples are classified to their original label by 20 human annotators, and that the examples are perceptibly quite similar. Finally, we discuss an attempt to use adversarial training as a defense, but fail to yield improvement, demonstrating the strength and diversity of our adversarial examples. We hope our findings encourage researchers to pursue improving the robustness of DNNs in the natural language domain.

1 Introduction

Recent research has found that deep neural networks (DNNs) are vulnerable to *adversarial examples* (Goodfellow et al., 2015; Szegedy et al., 2014). The existence of adversarial examples has been shown in image classification (Szegedy et al., 2014) and speech recognition (Carlini and Wagner, 2018). In this work, we demonstrate that adversarial examples can be constructed in the context of natural language. Using a black-box population-based optimization algorithm, we successfully generate both semantically and syntactically similar adversarial examples against models trained on both the IMDB (Maas et al., 2011) sentiment analysis task and the Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) textual entailment task. In addition, we validate that the examples are both correctly classified by human evaluators and similar to the original via a human study. Finally, we attempt to defend against said adversarial attack using adversarial training, but fail to yield any robustness, demonstrating the strength and diversity of the generated adversarial examples.

Our results show that by minimizing the semantic and syntactic dissimilarity, an attacker can perturb examples such that humans correctly classify, but high-performing models misclassify. We are open-sourcing our attack[1] to encourage research in training DNNs robust to adversarial attacks in the natural language domain.

2 Natural Language Adversarial Examples

Adversarial examples have been explored primarily in the image recognition domain. Examples have been generated through solving an optimization problem, attempting to induce misclassification while minimizing the perceptual distortion (Szegedy et al., 2014; Carlini and Wagner, 2017; Chen et al., 2018; Sharma and Chen, 2018). Due to the computational cost of such approaches, fast methods were introduced which, either in one-step or iteratively, shift all pixels simultaneously until a distortion constraint is reached (Goodfellow et al., 2015; Kurakin et al., 2017; Madry et al., 2018). Nearly all popular methods are gradient-based.

* Moustafa Alzantot and Yash Sharma contribute equally to this work.

[1] https://github.com/nesl/nlp_adversarial_examples

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2890–2896
Brussels, Belgium, October 31 - November 4, 2018.

Such methods, however, rely on the fact that adding small perturbations to many pixels in the image will not have a noticeable effect on a human viewer. This approach obviously does not transfer to the natural language domain, as all changes are perceptible. Furthermore, unlike continuous image pixel values, words in a sentence are discrete tokens. Therefore, it is not possible to compute the gradient of the network loss function with respect to the input words. A straightforward workaround is to project input sentences into a continuous space (e.g. word embeddings) and consider this as the model input. However, this approach also fails because it still assumes that replacing *every* word with words nearby in the embedding space will not be noticeable. Replacing words without accounting for syntactic coherence will certainly lead to improperly constructed sentences which will look odd to the reader.

Relative to the image domain, little work has been pursued for generating natural language adversarial examples. Given the difficulty in generating semantics-preserving perturbations, distracting sentences have been added to the input document in order to induce misclassification (Jia and Liang, 2017). In our work, we attempt to generate semantically and syntactically similar adversarial examples, via word replacements, resolving the aforementioned issues. Minimizing the number of word replacements necessary to induce misclassification has been studied in previous work (Papernot et al., 2016), however without consideration given to semantics or syntactics, yielding incoherent generated examples. In recent work, there have been a few attempts at generating adversarial examples for language tasks by using back-translation (Iyyer et al., 2018), exploiting machine-generated rules (Ribeiro et al., 2018), and searching in underlying semantic space (Zhao et al., 2018). In addition, while preparing our submission, we became aware of recent work which target a similar contribution (Kuleshov et al., 2018; Ebrahimi et al., 2018). We treat these contributions as parallel work.

3 Attack Design

3.1 Threat model

We assume the attacker has black-box access to the target model; the attacker is not aware of the model architecture, parameters, or training data, and is only capable of querying the target model with supplied inputs and obtaining the output predictions and their confidence scores. This setting has been extensively studied in the image domain (Papernot et al., 2017; Chen et al., 2017a; Alzantot et al., 2018), but has yet to be explored in the context of natural language.

3.2 Algorithm

To avoid the limitations of gradient-based attack methods, we design an algorithm for constructing adversarial examples with the following goals in mind. We aim to minimize the number of modified words between the original and adversarial examples, but only perform modifications which retain semantic similarity with the original and syntactic coherence. To achieve these goals, instead of relying on gradient-based optimization, we developed an attack algorithm that exploits population-based gradient-free optimization via genetic algorithms.

An added benefit of using gradient-free optimization is enabling use in the black-box case; gradient-reliant algorithms are inapplicable in this case, as they are dependent on the model being differentiable and the internals being accessible (Papernot et al., 2016; Ebrahimi et al., 2018).

Genetic algorithms are inspired by the process of natural selection, iteratively evolving a population of candidate solutions towards better solutions. The population of each iteration is a called a *generation*. In each generation, the quality of population members is evaluated using a *fitness* function. "Fitter" solutions are more likely to be selected for breeding the next generation. The next generation is generated through a combination of *crossover* and *mutation*. Crossover is the process of taking more than one parent solution and producing a child solution from them; it is analogous to reproduction and biological crossover. Mutation is done in order to increase the diversity of population members and provide better exploration of the search space. Genetic algorithms are known to perform well in solving combinatorial optimization problems (Anderson and Ferris, 1994; Mühlenbein, 1989), and due to employing a population of candidate solutions, these algorithms can find successful adversarial examples with fewer modifications.

`Perturb` Subroutine: In order to explain our algorithm, we first introduce the subroutine `Perturb`. This subroutine accepts an input sentence $\mathbf{x}_{cur}$ which can be either a modified sentence or the same as $\mathbf{x}_{orig}$. It randomly selects a word w

in the sentence $\mathbf{x}_{cur}$ and then selects a suitable replacement word that has similar semantic meaning, fits within the surrounding context, and increases the $target$ label prediction score.

In order to select the best replacement word, `Perturb` applies the following steps:

- Computes the N nearest neighbors of the selected word according to the distance in the GloVe embedding space (Pennington et al., 2014). We used euclidean distance, as we did not see noticeable improvement using cosine. We filter out candidates with distance to the selected word greater than δ. We use the counter-fitting method presented in (Mrkšić et al., 2016) to post-process the adversary's GloVe vectors to ensure that the nearest neighbors are synonyms. The resulting embedding is independent of the embeddings used by victim models.
- Second, we use the Google 1 billion words language model (Chelba et al., 2013) to filter out words that do not fit within the context surrounding the word w in $\mathbf{x}_{cur}$. We do so by ranking the candidate words based on their language model scores when fit within the replacement context, and keeping only the top K words with the highest scores.
- From the remaining set of words, we pick the one that will maximize the target label prediction probability when it replaces the word w in $\mathbf{x}_{cur}$.
- Finally, the selected word is inserted in place of w, and `Perturb` returns the resulting sentence.

The selection of which word to replace in the input sentence is done by random sampling with probabilities proportional to the number of neighbors each word has within Euclidean distance δ in the counter-fitted embedding space, encouraging the solution set to be large enough for the algorithm to make appropriate modifications. We exclude common articles and prepositions (e.g. a, to) from being selected for replacement.

Optimization Procedure: The optimization algorithm can be seen in Algorithm 1. The algorithm starts by creating the initial generation $\mathcal{P}^0$ of size S by calling the `Perturb` subroutine S times to create a set of distinct modifications to the original sentence. Then, the fitness of each population member in the current generation is computed as the target label prediction probability, found by querying the victim model function f. If a population member's predicted label is equal to the target label, the optimization is complete. Otherwise, pairs of population members from the current generation are randomly sampled with probability proportional to their fitness values. A new *child* sentence is then synthesized from a pair of parent sentences by independently sampling from the two using a uniform distribution. Finally, the `Perturb` subroutine is applied to the resulting children.

Algorithm 1 Finding adversarial examples

for $i = 1, ..., S$ in population **do**
  $\mathcal{P}_i^0 \leftarrow \texttt{Perturb}(\mathbf{x}_{orig}, target)$
for $g = 1, 2...G$ generations **do**
  for $i = 1, ..., S$ in population **do**
    $F_i^{g-1} = f(\mathcal{P}_i^{g-1})_{target}$
  $\mathbf{x}_{adv} = \mathcal{P}^{g-1}_{\arg\max_j F_j^{g-1}}$
  if $\arg\max_c f(\mathbf{x}_{adv})_c == t$ **then**
    return $\mathbf{x}_{adv}$ ▷ {Found successful attack}
  else
    $\mathcal{P}_1^g = \{\mathbf{x}_{adv}\}$
    $p = Normalize(F^{g-1})$
    for $i = 2, ..., S$ in population **do**
      Sample $parent_1$ from $\mathcal{P}^{g-1}$ with probs p
      Sample $parent_2$ from $\mathcal{P}^{g-1}$ with probs p
      $child = Crossover(parent_1, parent_2)$
      $child_{mut} = \texttt{Perturb}(child, target)$
      $\mathcal{P}_i^g = \{child_{mut}\}$

4 Experiments

To evaluate our attack method, we trained models for the sentiment analysis and textual entailment classification tasks. For both models, each word in the input sentence is first projected into a fixed 300-dimensional vector space using GloVe (Pennington et al., 2014). Each of the models used are based on popular open-source benchmarks, and can be found in the following repositories[2][3]. Model descriptions are given below.

Sentiment Analysis: We trained a sentiment analysis model using the IMDB dataset of movie reviews (Maas et al., 2011). The IMDB dataset consists of 25,000 training examples and 25,000 test examples. The LSTM model is composed of 128 units, and the outputs across all time steps are

[2] https://github.com/keras-team/keras/blob/master/examples/imdb_lstm.py

[3] https://github.com/Smerity/keras_snli/blob/master/snli_rnn.py

Original Text Prediction = **Negative**. (Confidence = 78.0%)
*This movie had **terrible** acting, **terrible** plot, and **terrible** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **considered** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **kids** they didn't understand that theme.*
Adversarial Text Prediction = **Positive**. (Confidence = 59.8%)
*This movie had **horrific** acting, **horrific** plot, and **horrifying** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **regarded** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **youngsters** they didn't understand that theme.*

Table 1: Example of attack results for the sentiment analysis task. Modified words are highlighted in green and red for the original and adversarial texts, respectively.

Original Text Prediction: **Entailment** (Confidence = 86%)
Premise: *A runner wearing purple strives for the finish line.* **Hypothesis:** *A **runner** wants to head for the finish line.*
Adversarial Text Prediction: **Contradiction** (Confidence = 43%)
Premise: *A runner wearing purple strives for the finish line.* **Hypothesis:** *A **racer** wants to head for the finish line.*

Table 2: Example of attack results for the textual entailment task. Modified words are highlighted in green and red for the original and adversarial texts, respectively.

	Sentiment Analysis		Textual Entailment	
	% success	% modified	% success	% modified
`Perturb` baseline	52%	19%	–	–
Genetic attack	97%	14.7%	70%	23%

Table 3: Comparison between the attack success rate and mean percentage of modifications required by the genetic attack and perturb baseline for the two tasks.

averaged and fed to the output layer. The test accuracy of the model is 90%, which is relatively close to the state-of-the-art results on this dataset.

Textual Entailment: We trained a textual entailment model using the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015). The model passes the input through a ReLU "translation" layer (Bowman et al., 2015), which encodes the premise and hypothesis sentences by performing a summation over the word embeddings, concatenates the two sentence embeddings, and finally passes the output through 3 600-dimensional ReLU layers before feeding it to a 3-way softmax. The model predicts whether the premise sentence entails, contradicts or is neutral to the hypothesis sentence. The test accuracy of the model is 83% which is also relatively close to the state-of-the-art (Chen et al., 2017b).

4.1 Attack Evaluation Results

We randomly sampled 1000, and 500 *correctly classified* examples from the test sets of the two tasks to evaluate our algorithm. Correctly classified examples were chosen to limit the accuracy levels of the victim models from confounding our results. For the sentiment analysis task, the attacker aims to divert the prediction result from positive to negative, and vice versa. For the textual entailment task, the attacker is only allowed to modify the hypothesis, and aims to divert the prediction result from 'entailment' to 'contradiction', and vice versa. We limit the attacker to maximum $G = 20$ iterations, and fix the hyperparameter values to $S = 60$, $N = 8$, $K = 4$, and $\delta = 0.5$. We also fixed the maximum percentage of allowed changes to the document to be 20% and 25% for the two tasks, respectively. If increased, the success rate would increase but the mean quality would decrease. If the attack does not succeed within the iterations limit or exceeds the specified threshold, it is counted as a failure.

Sample outputs produced by our attack are shown in Tables 1 and 2. Additional outputs can be found in the supplementary material. Table 3 shows the attack success rate and mean percentage of modified words on each task. We compare to the `Perturb` baseline, which greedily applies the `Perturb` subroutine, to validate the use of population-based optimization. As can be seen

from our results, we are able to achieve high success rate with a limited number of modifications on both tasks. In addition, the genetic algorithm significantly outperformed the `Perturb` baseline in both success rate and percentage of words modified, demonstrating the additional benefit yielded by using population-based optimization. Testing using a single TitanX GPU, for sentiment analysis and textual entailment, we measured average runtimes on success to be 43.5 and 5 seconds per example, respectively. The high success rate and reasonable runtimes demonstrate the practicality of our approach, even when scaling to long sentences, such as those found in the IMDB dataset.

Speaking of which, our success rate on textual entailment is lower due to the large disparity in sentence length. On average, hypothesis sentences in the SNLI corpus are 9 words long, which is very short compared to IMDB (229 words, limited to 100 for experiments). With sentences that short, applying successful perturbations becomes much harder, however we were still able to achieve a success rate of 70%. For the same reason, we didn't apply the `Perturb` baseline on the textual entailment task, as the `Perturb` baseline fails to achieve any success under the limits of the maximum allowed changes constraint.

4.2 User study

We performed a user study on the sentiment analysis task with 20 volunteers to evaluate how perceptible our adversarial perturbations are. Note that the number of participating volunteers is significantly larger than used in previous studies (Jia and Liang, 2017; Ebrahimi et al., 2018). The user study was composed of two parts. First, we presented 100 adversarial examples to the participants and asked them to label the sentiment of the text (i.e., positive or negative.) 92.3% of the responses matched the original text sentiment, indicating that our modification did not significantly affect human judgment on the text sentiment. Second, we prepared 100 questions, each question includes the original example and the corresponding adversarial example in a pair. Participants were asked to judge the similarity of each pair on a scale from 1 (very similar) to 4 (very different). The average rating is 2.23 ± 0.25, which shows the perceived difference is also small.

4.3 Adversarial Training

The results demonstrated in section 4.1 raise the following question: How can we defend against these attacks? We performed a preliminary experiment to see if adversarial training (Madry et al., 2018), the only effective defense in the image domain, can be used to lower the attack success rate. We generated 1000 adversarial examples on the cleanly trained sentiment analysis model using the IMDB training set, appended them to the existing training set, and used the updated dataset to adversarially train a model from scratch. We found that adversarial training provided no additional robustness benefit in our experiments using the test set, despite the fact that the model achieves near 100% accuracy classifying adversarial examples included in the training set. These results demonstrate the diversity in the perturbations generated by our attack algorithm, and illustrates the difficulty in defending against adversarial attacks. We hope these results inspire further work in increasing the robustness of natural language models.

5 Conclusion

We demonstrate that despite the difficulties in generating imperceptible adversarial examples in the natural language domain, semantically and syntactically similar adversarial examples can be crafted using a black-box population-based optimization algorithm, yielding success on both the sentiment analysis and textual entailment tasks. Our human study validated that the generated examples were indeed adversarial and perceptibly quite similar. We hope our work encourages researchers to pursue improving the robustness of DNNs in the natural language domain.

Acknowledgement

This research was supported in part by the U.S. Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-16-3-0001, the National Science Foundation under award # CNS-1705135, OAC-1640813, and IIS-1760523, and the NIH Center of Excellence for Mobile Sensor Data-to-Knowledge (MD2K) under award 1-U54EB020404-01. Ahmed Elgohary is funded by an IBM PhD Fellowship. Any findings in this material are those of the author(s) and do not reflect the views of any of the above funding agencies. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, and Mani Srivastava. 2018. Genattack: Practical black-box attacks with gradient-free optimization. *arXiv preprint arXiv:1805.11090*.

Edward J Anderson and Michael C Ferris. 1994. Genetic algorithms for combinatorial optimization: the assemble line balancing problem. *ORSA Journal on Computing*, 6.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing*.

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*.

Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *Advances in neural information processing systems: Deep Learning and Security Workshop*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2018. EAD: Elastic-net attacks to deep neural networks via adversarial examples. In *AAAI Conference on Artificial Intelligence*.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017a. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced lstm for natural language inference. In *Association for Computational Linguistics*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Association for Computational Linguistics*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *North American Association for Computational Linguistics*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing*.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems. *arXiv preprint arXiv:1602.02697*.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial machine learning at scale. In *International Conference on Learning Representations*.

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 *Association for Computational Linguistics: Human Language Technologies*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *North American Chapter of the Association for Computational Linguistics*.

Heinz Mühlenbein. 1989. Parallel genetic algorithms, population genetics and combinatorial optimization. In *Workshop on Parallel Processing: Logic, Organization, and Technology*.

N. Papernot, P. McDaniel, A. Swami, and R. Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. *arXiv preprint arXiv:1604.08275*.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *ACM on Asia Conference on Computer and Communications Security*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Association for Computational Linguistics*.

Yash Sharma and Pin-Yu Chen. 2018. Attacking the madry defense model with l1-based adversarial examples. In *International Conference on Learning Representations: Workshops*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.