Adversarial Examples for Natural Language Classification Problems

1. **What is the primary lesson(s) you took away from this paper (avoid abstract level summary)?**

   The authors in this paper proposed a white-box attack (the attackers know the model architecture, parameters, or training datasets) that uses a general optimization algorithm for constructing adversarial inputs and referred to this general type of adversarial attack as *altered* adversarial examples, comprising two constraints that capture sentence similarity on two levels of similarity: semantic and syntactic.

   Additionally, the authors studied adversarial examples on three natural language classification tasks: spam filtering using the Trec07p dataset, sentiment analysis using Yelp reviews, and fake news detection using the Fake News dataset, using four models: Naive Bayes, LSTM, word-level CNNs, and character-level CNNs, to perform these classification tasks. All models were susceptible to adversarial examples to a certain degree, which depends partly on the task. Specific problems, such as spam filtering, seem easier to classify and are less amenable to adversarial inputs; conversely, it is easier to fool the models on more complex tasks, such as fake news detection.

   Lastly, the authors verified the quality of their adversarial examples via human experiments on Amazon Mechanical Turk; the human evaluators achieved similar accuracies, suggesting that their adversarial examples preserved key semantics sufficiently to be recognized by a human. The authors further studied the use of adversarial examples to defend these models and improve the accuracy of classification algorithms via adversarial training.

2. **Could I have done this work if I had the idea why or why not?**

   I think if I had the idea first, I could have done this work. The high-level idea of the attack is to iteratively consider, at each step, all valid one-word changes to a sentence (i.e., which satisfy the semantic and syntactic constraints) and choose the one word that improves the objective the most, using Word2Vec's Word Cosine distances and GloVe's most similar words. Since I have good experience working with GloVe and its features, I believe I could use this knowledge to work with Word2Vec and its features. Plus, the authors' experiments were based on models and datasets that are publicly available; they only developed their attack algorithm.

3. **Is there anything I could do to repeat or validate? (Make notes of materials available from the paper- data sets, source code).**

   Yes, the authors are open-sourcing their attack and implementations of the four used models and publicly published them on GitHub here: https://github.com/ldf921/adversarial-text. The authors also used three free access datasets: the 2007 TREC Public Spam Corpus [1], the Yelp Review Polarity dataset [2], and George McIntire Fake News dataset [3].

   [1] https://plg.uwaterloo.ca/~gvcormac/treccorpus07.
   [2] http://goo.gl/JyCnZq.
   [3] https://github.com/payamesfandiari/fake_news_finder/tree/master/data.

4. **What is my best idea for follow on work that I could personally do?**

   My best idea for the follow-on work that I could personally do is to first repeat the same work using their English datasets to develop a better understanding of how the attack works. Secondly, I could repeat the same work or part of it (sentiment analysis) using Arabic sentiment analysis datasets, examine the generated results, and test if this kind of adversarial text attack works in different languages like Arabic.

5. **What is my best idea for follow on work that I'd like to see the authors or others do?**

My best idea for the follow-on work that I would like to see the authors or others do is to improve the authors' simple perturbations via a more sophisticated algorithm that takes advantage of language processing technologies, such as syntactic parsing, named entity recognition, or even paraphrasing.

6. **Any logistical experimental lessons I learned (this is a little different than #1 here you are looking for techniques)?**

I liked how the authors mathematically defined the adversarial examples in both image and text classifications in Sections 2 and 3 and clearly defined the two constraints that capture the sentence similarity on the two levels of semantic similarity and syntactic similarity.

I also liked the idea of having a human study as an evaluation method for their attack's results, and at the same time, they generated new results that could be a contribution by comparing them to the attack's results. The authors used crowd workers in Amazon Mechanical Turk to assign labels positive or negative reviews to both the original data points and their adversarially altered versions.

7. **How does this compare to the other papers we read? Most similar? How different? Other comparisons?**

This paper, Kuleshov et al. (2018), is the second paper I read that discusses the white-box attacks (the attackers are aware of the model architecture, parameters, or training datasets). Kuleshov et al. (2018) is similar to Alzantot et al. (2018) in terms of the embedding space and the used method of word-level replacements and similar to Ebrahimi et al. (2018) in terms of the attack type of white-box and the used method of word-level replacements as well. On the other hand, Kuleshov et al. (2018) differs from both papers Gao et al. (2018) and Ebrahimi et al. (2018) in terms of the input space and differs from both papers Gao et al. (2018) and Alzantot et al. (2018) in terms of the attack type of black-box.

8. **What is your biggest criticism of the paper?**

There is no big criticism of the paper from my side. Yet, the paper was submitted to the ICLR 2018 conference and was rejected. The paper's reviewers pointed out a few major criticisms that can be openly accessed here: https://openreview.net/forum?id=r1QZ3zbAZ.

9. **List 3 cited references or terms/concepts that you would be most interested in reading/learning more about.**

I am interested in reading the four references of the used machine learning/deep learning models in this paper that are: Naive Bayes (Wang & Manning, 2012), LSTM (Zhang et al., 2015), word-level CNNs (Kim, 2014), and character-level CNNs (Conneau et al., 2016).

Sida Wang and Christopher D Manning. ***Baselines and bigrams: Simple, good sentiment and topic classification.*** *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pp. 90–94. Association for Computational Linguistics, 2012.*

Xiang Zhang, Junbo Zhao, and Yann LeCun. ***Character-level convolutional networks for text classification.*** *In Advances in Neural Information Processing Systems, pp. 649–657, 2015.*

Yoon Kim. ***Convolutional neural networks for sentence classification.*** *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.*

Alexis Conneau, Holger Schwenk, Loic Barrault, and Yann Lecun. ***Very deep convolutional networks for text classification.*** *arXiv preprint arXiv:1606.01781, 2016.*