Fine-Tuning Using SQuAD Adversarial to Improve Question Answering

Abstract

Our goal is to use the challenge dataset: Adversarial SQuAD to better fine-tune the ELECTRA-small model for question-answering in order to improve performance on the types of challenging examples seen in this dataset. We trained our model on the SQuAD dataset to determine a baseline for how the model performs on typical examples and then compare evaluation results on this dataset as well as Adversarial SQuAD. We then fine-tuned this model by training it on various increasing amounts of adversarial examples in conjunction with the original SQuAD examples with the hope that the model will then be able to distinguish between the made-up distractor context examples and the correct context of interest. Using this method, we observed up to an approximate 8 point boost for the EM and for the F1 score on average for the Adversarial SQuAD evaluation split, with an increasing trend based on the number of adversarial examples trained on.

1 Introduction

Question answering has been a critical NLP problem where a user poses a question in natural language and a QA system provides an immediate response in return. SQuAD is a popular reading comprehension dataset which consists of questions formulated by crowdworkers based on a set of Wikipedia articles. The answers to these questions is a "span" or section of text from a specific article (Rajpurkar et al., 2016). This reading comprehension task has been extremely successful in the past and our current ELECTRA-small model has an EM score of 78.3 and F1 score of 86.3. However, the limitations of true language understanding become evident when these language models are exposed to text that contains adversarially inserted sentences which closely imitate the correct answer (Jia and Liang, 2017). The Adversarial SQuAD dataset was created for this purpose in order to demonstrate how these models can easily fall victim to intentionally misleading natural language text. There have been several reasons for why this challenge dataset causes difficulty for models. It may be due to an inherent weakness in the dataset such as a lack of diversity in the questions and context text or an actual weakness in the model itself by being unable to properly adapt to the distribution of challenging examples. As described in (Liu et al.2019), a method for analyzing such a challenging dataset was creating in the form of fine-tuning a language model such that it is exposed to increasing amounts of challenge examples with the hopes that the model will be able to develop a more robust language comprehension.

We aim to apply this method of fine-tuning to the ELECTRA-small model training on different datasets which have examples from the original SQuAD set alongside an incrementally increased number of adversarial examples. Using the newly trained model, we will compare the types of mistakes made before and after with the hopes that many of the commonly made mistakes will be rectified in the newly trained model.

2 Approach

2.1 Datasets

In conjunction with the SQuAD dataset for training, we incorporated examples from the SQuAD adversarial dataset, specifically the AddSent examples. These examples used up to five candidate adversarial sentences that do not correctly answer the question but contain many similarities with the question. The training and test split varied based on the number of examples used for each training

session. The maximum number of adversarial examples used was 1000 in order to limit the bias towards these examples and so that there was a large enough sample of test examples for evaluation.

2.2 Experiments

We first trained the ELECTRA-small model on the SQuAD dataset and then evaluated it on the SQuAD adversarial dataset to have a baseline for how this model performs with adversarial examples. We then did some investigation into the types of mistakes that the model tends to make under these conditions. This analysis was performed through both human observation and a python script to compare similarities between the question, context, and the prediction. Once we developed an understanding for some specific examples and general mistakes that the model made, we then began training the model on datasets that added various amounts of adversarial examples to the original SQuAD dataset and compared how each version of the trained model handled adversarial examples. We predict that the more of these examples that we incorporate into training, the total number of incorrect predictions for adversarial examples would decrease on average.

Furthermore, we will observe how learning on these examples impacts evaluation on the original SQuAD dataset. If we do observe a decrease in performance on these original examples, this training may not actually be helping the model distinguish between challenging examples and may just be noticing that the final sentence in a given context is incorrect rather than developing meaningful comprehension of the text.

To better understand if this may be the case we also perform a similar training process but switching the SQuAD and Adversarial SQuAD datasets. So first strictly train on the adversarial and then set a baseline for how the model performs on the SQuAD set. Then we incorporate examples from this original set into the adversarial set to see what this fine-tuning training method truly understands from the text.

3 Results

3.1 Baseline Model Scores and Mistakes

When the SQuAD trained model is evaluated on the Adversarial SQuAD dataset, we see an EM score of 54.69 and an F1 score of 61.04. In comparison to when we evaluate on SQuAD examples, which has an EM score of 78.45 and an F1 score of 86.28, there is a significant drop in performance. The AddSent dataset takes SQuAD examples and adds one sentence to the end of the Wikipedia text. If only one sentence causes this impactful drop in performance, we wanted to analyze what characteristics of the adversarial example causes this drop in accuracy.

When taking a look at the types of mistakes that our trained model makes on adversarial examples, the average number of words that differ between the question and the adversarial sentence is about 2 words. This confirms why these adversarial examples are so challenging for the model and indicates that the model may not have a meaningful understanding of the text and rather determines an answer to a question based on the occurrences of keywords. Such occurrences were seen in examples to questions such as “Where did Super Bowl 50 take place?” The context of interest that the model selected was “The Champ Bowl 40 took place in Chicago” and as a result answered “Chicago.” We predict that the model prioritized the word “Bowl” as an indicator for the answer in conjunction with “took place” which led to an incorrect prediction. A stronger example of this is the question: “How long may the Amazon rainforest be threatened, according to some computer models?” The model answered “very soon” rather than “through the 21st century” using the adversarial sentence “The Amazon rainforest will be threatened very soon, according to some computer models.” As the predicted answer does not even make sense in the context of the question, this again indicates that rather than understanding the meaning behind the question and text, the model seems to be matching key words to form its prediction.

Furthermore, the model seems to take into account the length of the question provided and looks for an answer with a similar length. We found that the average length of a question was abel 11 words and the average length of adversarial sentences was also approximately 11 words. This is another indication that the model is looking for similar sentence structures rather than the actual meaning behind the words in use.

We will follow a similar method as mentioned in (Liu et al.2019) which uses a few challenging examples for training. Our method adds in these challenging examples from the adversarial set to

the SQuAD set for training with the hopes that if the model is exposed to even some of these examples it will not rely on finding keywords or similar sentence lengths in order to make predictions for question answering.

3.2 Training

Here, we display the results from using the fine-tuning method in Table 1 which shows the relation between EM Score and the number of challenge examples.

Challenge Examples	Exact Match
0	54.69101124
10	56.17977528
50	57.19101124
100	57.16292135
400	62.41573034
500	62.44382022
750	69.57865169
1000	78.65657521

Table 1: Average EM on Adversarial SQuAD

In general, the average EM score increases as we include an increasing number of challenging examples from the adversarial dataset to the SQuAD dataset for training. Additionally, the increase in score seems to correlate based on the magnitude of increase in the number of examples since the most significant increase in EM comes when we include 400 examples to our training set and once again when we reach 1000 examples. In the end, we increased the number of exact matches by approximately 24 points which is significant. When we only include a small number of examples such going from 0 to 10 or even 400 to 500, we see a negligible increase of up to one point.

The results for the exact match scores coincide with our prediction and with the results from (Liu et al.2019) that including challenging examples in the training loop can help the model deal with more challenging examples that closely match the question.

Next, when taking a look at the effect of this training process on F1 scores, we display the results in Table 2 which carries the same information as our previous graphics.

Challenge Examples	F1 Score
0	61.04371383
10	62.68749162
50	63.70663983
100	63.50396606
400	68.25372556
500	68.46044379
750	74.84675698
1000	86.27946885

Table 2: Average F1 on Adversarial SQuAD

Once again, we see a similar trend in F1 scores compared to the EM scores. As we increase the number of challenging examples used for training, the model performs better once evaluated on Adversarial SQuAD. The most significant increase in F1 is also seen at the 400 example mark and the scores only increase marginally in the samples below that. Furthemore, the largest jump is at the 1000 examples mark which sees a roughly 11 point jump in performance compared to 750 examples. So overall, including many challenging examples seems to benefit the model's training loop and allows it to make better predictions when faced with adversarial examples.

While the overall number of mistakes continues to decrease as we include more challenging examples into the training set, the types of mistakes made seem to change as well. In Table 3 below, we see the percentage of mistakes that occurred due to the AddSent sentences in the challenging examples based on the number of challenging examples used in training

Challenge Examples	Mistakes due to AddSent
0	75.36%
10	74.47%
50	72.69%
100	72.81%
400	71.85%
500	670.65%
750	66.56%
1000	64.27%

Table 3: Percentage of Mistakes due to AddSent

Here we see that in addition to an overall decrease in the number of mistakes, the percentage of mistakes caused by the adversarial sentences decrease as well. So as we introduce more challenging examples, the model becomes less prone to mistaking the adversarial sentence for the correct answer. A similar trend is seen here as with the EM and F1 scores. There is a large decrease in these types of mistakes once we reach 400/500 examples and once we reach 1000 examples, we see

an almost 10 percent decrease from the baseline percentage.

We further investigated how this type of training may affect performance on the original SQuAD dataset if the model is exposed to mostly adversarial examples before evaluation. The EM scores and F1 scores are displayed in Table 4 and Table 5, respectively.

SQuAD Examples	Exact Match
0	39.97161779
1000	45.14664144
1500	52.00567644
2000	54.95742668

Table 4: Average EM on SQuAD

SQuAD Examples	F1 Score
0	49.54631321
1000	55.01127848
1500	62.34820427
2000	64.7921708

Table 5: Average F1 on SQuAD

Here we see that when the model is originally trained with Adversarial SQuAD we have an EM score of approximately 40 and an F1 of 49.5. Once we incorporate original non-adversarial examples into the training set, the scores improve to an EM of 55 and F1 of 65 with 2000 examples. Overall, there is an increase in performance with this fine-tuning method regardless of if we fine tune with adversarial or non-adversarial.

Conclusions

Overall, the fine-tuning technique that we explored provided beneficial results for the Question Answering task. We saw up to a 24 point increase in EM scores and 25 point increase in F1 scores when introducing adversarial examples alongside the original SQuAD dataset for training our model. Furthemore, the percentage of the mistakes made by our model due to adversarial examples decreased alongside the total number of mistakes made. All of this indicates that this method of training does help the model's predictions and limits its susceptibility to examples which are created to intentionally challenge the model's understanding of the text. Looking at the results from originally training on Adversarial SQuAD, then introducing SQuAD examples, and finally seeing how the model performs on SQuAD, shows us that this technique does lead to an overall improvement in performance for question answering.

However, we do see some potential limitations from this experiment in that SQuAD and Adversarial SQuAD were the only two datasets utilized. This may lead to the model experiencing a bias for similar sentence structures. Thus, this model may still struggle on novel question answering datasets, and utilizing more training sets for fine-tuning could further improve performance on various datasets. Future investigation could improve upon this method, by introducing examples from multiple different adversarial datasets into the fine-tuning process. Furthermore, incorporating adversarial examples that vary in sentence structure, placement, and multiple sentences would help provide deeper insight into this method.

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lo-pyrev, 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, November. Association for Computational Linguistics.

[2] Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2171–2179, Minneapolis, Minnesota, June. Association for Computational Linguistics.

[3] 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, September. Association for Computational Linguistics