NLP Final: An Investigation of Concatenated Adversarial Sets

Viet Le, University of Texas at Austin

Abstract

This project dives into Adversarial Sets and their usage. We explore a simple fix of presenting the model with adversarial sets, even to the point of data leakage, as a way to explore how adversarial sets work with just concatenation and in context of semantic destabilization as a way to analyze whether or not adversarial questions concatenating at the end of a context is enough of a metric, or if some shuffling of sentences is warranted.

1 Introduction

This final project keeps the actual code and “fixes” simple. While originally set out to find a decent fix for the problem of adversarial sets causing models to preform worse on average (Robin Jia & Percy Liang, 2017). The proposal to append an adversarial sentence to the end of the context phrase was somewhat arbitrary; this project will check how this decision factors into preserving the model’s ability to answer question in relation to some simple “fixes”. Another topic is that, while we may train on information that is semantically structured the same, i.e., starting with a topic sentence and giving details, does testing on contexts where the semantic structure is perturbed yield interesting enough results that might alter the way we train our models?

The question of adversarial sets creating noise through an adversarial sentence only at the end of the context paragraph has been placed in motion. This technique tricks models that are not looking deep enough to find an answer through the context due to simply finding a phrase in the context that looked similar enough to the question posed. While this project is partly a review of the adversarial sets currently presenting the technique to concatenate at the end, it also looks at how the semantic structure of the context affects the evaluation of a model.

Our models may tend to do worse with adversarial sentences if they lack the ability to have selective attention. This is a phenomenon that we humans are able to do if we have cognitive function; this is displayed when we need to study in a busy area, we are generally able to ignore our surroundings and focus on the work in front of us. For our model, this will be measured in their ability to ignore adversarial sentences to focus on the actual information needed.

Also, apologies for any weird figures/tables. I ended up trying to avoid them the most I can due to the way word is currently messing them up, however, I have no clue how else to do this other than to image them in as word is not on my side. The instructions I found were not helpful so bear with me.

2 Failure Analysis

Squad	Squad AddOneSent	Squad AddSent	AddOneSent Adverse Only	AddSent Adverse Only
78.28	63.9	55.11	45.61	46.05

Our original model, ELECTRA-small (Clark et al., 2020), reported scores in the table associated (please bear with the lack of labels on tables). The values reported are the accuracies for the ELECTRA-small model, starting with the original SQuAD validation set, then augmented versions of SQuAD with adversary sets. In order we have a SQuAD validation set using Jia & Percy’s AddOneSent technique and their AddSent technique. After that, I parsed through their set and removed the items that are meant to show that the model could still answer the original questions and looked solely at items with adversarial sentences at the end, respectively from AddOneSent and AddSent. These results follow our expected hypothesis that the adversary sets do perturb the model’s ability to find the correct answer. Though it still is able to perform decently well, however, by looking at the last two sets we see that the model really underperforms for the adverse contexts.

Jia & Percy make the analysis that the models seem to perform well when the question has an *n*-gram match with a phrase in the sentence. When *n* is low, the adversarial sentence, which is an augmentation of the correct sentence (changing the subject and fact), the sentence is different enough, as a large percentage of the *n* changes due to their adversarial manipulation. This means the model may be able to more easily disregard the adversarial sentence, while if the sentence is long enough, the change will be proportionally smaller which makes the sentence close enough so that the model has a more difficult time deciding between which sentence, the correct sentence or adversarial sentence, contains the right answer. This is also affected by the matching of the question and the adversarial sentence, so that the question posed looks extremely similar to the adversarial question, except for some key subjects. A specific example is that the question: "Where did Super Bowl 50 take place?" was originally correctly answered "Levi's Stadium in the San Francisco Bay Area at Santa Clara, California," but when given the adversarial cue: "Champ Bowl 40 took place in Chicago." the model instead simply predicts "Chicago." Here we see that the question matches well with the adversarial cue, but the previous answer had many more tokens to so to the model it may have seemed like the adversarial cue more directly answered the question.

At this point, we have identified the power of adversarial set testing as the model seems to perform worse as it prefers sentences that are more similar to the question. From the above table, we can see this behavior causes a general decrease in performance for this specific type of attack. This adversarial attack may not be uncommon either, as since we are pulling data from the web (SQuAD is based on Wikipedia entries) and it could just be that the context references other subjects in a similar manner. We want our model to be based on its ability to relate the subject of the sentence to the rest of the sentence and find if it answers the question, not if our passage has a sentence that contains a high enough word similar that it will pay attention to that sentence.

3 A Simple "Fix"

This fix relies on a simple goal, what if we were somehow able to inoculate the model to this type of attack. To do so, we took the original model, and gave it part of the adversarial set (just under half) to train on. Since the set is smaller, if the original schema the model formed is in fact wrong, in that it only went as far as spurious *n*-gram association between questions and answers, we should see a decrease in general accuracy, but an increase in our test-set for adversarial questions (that the model did not train on).

Squad	AddOneSent Adverse Half ▲
75.32	72.15

We retrain ELECTRA-small model while preserving its original weights with some items containing only items with adversarial sentences concatenated at the end. While this doesn't test against the same exact set, as we now only have a smaller subset available, we can see accuracy has majorly increased for the adversary sets, and minorly decreased for the SQuAD set.

However, I believe this "fix" is not one that is an actual fix. This fix may simply be the model learning to just ignore the last sentence. This is where the idea of concatenation may fail. If a machine learning trainer, such as I, is either naïve enough or malicious enough, we could try just training our model to ignore these adversarial sets and since it is in the common pattern of being in the last sentence position, the model is able to pick this up quite easily.

4 Shifting Placement

One of the cited reasons to concatenate at the end is that we don't want to disturb the semantic structure of the context, which is generally going to have a topic sentence start and more details that would contain the answer of the subject. Our model does not learn to expect that other subjects may exist with similar information that question asks for. In response to that we can train our model to specifically ignore the concatenated adversarial sentences. However, in the real world, with less data cleaning, we might expect more sentences that relate to other subjects. We might also give a context, but we want to know more about a smaller subject within the context that it wasn't intended for. While these goals are out of the scope of this project, in terms of question answering, we might think the ability to do this sort of selective attention is a worthwhile goal.

To explore this idea, we decided to take the adversarial sentence and randomly place it within the context, instead of just concatenating it to the end. While the idea of maintaining semantic structure may seem important, the questions posed for the most part can and are generally answered within a singular sentence so the likelihood of inserting the adversarial sentence between two other sentences changing the meaning of the overall context is minimal.

	Squad	AddOneSent Adverse Half	Random Pos Half
Original	78.28	49.18	49.41
Original + Half AddOneSent	75.32	79.96	64.03

For the most part, the data agrees with the previous idea. When we take the adversarial data and shift the placement of it, the evaluation score no longer shows that dramatic improvement. However, it is important to note that the other model does train with adversarial sentences at the end and the improvement still exists, albeit smaller. This may be caused by the model being able to learn some of that selective attention, though not to the same level as its general question-answering ability.

It is important to note though, that the loss in accuracy for the random position idea may be the result of changing the positioning of the sentences. The model may have learned some information from the relationship from the sentences that may be useful and cause a similar fall in accuracy.

	Squad	Random Pos Half	Random Pos Squad ▼	Random Shuffle Squad
Original	78.28	49.41	76.62	74.37
Original + Half AddOneSent	75.32	64.03	73.09	70.89

From the data we generated, it seems that this may be the case, however, it does not seem like the changing of position causes a very significant change in accuracy. Here "Random Pos Half" is when we insert the adversarial set into a random location into the context for evaluation. Likewise, "Random Pos Squad" takes the original SQuAD validation set and takes the last sentence and inserts it randomly. "Random Shuffle Squad" takes the SQuAD validation set and shuffles the ordering of the sentences.

All of the sentences still maintain their individual meaning, but the overall context suffers from this change slightly as its semantic structure may be more incoherent. For the most part, this confirms the assumption that maintaining semantic structure does not disrupt the model's ability as much as adversarial attacks.

4.1 Shifting Placement Training

An interesting piece of the previous analysis is that even though we trained on adversarial being at the end, there is still an improvement in the model's ability of selective attention. To push this further, we decided to try to train the model with the original SQuAD training set and then train further using half of an adversary set, however with the same random inserted position. The hypothesis still stands from before, we expect a drop in SQuAD accuracy as it tries to learn a new schema and an increased ability for selective attention in ignoring the adversarial sentence.

	Squad	AddOneSent Adverse Half	Random Pos Half	Random Pos Squad	Random Shuffle Squad
Original	78.28	49.18	49.41	76.62	74.37
Original + Half AddOneSent	75.32	79.96	64.03	73.09	70.89
Original + Rand Pos Half Adv	62.33	57.81	52.8	60.4	56.7

For the most part, this ended up fitting our hypothesis much better, though it shows a more significant drop in accuracy than the previous method. We believe that this is due to the fact that the re-schematization process would be more difficult, and the training set was not large enough to regain that missing accuracy. It seems like it would be more difficult to train as since the previous method still kept the adversarial sentences at the end, the model may have just been able to find what type of sentences to ignore much more easily as it was always in the same location, whereas here it may not have been able to find that it was a sentence meant to be ignored. This problem would probably be mitigated if more examples were used to train, however we lack time and enough items to still train on without data leakage.

5 Subsidiary Gain/Problems

This section is called subsidiary gains/problems as these are results from other trained models that we created that are not as important to the main scope but may pose some questions and have potential gains for future study.

5.1 Is Semantic Destabilization *Really* Minimal?

While the above makes it seem like the reordering of sentences does not have a large effect, we decided to test this against models that were trained to cheat on adversarial sets. These models were trained with the adversarial sets on top of the original SQuAD set, so we should expect that when we evaluate on the adversarial sets again, they should have a much better score. Our data suggests this is the case, however, when we evaluate the set against the random position for the adversarial sentence, we see a drop in accuracy, which implies that the model was able to learn something that it can guess well with the adversarial sentence at the end, but not somewhere straddled in the middle. This drop is slightly unexpected as the above sets seem not to have such an extreme change, and the models themselves are meant to cheat. The way that the random position evaluates seems to suggest that the model may not always be learning selective attention in the way we might expect, but a combination of selective attention and position. This may be useful in helping to make the case that we should implement an adversarial set that contains random insertions instead of a simple concatenation.

	Squad	AddOneSent Adverse Only	AddSent Adverse Only	Random Pos Half
Original + AddOneSent	76.71	97.58	96.32	83.73
Original+ AddSent	74.41	99.49	99.6	87.67

5.2 All your data are *should* belong to us

Three other models were also trained; however, they were trained from scratch instead of off the base SQuAD trained ELECTRA-small model. They do have the problem of each of them not having enough data to make significant gains, however, the difference between them is palpable. Probably if given enough data, these results may seem more significant. However, what is significant is that though they are cheating as well as I allowed these models to train on all the adversarial data, it corroborates that the semantic destabilization may not be as minimal for adversarial sets by comparing the Adverse Only columns and the Random Pos column. An unanswered question we had following this is also the way that the accuracies differ between models, it is out of scope given the other questions we want to answer, but it seems interesting.

	Squad	AddOneSent Adverse Only	AddSent Adverse Only	Random Pos AddOne	Random Pos Squad	Random Shuffle Squad
AddOneSent	31.99	65.43	64.45	40.53	33.73	31.99
AddSent	37.69	94.02	95.19	67.59	36.91	34.67
AddAllAdverse	40.37	99.11	99.29	82.84	39.12	37.67

6 Discussion

While adversarial sets are useful in helping us evaluate models for their selective attention, the process in which we do so seems to be important as well and plays a role in how we should think about training our models and evaluating them. We have shown that it is somewhat easy to cheat the adversarial set process that just concatenates by just adding some examples of concatenated adversarial sentences. However, when we test against adversarial sentences that are inserted randomly, we see that this fix of inoculation does somewhat help against this specific type of attack, but it doesn't help enough to match the general performance.

Another interesting tidbit is that, for the most part, it seems like shifting the order of the sentences seems to have minimal effects on the accuracy of the model, it does negatively impact the model, but not enough to throw away this idea of random insertion for adversarial sentence.

We also think this makes the most sense overall as this direction would provide more focus on the idea of selective attention, where the model is able to limit its attention to sentences actually relevant to the question, instead of sentences that have an overlap in word and content. Though the naïve fix was able to bridge a lot of problems, we may want to pursue this random adversarial training as a way to be able to help generalize the context to real world data that is pluralist and not as well kept as SQuAD's data are.

7 Extra Graph

Here are all the models we trained and their resulting accuracy, this is not necessary for the paper and contains extra data that may not matter as much to the reader but is just meant to show more of the scale of the work done.

	Squad	AddOneSent Adverse Only	AddSent Adverse Only
Original	78.28	45.61	46.05
Original + AddOneSent	76.71	97.58	96.32
Original+ AddSent	74.41	99.49	99.6
Original + Half AddOneSent	75.32	82.08	79.96
Original + Rand Pos Half Adv	62.33	57.81	56.91
AddOneSent	31.99	65.43	64.45
AddSent	37.69	94.02	95.19
AddAllAdverse	40.37	99.11	99.29

	Random Pos AddOne	Random Pos Half	Random Pos Squad	Random Shuffle Squad
Original	44.85	49.41	76.62	74.37
Original + AddOneSent	83.73	86.54	74.75	72.17
Original+ AddSent	87.67	89.32	72.39	70.2
Original + Half AddOneSent	68.99	64.03	73.09	70.89
Original + Rand Pos Half Adv	52.85	52.43	60.4	56.7
AddOneSent	40.53	43.61	33.73	31.99
AddSent	67.59	72.38	36.91	34.67
AddAllAdverse	82.84	85.61	39.12	37.67

References

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pretraining Text Encoders as Discriminators Rather Than Generators. In Proceedings of the International Conference on Learning Representations (ICLR).

Ceyda Cinarel. Datasets: squad_adversarial. https://huggingface.co/datasets/squad_adversarial