Implementing and Evaluating the Probability Weighted Word Saliency Algorithm as a Method of Adversarial Example Generation for Deep Neural Networks

Christiana Marchese
Pomona College
cemb2020@mymail.pomona.edu

1 Introduction

With the increasing ubiquity of large deep learning models for tasks in Natural Language Processing (NLP), like ChatGPT, methods of "tricking" such models have rapidly grown in popularity among users. However, the exploitation of these models' vulnerabilities raises real concerns in the realm of privacy and content moderation, and these concerns raise a question: how do engineers find a model's weaknesses before the user? Fine-tuning models with adversarial examples is one solution. Adversarial examples are data inputs that expose the weaknesses of models by getting them to respond in unexpected, and usually undesirable, ways.

In my work, I implement a method of adversarial example generation called the Probability Word Saliency (PWWS) algorithm (Ren et al., 2019). I then test this algorithm on 75 neural networks trained for sentiment analysis. I measure the effectiveness and efficiency of the PWWS algorithm through its ability to create adversarial examples for these models given text that these models correctly analyze. I also look at how variations of models' vocabulary size and dataset size may affect the PWWS algorithm's efficiency and effectiveness.

2 Methods

2.1 The PWWS Algorithm

The PWWS algorithm [1] is an approach to adversarial example generation for NLP problems. In order to retain the semantic and syntactical structure of text, PWWS uses synonym replacement and named entity (NE) replacement in order to generate adversarial examples.

The algorithm starts with a sentence, calculating the word saliency of each word. Word saliency is the change in the output probability of the classifier if a word w_i is set to unknown (Ren et al., 2019). For a given word, the algorithm then creates a synonym set via WordNet[2] and assesses which synonym, upon word replacement, creates the greatest change in prediction accuracy. If the word is an NE, it goes into the list of given NEs and an NE of the same type is selected. These (word, replacement) pairs are then scored. The score function takes into account the word saliency of that word and the change in prediction accuracy by its replacement. The algorithm then replaces words in the original sentence, from word-replacement pairs with the highest score to the lowest, until an adversarial example is created; or no further replacements can occur, in which case, the algorithm has failed.

2.2 The Models and Data

In order to develop and evaluate the PWWS Algorithm, I trained 75 models: 5 iterations of models trained with 3 different vocab sizes (40,000, 60,000, and 80,000) and 5 different dataset sizes (10,000, 20,000, 30,000, 40,000, and 50,000), using the AWD-LSTM[3] pre-trained model. The models were trained over 23 epochs, unfreezing layers over time to maximize accuracy. The average validation accuracy of these models during training varied from 88.91% to 92.70%.

The data used to train and test the models is the IMDB Movie Review dataset[4], which is a dataset of movie reviews labeled under the sentiment categories of negative and positive.

2.3 Evaluation

In order to assess the PWWS algorithm's efficiency and effectiveness, 10 example reviews were taken from the training data; All 75 models accurately predicted the 10 reviews. Across all models, I then look at how many adversarial examples were able to be created, the number of word substitutions

[1] https://github.com/christymarc/AdversaryAlgorithm

[2] https://wordnet.princeton.edu/

[3] https://arxiv.org/abs/1708.02182

[4] https://ai.stanford.edu/ amaas/data/sentiment/

Label	Text	Adversarial Example Text
Positive	I find it hard to believe that this movie has such a low rating. It **is** arguably **one** of the best comedies ever made, and surely the best Bollywood comedy of the 90s. The film did not do too well on the box office and people had diametrically opposite reactions after seeing it. My guess is most people didn't expect it to be an all-out comedy and were expecting a regular movie. If you love comedies, this is a must-see. And Aamir Khan is outstanding.	I find it hard to believe that this movie has such a low rating . It **cost** arguably **zero** of the best comedies ever made , and surely the best Bollywood comedy of the 90s . The film did not do too well on the box office and people had diametrically opposite reactions after seeing it . My guess is most people did n't expect it to be an all-out comedy and were expecting a regular movie . If you love comedies , this is a must-see . And Aamir Khan is outstanding .

Table 1: Comparing original text of sentence 2 with adversarial example generated by PWWS

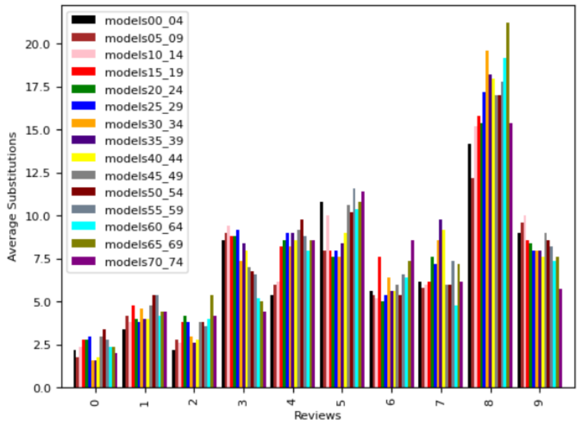

Figure 1: Models' average substitution rate over the 10 reviews

needed to create the adversarial example, and the sensibleness of generated examples.

3 Results

Of the 750 total reviews (10 per model) sent into 75 models, 748 adversarial examples were created; a yield rate of 99.73%. With only 2 iterations of model70_74 producing 9/10 adversarial examples.

The average rate of substitutions, in Figure 1, demonstrates that the efficiency of the algorithm is review dependent. Review length is one factor in this observation as review 8 is notably longer than many of the other reviews, but text complexity could also play a role and needs further investigation.

Regarding the sensibility of examples generated by PWWS, the replacements made by the algorithm do not always make semantic sense, as shown in Table 1. Notably, many words that WordNet deems as synonyms are labeled as such questionably so. This issue seems to elicit further investigation. However, what is interesting about the example in Table 1 is that despite the adversarial example, saying things like "this is a must-see", "Khan is outstanding", etc. the model still deems this review as negative sentiment. Therefore, while the algorithm is not perfect, it does still expose interesting weaknesses in the model.

4 Ethics Statement

Adversarial examples exploit the vulnerabilities of deep learning models; hence, the generation of such examples can expose model weaknesses that lead to things like privacy leaks and the production of toxic content. Therefore, in model training, an algorithm, like PWWS, that produces such examples is a great tool. However, a user using this algorithm could produce immense harm.

In addition, the generation of adversarial examples on large bodies of text is an energy and computationally expensive task. This type of large-scale computation can potentially produce real environmental harm.

5 Conclusion

With the growing importance of effective NLP model fine-tuning methods, the PWWS algorithm demonstrates promise in the area. As my results demonstrate, the PWWS algorithm has weaknesses in semantic production and efficiency with larger texts. However, the algorithm still exposes interesting weaknesses in models, which could ultimately be very beneficial for model fine-tuning. The ethical component of the PWWS algorithm also must be acknowledged as it has real consequences in the realm of privacy, content moderation, and environmental sustainability.

References

Muhao Ren, Xiang Yuan, Bo Zhang, and Wei Li. 2019. Generating natural language adversarial examples through probability weighted word saliency. *arXiv preprint arXiv:1904.06792*.

A Appendix

1) 41 hours worked 2) I did all of it.