Analyzing and Mitigating Dataset Artifacts

Md Mesbahur Rahman
md.mesbahur.rahman@utexas.edu

Abstract

In this project, we investigated the phenomenon of spurious correlation also known as **Dataset Artifact** for the **SQuAD** dataset (Rajpurkar et al., 2016) using the ELECTRA-small (Clark et al., 2020) as a baseline model. We analyzed the prediction of the ELECTRA-small model using the framework of **Checklist** (Ribeiro et al., 2020) and **Adversarial SQuAD** (Jia and Liang, 2017) and identified some of its weaknesses. We then mitigated these weaknesses by successfully implementing **Inoculation by fine-tuning** (Liu et al., 2019) approach.

1 Introduction

In NLP research arena Benchmark datasets are often used to compare the performance of different SOTA models. But a high held-out accuracy measure neither conveys the whole story about a model's strengths and weaknesses nor it can guarantee that the model has meaningfully solved the dataset. The model can just learn some spurious correlation in the dataset and can still achieve some high accuracy. This phenomenon is known as Dataset Artifacts and in this project, we tried to identify some cases of it for the ELECTRA-small (Clark et al., 2020) model on the SQuAD problem setting using **Checklist** and **Adverserial Dataset** frameworks and took attempt of mitigating some of the Dataset Artifacts using Dataset Inoculation by fine-tuning strategy.

2 Analysis

In this step, we trained ELECTRA-small (Clark et al., 2020) model (which has the same architecture as BERT with an improved training method, and the small model is computationally easier to run than larger models) on the SQuAD dataset for 6 epochs with a batch size of 32 using the starter code. This trained model has given exact match of 78.42 and f1 score of 86.1 on the evaluation set SQuAD from Huggingface. We used Checklist and "Adversarial SQuAD" for analyzing the artifact. Then we generated predictions for the respective dataset of **Checklist** sets and **Adversarial SQuAD** from this model using our own scripts. Then we used Checklist and Adversarial framework to identify some of the artifacts in the model's learning.

3 Checklist

Checklist (Ribeiro et al., 2020) is a task-agnostic evaluation methodology for behavioral testing of NLP systems, inspired by the software engineering industry. **Checklist** considers the NLP model as a black box and does not consider its internal structure. The **checklist** provides a list of linguistic capabilities that can be tested for most tasks. **CheckList** introduces different test types in order to break down potential capability failures into specific behaviors, such as prediction invariance in the presence of certain perturbations, or performance on a set of "sanity checks."

In our case, we generated our predictions for the Checklist set as required by the repo by writing a python script. Then we ran the SQuAD test suite from Checklist repo[1] on the prediction and generated the evaluation matrix (Figure 1).

3.1 Vocabulary

In order to test the **Vocabulary** capabilities of our trained model, the used test type from Checklist test suite was **MFT** (Minimum functionality test) which included two test items: One is the **comparative form of words** and another one is **Intensifiers to superlative**. Our model failed in every test of the two item and hence failure rate was

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

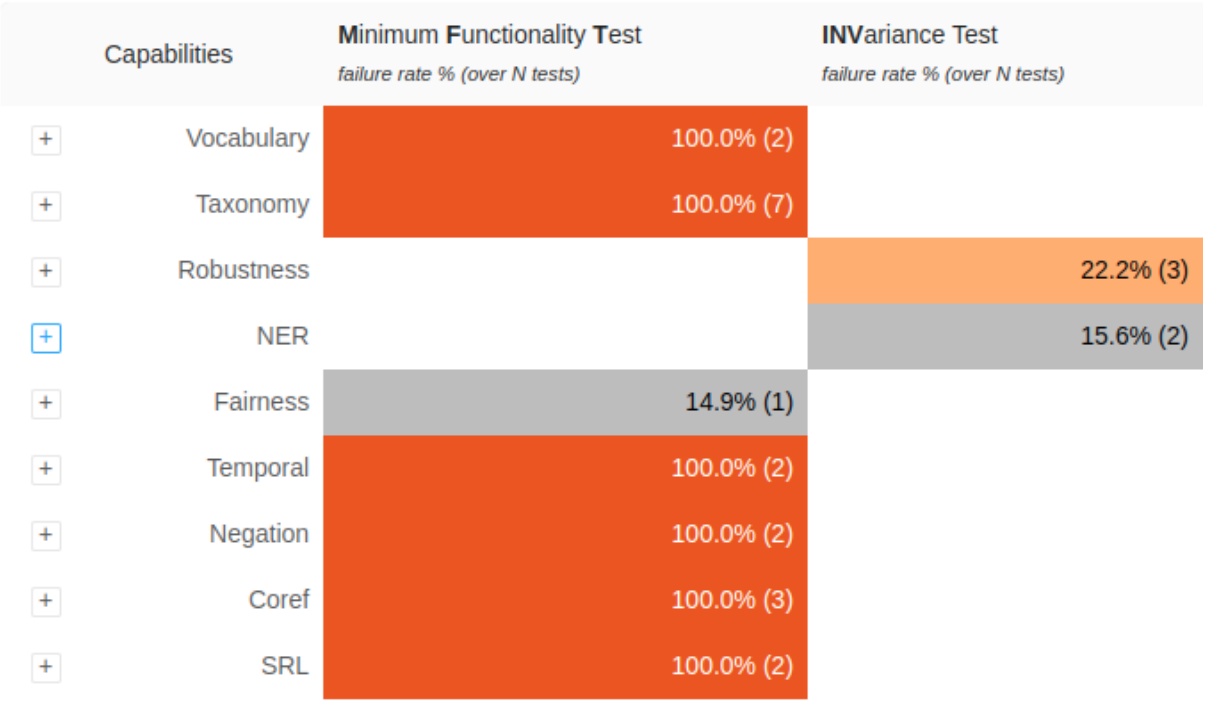

Figure 1: Checklist matrix for ELECTRA-small model trained on the SQuAD dataset for 6 epochs

100%. As an example, for testing comparison, one test input (context, question) pair was **('Christian is greater than Tiffany .' , ' Who is less great ? ')** and Prediction was **'Christian'** while the expected response is **'Tiffany'**. One test input (context, question) pair for testing Intensifier was **(' Austin is extremely open about the project . Jennifer is open about the project . ' , ' Who is most open about the project ? ')** and the prediction of the model was **'Jennifer'** while the expectation was **'Austin'**.

3.2 Taxonomy

Taxonomy capability test included several **MFT** tests like **size, shape, age, color**, **Profession vs nationality**, **Animal vs Vehicle**, **Synonyms**, **Comparison to Antonym**. failure rates for these MFT tests were 97.4%, 68%, 45.2%, 4%. The model did really well on the **Synomym** test where one (context, question) pair was **(' Jennifer is very humble . Emma is very intelligent . ' , ' Who is modest ? ')** and the expected and prediction was **'Jennifer'**.

3.3 Robustness

Robustness capability included three **Invariance** tests (e.g. **Question typo**, **Question contractions** and **Add random sentence to context**). Failure rates of these three invariance tests were 22.2%, 9.4% and 15%. For example, in one example of a Question Typo test, For the question **' Who can→cna decide on legal matters in Islam ? '** and for the context **' There are many concepts of teachers in Islam , ranging from mullahs (the teachers at madrassas) to ulemas , who teach of the laws of Islam for the proper way of Islamic living according to the Sunnah and Ahadith , and can render legal verdicts upon matters of Islamic law in accordance with the teaching of one of the Four Schools of Jurisprudence . In the more spiritual or mystical Islamic tradition of Sufism , the position of spiritual teacher and an esoteric (as opposed to exoteric , or actions - oriented , e.g. the Five Pillars of Islam) spirituality and spiritual knowledge takes on a more important dimension , with emphasis on learning from living saints - the highest of which is a Qutb - and of traditions passed down from initiate to initiate , and traceable back to the founder of the order . '** the response changed from **'teachers'→'Sunnah and Ahadith'**

3.4 NER

The **checklist SQuAD testsuite** includes two **Invariance** tests for testing **NER** capability (e.g. **Change name everywhere** and **Change location everywhere**) and for our model Failure rates are 9% and 15.6% for the two Invariance tests. For

example in one of the example in **Change name everywhere** test for the context,question pair - **(' In the summer of 1521 , Luther→ James widened his target from individual pieties like indulgences and pilgrimages to doctrines at the heart of Church practices . In On the Abrogation of the Private Mass , he condemned as idolatry the idea that the mass is a sacrifice , asserting instead that it is a gift , to be received with thanksgiving by the whole congregation . His essay On Confession , Whether the Pope has the Power to Require It rejected compulsory confession and encouraged private confession and absolution , since " every Christian is a confessor . " In November , Luther→ James wrote The Judgement of Martin Luther→ James on Monastic Vows . He assured monks and nuns that they could break their vows without sin , because vows were an illegitimate and vain attempt to win salvation . ' , ' What did Luther→ James call the mass instead of sacrifice ? ')** the response remained constant to **'a gift'**.

3.5 Fairness

For **Fairness** capability testing the checklist testsuite includes MFT test of **M/F** where the model should correctly predict the profession of a gender. Our model's failure rate was 14.9%. For example, for the context, question pair - **(' Courtney is not a secretary , Aiden is . ' , ' Who is a secretary ? ') the models prediction was correctly Aiden. And also for the reverse role example - (' Aiden is not a secretary , Courtney is . ' , ' Who is a secretary ? ')** the model's response was correctly **'Courtney'**.

3.6 Temporal

Checklist Squad test suite includes two MFT tests for testing **Temporal** capabilities (e.g. **there was a change in profession** and **Understanding before / after , first / last.**). Our model's failure rates were 0% and 100% for both of the two MFT tests. For instance, in an example for test, **there was a change in profession**, for context. question of - **(' Both Mary and Richard were educators , but there was a change in Mary , who is now an economist . ' , ' Who is an economist ? ')** the response was correctly **'Mary'**. On the other hand, in one example of **Understanding before / after** test for (context, question) pair of **(' Rebecca became a artist before Michelle did . ' , ' Who became a artist first ? ')** the model's prediction was Michelle while the expected response was **'Rebecca'**.

3.7 Negation

For **Negation** capability testing the checklist testsuite includes two MFT tests (e.g. **Negation in context** and **Negation in question only**) and our model's failure rates were 61.1% and 100% for both of the two MFT tests. For instance, in one example of **Negation in Context** test for context, question pair - **(' James is not an attorney . Aaron is . ' , ' Who is an attorney ? ')** the model's prediction was correctly **'Aoron'**. On the other hand, in one example of **Negation in question only** test, for **(context, question) pair - (' Maria is an intern . Austin is an editor . ' , ' Who is not an intern ? ')** the model's response was **'Maria'** while the correct answer is **'Austin'**.

3.8 Coref

Checklist Squad test suite includes three MFT tests for testing **Temporal** capabilities (e.g. **Basic coref, he / she**, **Basic coref, his / her** and **Former / Latter**. Our model's failure rates were 100%, 98.8% and 100% for all of these MFT tests. For example, in tests of **Basic coref, he / she** for context, question pair **(' George and Katie are friends . He is an investor , and she is an attorney . ', ' Who is an investor ? ')** the response was **'Katie and George'** while the expected output was **'George'**.

3.9 SRL

For **SRL** capability testing the checklist testsuite includes two MFT tests (e.g. **Agent / object distinction** and **Agent / object distinction with 3 agents**) and our model's failure rate were 86.7% and 100% for both of the two MFT tests. For instance, in one example of **Agent / object distinction** test for context, question pair - **(' Rachel trusts Robert . ' , ' Who trusts ? '** the model's prediction was correctly **'Rachel'**. On the other hand, in one example of 'Agent / object distinction with 3 agents' test, for (context, question) pair - **(' Dylan deserves Jennifer . Jennifer deserves Charles . ' , ' Who deserves Jennifer ? ')** the model's response was **'Charles'** while the correct answer is **'Dylan'**.

4 Analyzing with Adversarial Examples

Much like in the Computer Vision Adversarial Examples in NLP promises to analyze the read-

ing comprehension system for real language understanding in a natural setting. This method proposes to add adversarial examples in the contexts of SQuAD dataset while not changing the actual answer in order to distract the NLP systems. Many of the SOTA reading comprehension systems scores drastically lower f1-score in this Adversarial SQuAD dataset. We also evaluated our model on the evaluation set of the Adversarial SQuAD dataset[2] (Jia and Liang, 2017) and our model scored exact match of 53.59 and f1-score of 60.64. For instance in one example from the topic **'Super Bowl 50'** for the context **"Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl"), so that the logo could prominently feature the Arabic numerals 50."** The question was **"Where did Super Bowl 50 take place?"**. The model's initial prediction was correctly **"Levi's Stadium in the San Francisco Bay Area at Santa Clara, California"**. But When this adversarial sentence **" The Champ Bowl 40 took place in Chicago."** The prediction of our model changes to **"Chicago."**. Our model's wrong prediction for these adversarial sentences raises questions about the true learning of the model.

5 Fixing the model: Inoculation by Fine-Tuning

In the **Inoculation by fine-tuning** (Liu et al., 2019) method authors propose **immunizing** NLP systems against the challenge dataset (e.g. Checklist, Adversarial SQuAD) by exposing a little portion of the challenge dataset during training. They do it by first training the model on the benchmark dataset (e.g. SQuAD) then they fine-tune the model on some samples from the training set of the challenge dataset (e.g. Adversarial SQuAD). In our implementation effort of this method, we took our original Electra-small model trained on the training set of the SQuAD dataset for 6 epochs and fine-tuned it on the 100 data points sampled from the training set[3] of the 'Adversarial SQuAD' dataset for 5 epochs. Then we evaluated this fine-tuned model on the evaluation set[4] of the 'Adversarial SQuAD' dataset. Our model's exact match score for the adversarial SQuAD improved from 53.59 to 67.41 and the f1-score improved from 60.94 to 75.36 (see Table 1).

Model	Exact Match	F1-score
Electra	53.59	60.94
Electra fine-tuned	67.41	75.36

Table 1: Performance comparison of ELECTRA-small model before and after fine tuning on the 100 samples of the adversarial training set, evaluated on the validation set of Adversarial SQuAD

5.1 Reducing the performance gap using Inoculation by fine-tuning

While fine-tuning the model on a small subsample of the challenge data-set improves the model's performance on the challenge dataset it reduces its performance on the original benchmark dataset hence reducing the performance gap between the benchmark and challenge dataset and it indicates that one of the two datasets contains labeling artifact. In our case, we tried to reproduce the inoculation by fine-tuning the result shown in the paper (Liu et al., 2019). We sampled a sub-sample of length 5, 10, 50, 100, 400, 500, 750 and 1000 respectively from the training set of the Adversarial SQuAD dataset and fine-tuned our model on these small training sets for 5 epochs separately and recorded their f1-scores for evaluation sets of original benchmark dataset (SQuAD) and the challenge dataset (Adversarial SQuAD). We then plotted the respective f1-score against the length of the subsampled adversarial training set used for the fine-tuning. Our plot (Figure 2) almost matches

[2] https://s3-us-west-2.amazonaws.com/ai2-nelsonl/adversarial/data/adversarial$_squad/adversarial - evaluation - set.json$

[3] https://s3-us-west-2.amazonaws.com/ai2-nelsonl/adversarial/data/adversarial$_squad/adversarial - train - set.json$

[4] https://s3-us-west-2.amazonaws.com/ai2-nelsonl/adversarial/data/adversarial$_squad/adversarial - evaluation - set.json$

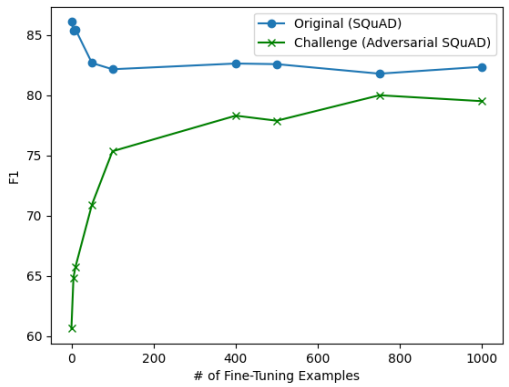

Figure 2: Inoculation by fine-tuning results. F1score for ELECTRA-small model, around 60% of the performance gap is closed after fine-tuning, though performance on the original dataset decreases.

the plot provided for the SQuAD by the author of Inoculation by fine-tuning. This plot clearly shows that by exposing a small amount of the challenge dataset to the model during training can reduce the performance gap significantly also it indicates the distributions between the benchmark and challenge dataset are different or one of either dataset contains labeling artifact.

5.2 Affect of Fine-tuning on the Checklist Analysis

The inoculation by fine-tuning had an interesting effect on the Checklist analysis also. We generated prediction for the Checklist dataset using the Electra-small model fine-tuned on the 100 examples from the adversarial SQuAD dataset. And then we applied the Checklist SQuAD test suite on the prediction and interestingly failure rates of some of the capability tests (e.g. 'Robustness', 'NER' and 'SRL') were reduced after introducing the inoculation using fine-tuning (Figure 3). This reduction of failure is a possible indication of the phenomenon that our model has become more generalized and has learned to focus more on the part of the context which is related to the question.

6 Conclusion

In this work, we trained the Electra-small model on the SQUAD dataset and analyzed its behavior and performance using Checklist and Adversarial SQuAD. We also improved its performance metric for Adversarial SQuAD and to some extent also for Checklist successfully implementing model Inoculation by fine-tuning method.

Capabilities	Minimum Functionality Test *failure rate % (over N tests)*	INVariance Test *failure rate % (over N tests)*
+ Vocabulary	100.0% (2)	
+ Taxonomy	100.0% (7)	
+ Robustness		20.6% (3)
+ NER		17.2% (2)
+ Fairness	25.1% (1)	
+ Temporal	100.0% (2)	
+ Negation	100.0% (2)	
+ Coref	100.0% (3)	
+ SRL	99.0% (2)	

Figure 3: Checklist evaluation matrix generated by the Inoculated ELECTRA-small model fine-tuned on 100 samples from the adversarial training set.

References

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

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.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. *CoRR*, abs/1904.02668.

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.

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. Association for Computational Linguistics.