CIS 530 - Milestone 2 Report

Alexander Feng, Benedict Florance Arockiaraj, Jianxiong Cai, Xiaoyu Cheng

1 December 2021

1 Evaluation Measure

1.1 Accuracy

To evaluate the performance of generated universal triggers, we evaluate the model accuracy on subsets of the development dataset with and without the triggers. The subsets contain only examples of the label to flip. Since there is only one true label, all elements will be either true positives or false negatives. Ideally, the trigger should lower the original model accuracy.

$$Accuracy = \frac{TP}{TP + FN}$$

Note: TP: True Positive, FN: False Negative.

2 Baselines

For all of the baselines excluding hardcoded trigger baselines, we run the algorithm for 5 epochs. We use a batch size of 1 for sentiment analysis and a batch size of 5 for NLI. For each step in the epoch, we choose 40 candidate trigger sequences and choose the best trigger sequence (that has the lowest loss) out of that lot.

2.1 Simple Baselines

2.1.1 Random Attack

Random trigger sequence combination is sampled from the vocabulary.

2.1.2 Nearest Neighbor Attack

In nearest neighbors attack, we take a small step in the direction of the averaged gradient and find the nearest vector in the embedding matrix using k-d tree.

2.1.3 Hardcoded Attack

In the hardcoded attack, we pick an intuitive trigger sequence based on the target task and directly evaluate the model on the sequence. Triggers used in results are the ones with the greatest effect among a few (< 3) options.

2.1.4 Top Frequent Words (sentiment analysis only)

For sentiment analysis, we first split the training dataset into two splits (positive / negative) based on the sentiment label of each sentence. After removing stop words and common words (e.g. movie, film), we count word frequencies in each split and visualize the top frequent word in figure 1. Then, we generate the trigger to be the top 3 frequent words for each split. (i.e. [good, funny, comedy] for positive target label, [bad, much, characters] for negative target label).

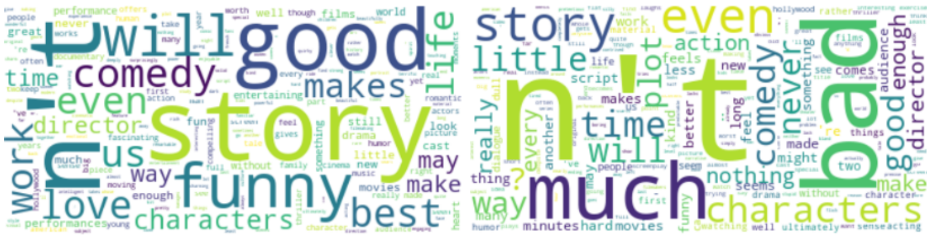

Figure 1: Top Frequency Words. Positive Sentiment (left) / Negative Sentiment (right)

2.2 Strong Baselines

2.2.1 Universal Adversarial Attack

We initialize the trigger by repeating the word ”the” multiple times as a placeholder. Next, we iteratively update the trigger words in order to increase the probability of the specific target prediction. For instance, a trigger for sentiment analysis is optimized to increase the probability of the negative class for various positive movie reviews. We perform the iterative updates based on the model's gradient equation from [1]

3 Performance of Baselines

3.1 Sentiment Analysis (Dataset: SST)

Baseline	Trigger Used/Generated	Acc w/o trig	Acc with trig
Random Triggers	pointless sooooo lifeless	0.909909	0.112612
Hardcoded Trigger	bad bad bad	0.909909	0.378378
Top Frequent Words	bad, much, characters	0.909909	0.38063
Nearest Neighbor Trigger	not its forefront	0.909909	0.680180
Universal Adversarial Trigger [1]	sucks lifeless lifeless	0.909909	0.085585

Table 1: Flipping positive to negative

Baseline	Trigger Used/Generated	Acc w/o trig	Acc with trig
Random Triggers	captivates unforgettable sensual	0.813084	0.203271
Hardcoded Trigger	positive positive positive	0.813084	.47196
Top Frequent Words	good, funny, comedy	0.813084	0.44159
Nearest Neighbor Trigger	above fascinates fascinating	0.813084	0.390186
Universal Adversarial Trigger [1]	vividly thought-provoking captivating	0.813084	0.093457

Table 2: Flipping negative to positive

3.2 Natural Language Inference (Dataset: SNLI)

Baseline	Trigger Used/Generated	Acc w/o trig	Acc with trig
Random Triggers	sisters	0.909582	0.006007
Hardcoded Trigger	Except	0.909582	0.051366
Nearest Neighbor Trigger	nobody	0.909582	0.00060
Universal Adversarial Trigger [1]	mars	0.909582	0.001201

Table 3: Flipping entailment to contradiction

Baseline	Trigger Used/Generated	Acc w/o trig	Acc with trig
Random Triggers	zombie	0.909582	0.001802
Hardcoded Trigger	spaghetti	0.909582	0.195854
Nearest Neighbor Trigger	no	0.909582	0.001802
Universal Adversarial Trigger [1]	joyously	0.909582	0.000030

Table 4: Flipping entailment to neutral

Baseline	Trigger Used/Generated	Acc w/o trig	Acc with trig
Random Triggers	talents	0.795302	0.667175
Hardcoded Trigger	because	0.795302	0.696156
Nearest Neighbor Trigger	touching	0.795302	0.666870
Universal Adversarial Trigger [1]	amusing	0.795302	0.660768

Table 5: Flipping contradiction to entailment

Baseline	Trigger Used/Generated	Acc w/o trig	Acc with trig
Random Triggers	festival	0.795302	0.660768
Hardcoded Trigger	spaghetti	0.795302	0.897498
Nearest Neighbor Trigger	anxiously	0.795302	0.660158
Universal Adversarial Trigger [1]	joyously	0.795302	0.595485

Table 6: Flipping contradiction to neutral

Baseline	Trigger Used/Generated	Acc w/o trig	Acc with trig
Random Triggers	rats	0.880680	0.103554
Hardcoded Trigger	because	0.880680	0.956723
Nearest Neighbor Trigger	mars	0.880680	0.014219
Universal Adversarial Trigger [1]	mars	0.880680	0.014219

Table 7: Flipping neutral to entailment

Baseline	Trigger Used/Generated	Acc w/o trig	Acc with trig
Random Triggers	no	0.880680	0.035239
Hardcoded Trigger	except	0.880680	0.412364
Nearest Neighbor Trigger	cat	0.880680	0.005106
Universal Adversarial Trigger [1]	cats	0.880680	0.026275

Table 8: Flipping neutral to contradiction

References

[1] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.