GENERATING ADVERSARIAL EXAMPLES IN TEXT CLASSIFICATION

Yuxiao QU & Zhenyuan LIU Supervisor: Michael R. Lyu
Department of Computer Science and Engineering, The Chinese University of Hong Kong

Introduction

Recent studies have shown that by generating a series of adversarial samples can cause a well-trained model to be fooled[1]. As we can see from the following example, after deleting four letters of the original sentence, we can flip the prediction of the classifier.

Original	Adversarial
DVD player crapped out after one year, I also began having the incorrect disc problems that I've read about on here. the VCR still works, but the DVD side is useless...	DVD player crapped out afer one year, I also began having the incorrec disc problems that I've read about on her. the VCR still works, but the DVD side is useess...
99.87% negative	47.22% negative

Model

In this project, we choose three models, include word-based LSTM [2], word-based CNN [3] and character-based CNN [4] to evaluate our attack strategies. The character-based CNN model is 9 layers with 6 convolutional layers and 3 fully-connected layers.

The word-based CNN model is similar to the character-based CNN model, plus an extra word embedding layer.

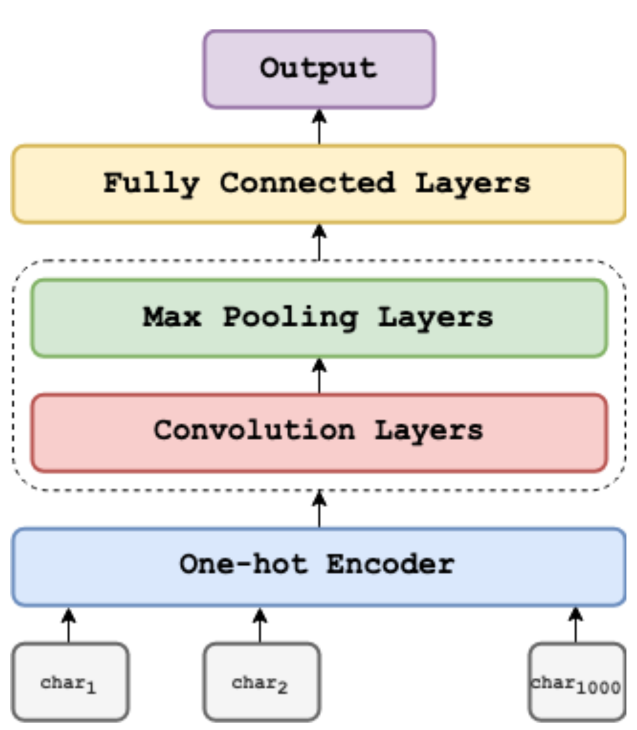

Figure 1. the structure of char-CNN

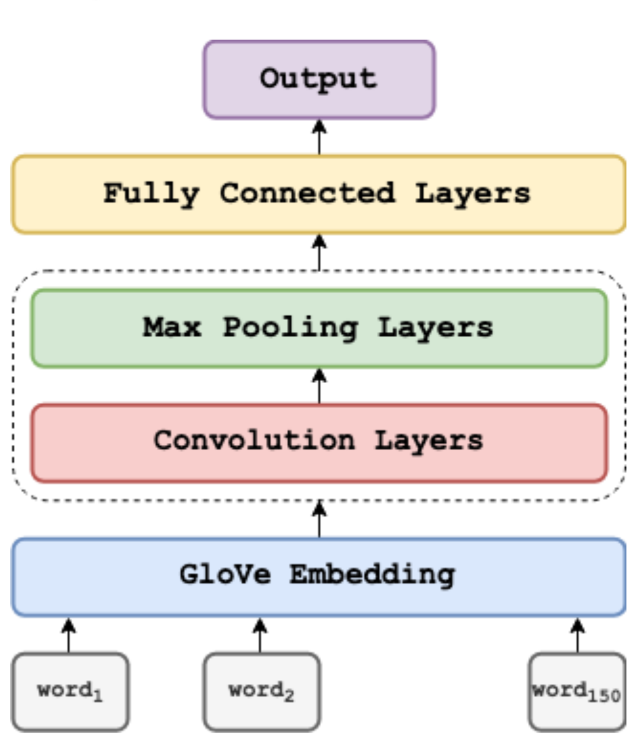

Figure 2. the structure of word-CNN

The LSTM model is consists of two LSTM layers with hierarchical attention, which is slightly variant of the hierarchical LSTM model proposed by Zichao Yang etc.[4]

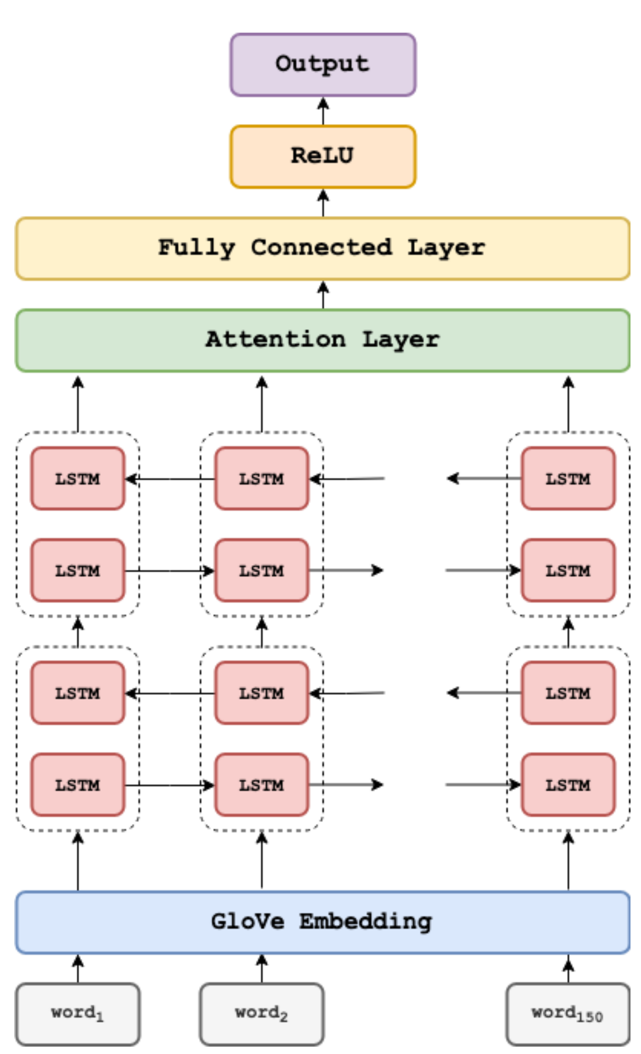

Figure 3. the structure of Bi-LSTM

Dataset

All these models are trained on Amazon Review Polarity Dataset, which is a binary classification dataset. Each class has 1,800,000 training samples and 200,000 testing samples.

Method

Recurrent Scoring Algorithm

Input: Input sequence x, Scoring function *score_func*,
Modification function *modif_func*, maximum edit distance ϵ
cost = 0
repeat forever:
score each token in x using *score_func*(·)
alter the token with greatest score using *modif_func*(·)
increase *cost* accordingly
if *cost* > ϵ or *length*(x) == 0:
return *ATTACK_FAIL*
if prediction of x flips:
return x

	Original	Occlusion	Deletion
Word-based	I love computer science I am from Hong Kong	I ____ computer science I am ____ Hong Kong	I computer science I am Hong Kong
Char-based	I love computer science I am from Hong Kong	I lov_ computer science I am f_om Hong Kong	I lov computer science I am fom Hong Kong

Table 1. Different **modification functions**

Delete-1 Score $D1S(x_i)$	I love computer science and engineering (x_i)
Delete-2 Score $D2S(x_i)$	I love computer science and engineering (x_i)
Temporal Head Score $THS(x_i)$	I love computer science and engineering (x_i)
Temporal Tail Score $TTS(x_i)$	I love computer science and engineering (x_i)
Combined Score $CS(x_i)$	$CS(x_i) = THS(x_i) + \lambda TTS(x_i)$

Figure 4. Scoring functions

Illustration of scoring token 'science' using different scoring functions. The score is equal to the prediction probability of the blue part minus the prediction probability of the orange part.

Experiment

Evaluation Metrics: The decrease of accuracy after the model being attacked

1. Compare scoring functions on different models with different maximum edit distance.

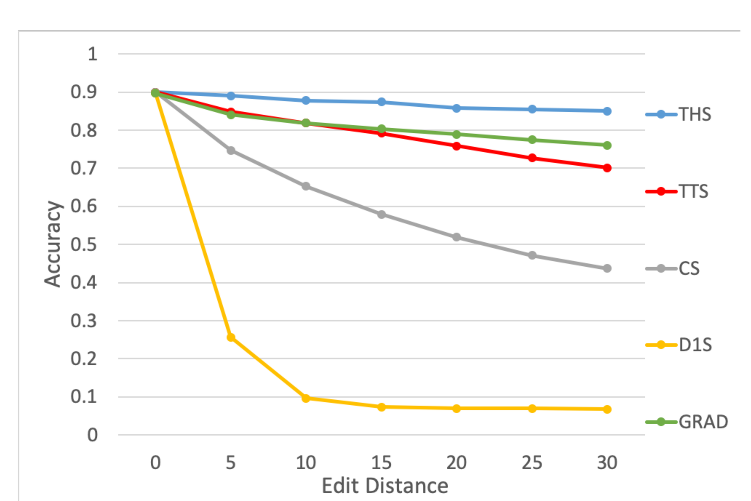

Figure 5. Char-CNN model

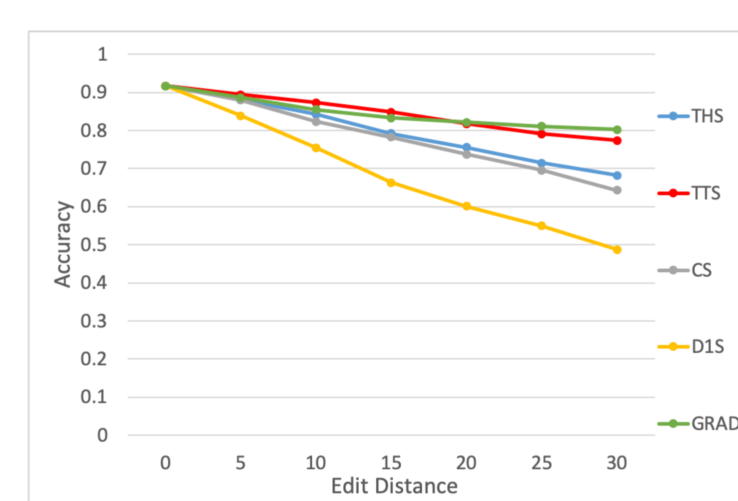

Figure 6. Word-CNN model

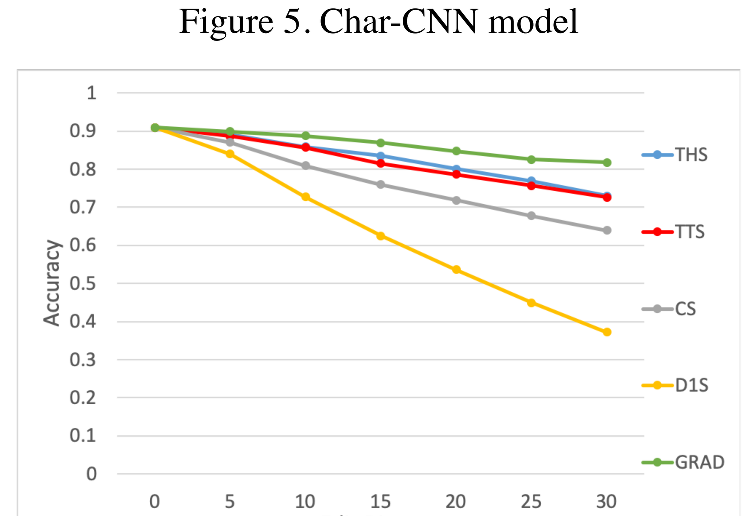

Figure 7. LSTM model

- Word-based models are more robust than character-based one
- Delete-1 scoring function is the best one among the four functions.

2. Attacking method is more efficient on Char-based model than on Word-based models.

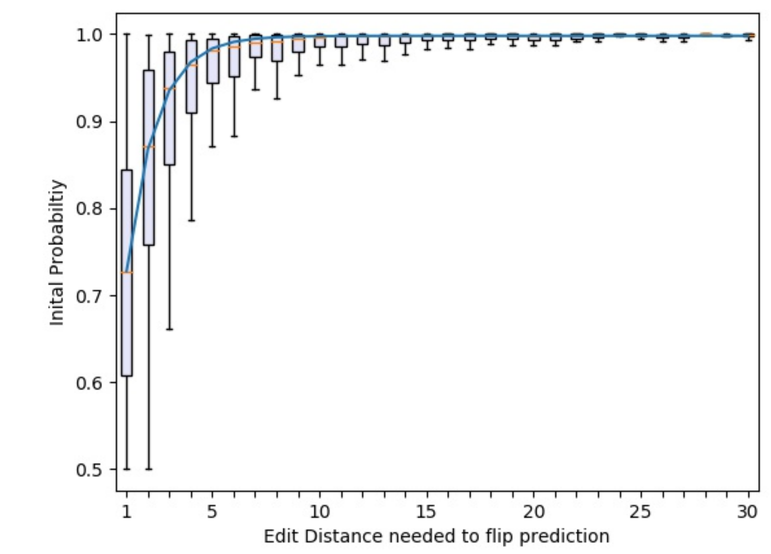

Figure 8. Char-CNN model

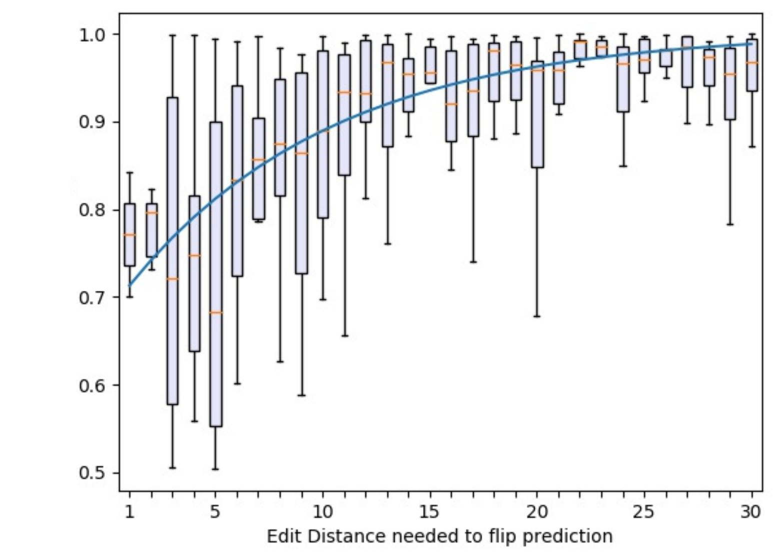

Figure 9. Word-CNN model

Word-based models are more robust since the attacking method is less efficient on word-based models

3. Comparison among delete-m functions as value of m varies

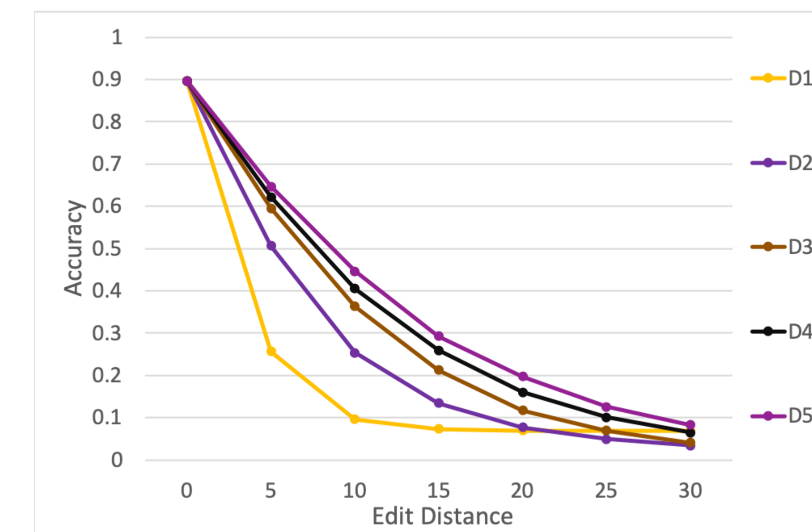

Figure 10. Delete-m

- There is some better scoring functions than the greedy one to conduct a black-box attack.
- Worth further investigations

4. Comparison between deletion and occlusion

	Char-CNN		Word-CNN	
	DEL	OCL	DEL	OCL
Original	90.00	90.00	90.97	90.97
D1S	6.79	6.79	37.26	37.26
THS	82.00	82.00	73.08	73.08
TTS	70.11	70.11	72.65	72.65
CS	43.74	43.74	63.92	63.92
D2S	3.36	3.36	55.98	55.98

DEL: Deletion OCL: Occlusion

Deletion and occlusion have the same attacking effect.

Conclusion

- Word-based models are more robust than character-based models in terms of accuracy decrease under the same constraint on maximum edit distance.
- Delete-m scoring functions may outperform the greedy algorithm.
- Deletion and occlusion have the same effects.

References

[1] Bin Liang et al. "Deep Text Classification Can be Fooled". In: CoRR abs/1704.08006 (2017). arXiv: 1704.08006. url: http://arxiv.org/abs/1704.08006.
[2] Sepp Hochreiter and Ju¨rgen Schmidhuber. "Long Short-Term Memory". In: Neural Comput. 9.8 (Nov. 1997), pp. 1735–1780. issn: 0899-7667. doi: 10.1162/neco.1997.9.8.1735. url: http://dx.doi.org/10.1162/neco.1997.9.8.1735.
[3] Yoon Kim. "Convolutional Neural Networks for Sentence Classification". In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. doi: 10.3115/v1/ D14-1181. url: https://www.aclweb.org/anthology/D14-1181.
[4] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. "Character-level Convolutional Networks for Text Classification". In: CoRR abs/1509.01626 (2015). arXiv: 1509.01626. url: http://arxiv. org/abs/1509.01626.