Fine-Tuning ELECTRA Through Hand Crafted Inoculation

Author
{Anonymous Solo Author}

Abstract

In this paper, we set out to improve a baseline ELECTRA model's performance on the SQuAD, Adversarial SQuAD and Natural Questions datasets through fine-tuned training. By identifying the types of questions the baseline model struggles with, we manually created a collection of labeled questions designed to teach the model to answer question frameworks it currently struggles to solve. In doing so, we find that our work typically causes more harm than benefit to performance, likely due to the original training problem overfitting on dataset artifacts. Our best approach decreased exact-match accuracy on SQuAD by 1.9 percentage points and 0.4 percentage points on Natural Questions, despite improving on the Adversarial SQuAD challenge set by 0.6 percentage points.

1 Introduction

Training a Question-Answer or QA model against a specific training dataset may result in strong performance on the task at-hand, but fail to truly understand the questions being asked. Instead, the model may learn spurious correlations that happen to result in success during training then fail when exposed to new questions. Known as dataset artifacts, these correlations can cause significantly degraded performance when testing a model on new information.

The effects of these artifacts have been exposed by introducing questions that are designed to fool a system through subverting expectations on a dataset with the same contexts (Jia and Liang, 2017) or datasets that require reasoning through multiple steps of logic instead of matching strings in the selected text (Yang et al.2018).

In our paper, we set out to expand on an alternative solution to dealing with dataset artifacts. Rather than transform the training task and data associated, our approach follows the inoculation method (Liu et al.2019) of fine-tuning our model after the initial training on examples that may mitigate the learning of dataset artifacts. By identifying the types of questions our baseline model struggles to answer, we attempt to feed it examples that teach it how to handle those situations, with the goal of improving performance both on the validation set of the initial training task as well as the external collection of adversarial examples.

2 Implementation

2.1 Baseline Model

In this paper, the baseline model refers to an ELECTRA-small model, as described in (Clark et al.2020). This transformer model was then run through three iterations of the SQuAD dataset to generate a trained model that serves as our baseline for understanding potential performance improvements.

2.2 Inoculation Method Applied

In (Liu et al.2019), it is shown that model performance can be improved by exposing a baseline model to a few hundred data points in a training setting, resulting in increased performance during evaluation settings. This paper sets out to replicate and expand on this by designing samples of additional data-points that result in performance improvements for our baseline model design above. The baseline model receives three iterations of fine-tuning

Figure 1:

Test Case	Example
Profession vs. Nationality	C: Helen is a DJ and Indonesian. Q: What is Helen's nationality?

with various collections of additional data curated for the experiment.

2.3 Stress Testing with CheckList Behavioral Test Suites

In (Ribeiro et al.2020), a methodology known as CheckLIST was proposed to challenge a model with test cases designed to grade specific functionality in predictions, much like unit-testing in software development. To understand our baseline model performance at a deeper level, we implement the recommended testing from their released codebase[1] against our model predictions.

In doing so, we learn that our baseline model struggles significantly with certain types of questions. Figure 1 outlines a selection of test cases and their results.

Of the categories, there are three classes of question prompts that the model fails to produce a single correct answer during testing: reasoning through an antonym of the context in the question, logically comparing the two subjects of the context, and ranking intensifier adjectives against each other.

Additionally, while the model can produce non-zero correct responses, the baseline model also struggled significantly with distinguishing classes of words within each other, as it often fails to disambiguate animals and vehicles, as well as nationalities and professions.

Overall, the test cases reveal that the exact-match from our accuracy, benchmarked in the Figure 2, is driven mostly by string-matching the question received against the context studied, and finding the answer within the scope of that matching, rather than truly understanding the context provided.

[1] https://github.com/marcotcr/checklist

Using the takeaways from the CheckLIST testing, we can build some filtering into our baseline model performance on the SQuAD validation data set to find similar patterns across the questions asked in this dataset. These results are found in Figure 2.

Figure 2:

Descriptor Case	**Acc (%)**
Entire Validation Dataset Benchmark	78.8
Question contains the word “Who”	83.8
Question contains the word “Which”	77.8
Question contains the word “Why”	58.7
Answer found in the first 25% of the context	80.8
Answer found in the last 25% of the context	77.3
Question contains negation with the word “not”	74.1

The overall performance on the SQuAD validation set had 78.8% exact-match accuracy for the predictions returned from the QA model. Additionally, the model was even stronger on questions with the keyword “who” in it, implying that questions requiring the model to scan the context for a name found within the selected text and return it are straightforward.

However, the question keywords “which” and “why” resulted in exact-match accuracy lower than the overall average. These questions often require logical reasoning, or comparisons of subjects, to understand which answer is more correct, something we have seen from the

CheckLIST testing that the baseline struggles with. For example, the validation dataset has a question asking “Why was [Nikola Tesla] unable to enroll at the university?”. The predicted answer included a list of information about Tesla from the context, the correct answer being found within that list. However, the model did not fully understand the contents of the list, and returned a collection of information related to Tesla, but unrelated to his academic status, implying a lack of understanding within the selected text.

Additionally, one thing found in the CheckLIST testing was that, when presented with a question and context that had two subjects to choose from, such as “Joe is faster than Mike. Who is less fast?” the model tended to return the first name in the list, regardless of the question. This phenomena is likely found in our evaluation results as well, as our accuracy is higher when the index of the answer is found in the first 25% of the context, compared to the final 25%.

Lastly, a final signal of limited true comprehension is that questions containing “not” were answered correctly at a lower rate than the average, meaning that if negation requires logical reasoning, the model could not perform at the same level.

With more of an understanding of the kinds of questions the baseline model can and cannot answer, we attempt to follow inoculation in an effort to improve the model in these specific scenarios. By curating additional data sources to train and fine-tune on, ideally we can teach the model how to handle these scenarios without fundamentally changing the model and losing performance on our benchmark SQuAD evaluation.

2.4 Additional Data Sources

To improve our model through the inoculation methodology as proposed requires additional data to fine-tune training with. The following sections describe the various supplemental data sources used for further training and evaluation:

2.4.1 Adversarial SQuAD Examples

In (Liu et al.2019), the methodology describes evaluation against the SQuAD Adversarial dataset described in (Jia and Liang, 2017). Designed to fool computer systems without changing the answer to questions by inserting additional sentences into the context, this dataset is included both as a source to evaluate performance against, as well as an option for additional training data to fine-tune with.

2.4.2 CheckLIST Test Cases Reformatted

In section 2.3, we discuss using the CheckLIST testing methodology to find challenges that the baseline model struggles to answer correctly. With the test cases written, our work includes transforming the records in the test suite into context, question and answer combinations that fit into the SQuAD data structure. By doing so, it allows us to construct an adversarial dataset as a potential option for fine-tuning the baseline model on particularly challenging examples.

2.4.3 SQuAD

The SQuAD, or Stanford Question Answering Dataset, as defined in (Rajpurkar et al.2016) is used to train the baseline model, as well as sampling questions to continue fine-tuning with. Additionally, it is used as a validation dataset to evaluate performance against.

2.4.4 Natural Questions

In addition to utilizing the above listed sources, work was done to include Google’s Natural Question dataset in fine-tuning the model and evaluating results. As described in (Kwiatkowsky et al.2019), the Natural Question dataset consists of questions asked by users to the Google search engine, and manually annotated answers. To generate a version of the

dataset that is compatible with SQuAD based QA pipelines, the code released alongside (Sen et al.2020) was used to reformat the original question set into the SQuAD format.[2] In doing so, we are able to leverage a QA dataset free of SQuAD-based dataset artifacts and synthetic test cases.

2.4.5 Manually Curated SQuAD Examples

In an effort to maintain some of the patterns learned in the SQuAD dataset, while still providing challenging examples to fine-tune our QA model, a final dataset incorporated into the approach was a collection of 30 hand-curated context, question and answer trios inspired by the SQuAD dataset. For these data points, the contexts were chosen from the previously existing contexts. However, the questions asked alongside the context were manually created to be more difficult. These questions often required the model to do multi-hop reasoning as inspired by (Yang et al.2018) or find the second or third number in the list rather than the first, something the CheckLIST revealed the model struggled to do. The intent of the curated examples was to provide data points to teach the model concepts it was known to struggle with, while potentially maintaining performance on the original SQuAD dataset by reusing its context.

3 Results

The results of our fine-tuning can be found in the table Figure 3. Across all iterations of additional data provided, the resultant performance was “Outcome 3” as described in (Liu et al.2019): even in the cases where our model performance improved on Adversarial SQuAD, it came at the expense of performance on the original SQuAD task.

Importantly, a result defined in (Liu et al.2019) has been confirmed: fine-tuning on Adversarial SQuAD data points improved performance on the adversarial challenge set, but came at the expense of performance on the original task, implying our methodology is sound..

Figure 3:

Model Iteration	SQuAD		
	Acc	F1	
Baseline	78.8	86.3	
+Adversarial	76.7	84.5	
+SQuAD	77.0	85.0	
+CheckLIST	77.6	85.2	
+Manually Created	78.2	86.1	
+SQuAD, CheckLIST, Manual	76.9	85.2	
+NQ	73.0	82.0	

*Note: due to time constraints I only converted the validation set for NQ, so training and evaluatio
practice, but I wanted to try this iteration and was pressed for time due to class deadlines.

Additionally, an interesting result shows that individually fine-tuning on additional SQuAD data, our CheckLIST tests, or the manually curated SQuAD questions resulted in decreased performance across all three evaluation sets. However, combining the three samples into one dataset to fine-tune with actually resulted in improved performance on the adversarial set while maintaining the negative performance impact on the other two.

Revisiting our analysis from the baseline model shows that despite including CheckLIST samples that were intended to teach the model to deal with very specific functionality tests, the performance on these tests decreased. These results can be seen in Figure 4.

Similarly, extending the types of questions from the CheckLIST tests into SQuAD questions that

---

2 https://github.com/amazon-science/qa-dataset-converter

require similar reasoning, the performance again decreases across the board, as displayed in Figure 5.

Figure 4:

Descriptor Case	**Baseline Acc (%)**	**+SQuAD, CheckLIST, Manual SQuAD (%)**
Profession vs. Nationality	64	61.6
Animals vs. Vehicles	28	20
Antonyms	0	0
Comparisons	0	0
Intensifiers	0	0

Figure 5:

Test Case	**Baseline Acc (%)**	**+SQuAD, CheckLIST, Manual SQuAD (%)**
Entire Validation Dataset Baseline	78.8	76.8
Question contains the word “Who”	83.8	82.6
Question contains the word “Which”	77.8	74.7
Question contains the word “Why”	58.7	54
Answer found in the first 25% of the context	80.8	79.3
Answer found in the last 25% of the context	77.3	75.2
Question contains negation with the word “not”	74.1	70.7

I believe these results confirm the power of dataset artifacts in the original training problem, resulting in a model overfit on the specific dataset it was trained on.

The inclusion of the CheckLIST examples, manual SQuAD inspired challenge questions and SQuAD examples from the initial training was supposed to provide the new tuned model with data points it could not yet solve alongside questions it knew how to answer to reinforce those patterns the model had already learned. Despite that, inclusion of the trio resulted in a performance on the SQuAD dataset that was lesser than providing each of the three individually. I hypothesize for further experimentation that introducing new data points caused the model to relax some of the overfit on the SQuAD dataset, resulting in limited performance on the original evaluation set.

In the case of feeding the model only additional SQuAD information, the selection provided for fine-tuning the model likely resulted in the model overfitting even further on the selected sample.

However, by loosening the grip of the overfit SQuAD examples with the inclusion of our CheckLIST questions, the model was able to perform better on the Adversarial SQuAD set by not conforming so specifically to SQuAD artifacts that the adversarial examples were designed to subvert.

With the inclusion of the Natural Questions dataset, however, we can verify that our improved Adversarial SQuAD performance does not translate to a model that approaches improved generalization, as the accuracy decreased for this entirely out-of-sample dataset.

4 Conclusion

Our model saw a decrease in accuracy for the SQuAD validation task across all iterations of fine-tuning, dropping by as much as 5.8 percentage points when tuning on Natural Question data. Despite that, our best inoculation method without including Adversarial SQuAD data resulted in 0.6 percentage point improvement on exact-match accuracy and 0.5 point improvement on F1 accuracy when evaluating against that dataset. These results imply that inoculation in the presence of dataset artifacts is difficult to maintain performance on the original task, however fine-tuning QA models on a small set of data can result in accuracy and comprehension improvements. To generate positive results, the curation of the dataset that drives additional training is a critical factor in whether or not the additional training results in general improvements.

5 Acknowledgements

I'd like to thank Professor Durrett and the entire course staff for a fun, engaging and informative class at a seemingly critical time for NLP research.

References

[Clark et al.2020] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.

[Jia and Liang2017] 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.

[Kwiatkowski et al.2019] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein et al. 2019. Natural questions: a benchmark for question answering research. In Transactions of the Association for Computational Linguistics 7, pages 453-466.

[Liu et al.2019] 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.

[Rajpurkar et al.2016] 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, November. Association for Computational Linguistics.

[Ribeiro et al.2020] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4902–4912, Online, July. Association for Computational Linguistics.

[Sen et al.2020] Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets?. arXiv preprint arXiv:2004.03490.

[Yang et al.2018] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2369–2380, Brussels, Belgium, October-November. Association for Computational Linguistics.

[Zhou and Bansal2020] Xiang Zhou and Mohit Bansal. 2020. Towards robustifying NLI models against lexical dataset biases. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8759– 8771, Online, July. Association for Computational Linguistics.