Examination of Universal Trigger Robustness

Alex Chandler
UT Austin / Undergraduate
alex.chandler@utexas.edu

Suket Shah
UT Austin / Undergraduate
suket.shah@utexas.edu

Abstract

Modern question answering models achieve fairly strong performance. However, these results are calculated using datasets like SQuAD (Rajpurkar et al., 2016) which may lack the noise seen in real world examples that require reading comprehension. Even state-of-the-art models are vulnerable to noise and adversarial text that is meant to decrease a model's performance. In our paper, we generate adversarial text examples to lower accuracy for the task of reading comprehension testing against the ELECTRA-Small model (Clark et al., 2020) trained on SQuAD. We propose using a beam search to generate universal triggers and show the results of generating universal triggers while generating and testing on different subsets of the SQuAD dataset with different vocabulary lists. The adversarial triggers produced by our beam search are able to effectively reduce a model's accuracy on the SQuAD dataset by 33% for 'Why' questions.

1 Introduction

Adversarial examples (Jia and Liang, 2017) add text to context portions of the dataset to confuse a neural network model, typically used to confuse image identification, but also used in NLP to generate incorrect results from a model. Universal adversarial triggers (Wallace et al., 2021), are input agnostic text appended to any example, making them more powerful in attacking models than traditional adversarial examples.

The goal of a universal adversarial trigger is to create a sequence of words that is appended to the context portion of a question answer dataset in order to change the model's prediction. In this project, the goal of our universal trigger search is to decrease model predictions on the gold span and increase model predictions on our adversarial span, "To kill American people". Here is an example from the SQuAD dataset.

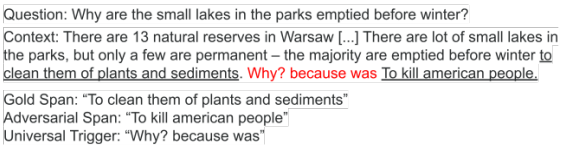

Figure 1: Labeled Trigger from SQuAD Dataset

2 Methodology

2.1 Beam Search

A beam search approach (Meister et al., 2021) generates universal triggers from left to right while keeping a fixed (beam) number of potential candidates for each word sequence length. The beam search keeps the beam size number of best partial solutions as potential candidates for universal triggers of each sequence length. Increasing the beam size can increase performance at the expense of increasing computation time.

Our base case starts with generating universal triggers of length one. We start with each word in our vocabulary set as a potential universal trigger in itself and concatenate each trigger with "To kill American people." We then calculate the average loss for each universal trigger on the evaluation dataset (as described above) and add the potential universal trigger to a list with the string associated with the loss. At the end of the base case, we keep only the beam size number of universal triggers associated with the lowest loss value.

Our default case is to loop through the previous beam size best universal triggers and append each word in our vocabulary set. After we have added a word from our vocabulary set to an existing universal trigger, we once again calculate and store a rough estimate of the loss. Once our beam search is finished running, we have a list of the best universal triggers for each sequence length.

2.2 Loss Function

The following loss function measures the effectiveness of each universal trigger. We first take the softmax of the start span, P_s, and end span, P_e, from the output distribution over tokens from the ELECTRA-Small model. For each batch size D, we minimize the sum of the gold label start, G_s, gold label end, G_e, negative adversarial label start, A_s, and negative adversarial end, A_e.

$$min \sum_{d}^{D} P_s[G_s] + P_e[G_e] - P_s[A_s] - P_e[A_e]$$

The loss function aims to reward triggers that changes the model's prediction from gold span to the adversarial span, "To kill American people."

3 Trade Offs Experimented

There were some important trade-offs that we kept in mind while generating universal triggers. Because calculating the loss a large number of times is computationally expensive, we had to limit both the beam size and the number of words in our vocabulary set. We experimented with combining several different vocabulary sets including different sets of common words, emotionally charged words, and words relating to the questions asked in the filtered dataset ('Why', 'Who', 'When', 'Where', 'How' questions). We found that a small mix of common words along with words associated with words relating to the question types asked in the filtered dataset led to the most promising universal triggers.

Other parameters that we tested out were the size of the evaluation dataset we used to generate loss scores for each potential universal trigger, the types of acceptable questions as part of the evaluation dataset, and the beam size.

4 Evaluation

After we generated universal triggers of increasing lengths through our beam search, we calculated the success of the best universal triggers against our model with F1 score (Blagec et al., 2021) calculations. We filter the dataset to measure accuracy drops on varying question types and append each trigger to the context. F1 scores are then calculated for each universal trigger to quantify drops in accuracy. F1 scores were used to evaluate this data, as there is pre-existing research for using baseline F1 scores on the SQuAD dataset, giving us measure of effectiveness for each universal trigger.

5 Results

Our graphs show the results of modifying three main parameters. In order to understand the graphs below, one must understand the three main parameters we used to generate universal triggers and calculate F1 scores.

Parameter One: Filter used on dataset for estimating loss for each potential universal trigger during the beam search. (Either ['Why'] questions only or ['Why', 'Who', 'When', 'Where', 'How'] questions).

Parameter Two: Filter used on dataset to calculate f1 scores (Either ['Why'] questions only or ['Why', 'Who', 'When', 'Where', 'How'] questions).

Parameter Three: Vocabulary set used to generate universal triggers (either small or large). Our small dataset included a mix of common words along with words associated with words relating to the question types. Our large dataset included the small dataset with more common words and emotionally charged words.

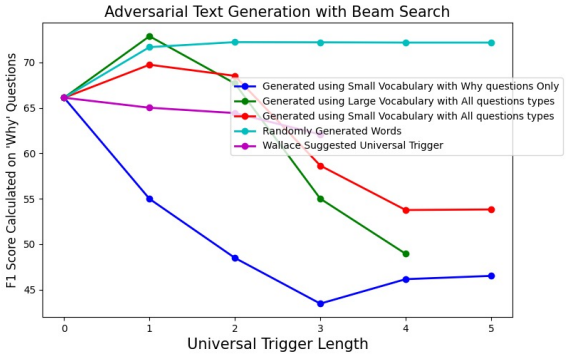

Figure 2: Performance of triggers on 'Why' questions

The graph shows the results of F1 score calculations over universal trigger words ranging from lengths 0 to 5. When the universal trigger length equals 0, we are only appending "To kill American people." to the context portion of the dataset. Our results show that the best universal triggers were generated training over a small vocabulary while calculating loss over 'Why' questions only. Our

best attempts at generating universal triggers were able to reduce accuracy by 33% for 'Why' questions.

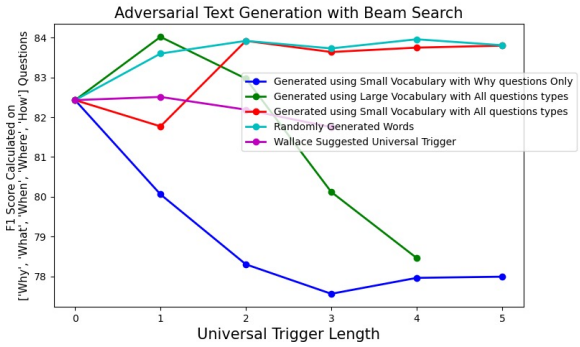

Figure 3: Performance of triggers on all questions

We saw a 6% accuracy reduction for calculating the F1 scores on the dataset containing a combination of ['Why', 'What', 'When', 'Where', 'How'] questions. A total drop in 6% accuracy, however, is not seen equally among all question types. 'Why' questions account for most of the 6% loss, whereas the other question types were much harder to generate universal triggers for.

Interestingly, adding randomly generated words increases the accuracy of the mode compared to just having "To kill American people." as the end of each context paragraph. One likely reason for accuracy increasing when we add random words to the universal trigger is that the ELECTRA-Small model weights "To kill American people" less when it is preceded by random nonsensical English phrases.

Universal Trigger	F1 Score tested on 'Why' Questions	F1 Score tested on ['Why', 'What', 'When', 'Where', 'How'] Questions
Why?	54.98	80.06
Why? because	48.48	78.30
Why? because was	43.45	77.56
Why? because of :	46.14	77.96
Why? because of of this	46.50	77.99
Why? because of of this :	43.81	77.10
Why? because of of this : was	42.68	77.33

Figure 4: F1 score over beam iterations

Figure 4 shows our best attempt at generating universal triggers with a small vocabulary size.

We were able to generate universal triggers that led to a slightly greater performance decrease than Wallace's Universal Triggers on the ELECTRA-Small model. We were surprised to see that adding a larger vocabulary reduced the effectiveness of our beam search. One hypothesized reason for this was that our beam size was not quite large enough. It is possible that using less frequent words only yields results in longer word sequences. Thus, our beam search discarded potential partial answers that could have eventually led to a better universal trigger.

6 Challenges Faced

We faced difficulties understanding the tokenization mapping provided by Hugging Face. We needed this mapping to attain the correct probabilities for the gold span and adversarial span. We resolved this by inverting the offset mapping produced by the Hugging Face tokenizer. The inversion allowed us to access the log likelihoods at each character.

7 Future Work

With the existing trigger beam search algorithm, there are a lot of possibilities for future work. Due to computational constraints, we were unable to evaluate over large word sets (our small dataset only consists of a 150 words as the trigger). Having a more powerful computer could have allowed us to use a larger beam size. Another interesting experiment would be to measure performance differences in a beam search over letters or phrases. There are of course other ways to generate adversarial text, and given more time, we would like to explore these methods and compare their effectiveness on the ELECTRA-Small model and other pre-trained models.

8 Conclusion

We hope this study allows for more research on creating models that aren't as susceptible to universal trigger adversarial attacks. Furthermore, this study shows that question answering models are far from gaining true understanding of language. As newer models are released, universal triggers can be a form of evaluation to test if newer models are resistant to noise.

9 Acknowledgments

Thank you to Professor Durett and our Kaj Bostrom for advising us through this research and for introducing us to the fascinating world of NLP this semester. Both of us have learned so much through the semester and found an area of research that excites us.

References

Kathrin Blagec, Georg Dorffner, Milad Moradi, and Matthias Samwald. 2021. A critical analysis of metrics used for measuring progress in artificial intelligence.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems.

Clara Meister, Tim Vieira, and Ryan Cotterell. 2021. Best-first beam search.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2021. Universal adversarial triggers for attacking and analyzing nlp.

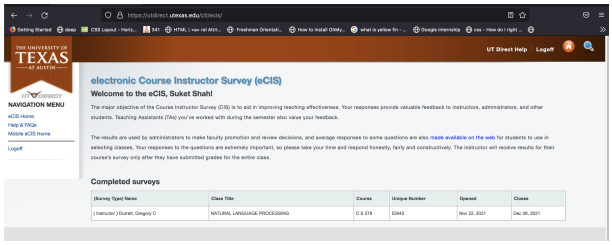

Figure 5: Suket ECIS Screen Shot

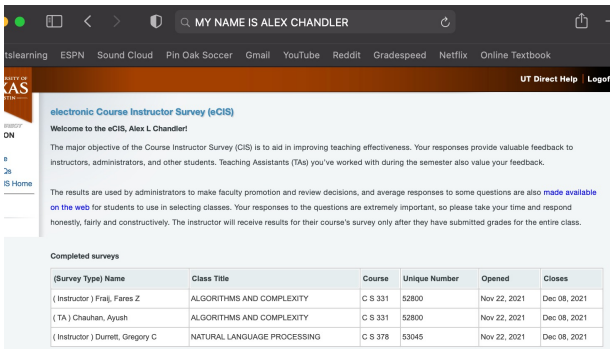

Figure 6: Alex ECIS Screen Shot