Naughty or Nice? CheckList Twice. ELECTRA's Coming to Town.

Amann Islam, Caleb Bucci
University of Texas at Austin
{amannislam, cbucci}@utexas.edu

1 Abstract

ELECTRA is an NLP model from Google and Stanford that diverges from the mask-language-models by discriminating tokens rather than generating masked words in context. This paper focuses on improving the performance on QA tasks for ELECTRA trained with SQuAD datasets, and evaluated against CheckList [5], a fine-grained category test set. Discriminative models historically do not perform on questions requiring sentence relative sentence comprehension and SQuAD's example distribution particularly has limited exposure in Question-Answering (QA). The base model was fine-tuned with an 80/20 split of SQuAD and CheckList data to increase QA examples. The new model performance vastly improved against the CheckList evaluation while retaining the same in-domain performance.

2 Introduction

Transformers have become the standard backbone for extractive Question-Answering (QA) tasks, and ELECTRA is among the most efficient of these architectures. ELECTRA was able to achieve state of the art performance during its time on various types of datasets, such as natural language inference (NLI), QA, and sequence tagging. Although ELECTRA-small fine-tuned on SQuAD achieves competitive dev-set performance[1, 4], this can mask deeper model brittleness.

CheckList provides a way to expose this brittleness by organizing model behaviors into fine-grained capability categories (Table 1), and constructing targeted test suites for each. Previous work with CheckList has shown that strong model performance is *not* an indicator of model performance on fine-grained categories, and the model can systematically fail on simpler linguistic patterns such as negation and text perturbations[5]. We find the same pattern for ELECTRA-small. Despite strong SQuAD performance, our baseline model systematically fails on comparative questions and name-based reasoning, and it shows high error rates across several CheckList categories.

In this work, we investigate whether fine-tuning can mitigate these failures without sacrificing overall performance on SQuAD. We do the following:

- Train an ELECTRA-small baseline on SQuAD and categorize its failures using CheckList and its fine-grained capability categories.
- Fine-tune the model using purely synthetic data generated from selected CheckList subcategories where the baseline fails most, adversarial examples from Adversarial SQuAD[2], and a combined 80/20 mixture of SQuAD and CheckList-generated examples, all using a LoRA adapter.
- Evaluate how each of these fine-tuning strategies performs in overall SQuAD performance versus CheckList failure rates.

Preprint. Under review.

3 Background and Data

3.1 ELECTRA and Extractive QA

ELECTRA[1] trains a discriminator to distinguish real input tokens vs fake input tokens generated by another neural network, which is similar to the discriminator of a Generative Adversarial Network (GAN). ELECTRA-small performs well on low compute power and still performs competitively on the SQuAD dataset, making it useful for our implementation.

3.2 Systematic Model Testing with CheckList

CheckList[5] is a test suite designed to test the capabilities of NLP models on various test types and linguistic phenomena. It provides us with the ability to test our model and find weaknesses that can be addressed in order to improve general model performance. In this work, we use CheckList both as an evaluation suite and as a generator of synthetic data examples for our fine-tuning.

Category	Short description
Vocabulary + POS	Sensitivity to lexical choice and part-of-speech changes.
Taxonomy	Reasoning over class hierarchies and *is-a* relationships.
Robustness	Stability under paraphrases, typos, and small input perturbations.
Named Entity Recognition	Handling and identifying people, locations, organizations, etc.
Temporal Understanding	Interpreting time expressions and ordering events correctly.
Negation	Correctly interpreting negative statements and reversals of meaning.
Coreference	Tracking entities across pronouns and referring expressions.
Semantic Role Labeling	Capturing who did what to whom, when, where, and how.

Table 1: Fine-grained CheckList capability categories considered in this work.

3.3 SQuAD

SQuAD[4], or the **S**tanford **Qu**estion **A**nswering **D**ataset, is a dataset designed specifically for extractive question answering models. It contains about 100k question-answer pairs split among the training and dev sets. This paper specifically uses the SQuAD 1.1.

3.4 Adversarial SQuAD

Adversarial SQuAD[2] is a dataset that uses examples from SQuAD and adds an automatically generated sentence to the context that doesn't change the gold label of the entry. This is used to distract the model and is helpful for testing model comprehension.

4 Approach

We initially captured a baseline performance of the ELECTRA-small model by training it on the SQuAD dataset using the HuggingFace Trainer. After obtaining our evaluation results on Extractive Question Answering (QA), we used the CheckList test suite to test our model against various fine-grained language categories[5] which can be seen in Table 1. The fine-grained category failures are seen in Table 4. where only NER and Robustness had acceptable performances.

The following adaptations were applied to improve performance:

- Generate synthetic test data from CheckList and fine-tune the base model.
- Fine-tune our model with a Adversarial SQuAD (AdvSQuAD) to improve model error rate[3].
- Combine some generated synthetic data using CheckList and data from SQuAD, and fine-tune our model on the combination of the two.

A LoRA adapter was used for fine-tuning with the following parameters:

- Learning Rate: 1e-5; Weight Decay: 0.01; Num Epochs: 1
- Lora Rank: 32; Lora Alpha: 128

4.1 Fine-Tuning on CheckList Generated Data

Category	Subcategories used for data generation
Vocabulary + POS	Comparison (less / more); Intensifiers / Reducers
Taxonomy	Size / shape / color; Jobs vs Nationalities; Animal vs Vehicle (v1/v2); Comparisons w/ simple and complex antonyms
Temporal Understanding	Change in profession; Before/After → First/Last
Negation	Negation in context; Negation not in context
Coreference	Job coreference; He/She coreference; Former/Latter coreference
Semantic Role Labeling	Agent/Object (3 agents)

Table 2: CheckList subcategories used to generate synthetic training data.

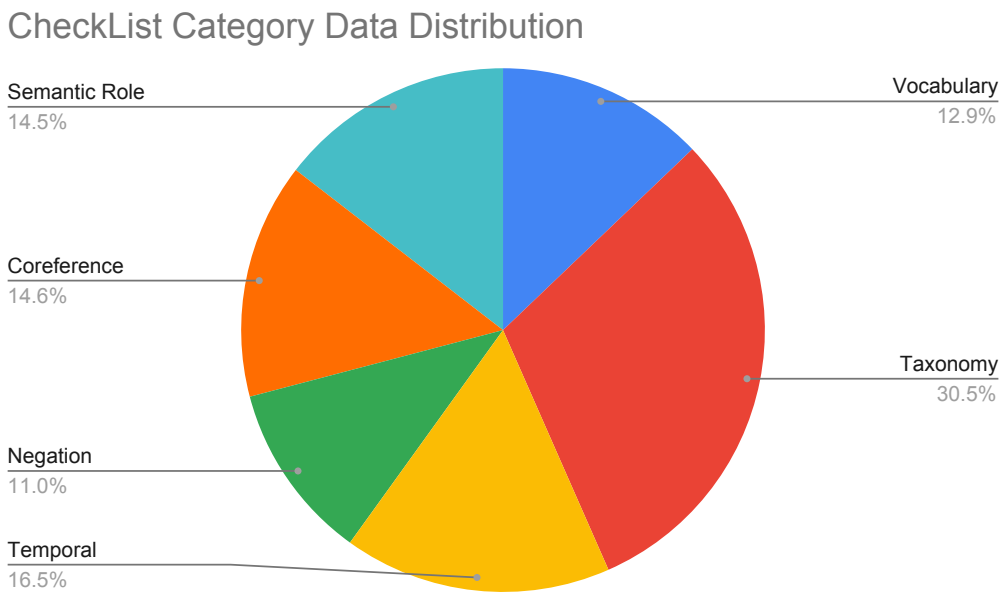

Figure 1: CheckList Category Data Distribution

We generated test cases for each subcategory shown in Table 4 using CheckList. After receiving the results for each test case, we selected a few of the subcategories that we then used to generate synthetic data. We selected these subcategories based on a failure threshold, so if a category performed well (like Robustness did), then it wasn't used to augment the data. These can also be seen in Table 4. The goal of this was to provide the model with data that can improve the accuracy of these individual categories, in hopes of improving overall performance. After retrieving the generated test cases, we then fine-tuned our baseline model on this data using the LoRA adapter described earlier.

4.2 Fine-Tuning on Adversarial SQuAD

We wanted to use another dataset to compare the differences in performance between fine-tuning on synthetic data designed specifically for our model's weaknesses, and adversarial data that is more general, but also of much higher quality. For this section, we trained our original dataset on SQuAD,

fine-tuned it on all challenge examples from Adversarial SQuAD with our LoRA adapter, and re-tested it on our evaluation set[3]. AdvSQuAD contained 5000 examples total. We ran the CheckList test suite on this fine-tuned model to see its performance on the fine-grained categories compared to the baseline.

4.3 Fine-Tuning on Combined CheckList/SQuAD Data

After comparing the AdvSquad performance with our baseline, we decided to create an augmented dataset. This dataset was going to combine the synthetic data designed for the model's weaknesses, with regular SQuAD examples in order to prevent overtraining on the 'fake' data. The examples selected were from categories above a self-selected threshold failure (15%). We chose a ratio of about 80% SQuAD and 20% CheckList. The exact ratio of data chosen from each category for this 20% can be seen in Figure 1. We formatted the synthetic data to match the formatting of SQuAD in order to maintain training consistency. 10000 Samples were generated from CheckList and then post-processed to remove all low quality or obviously incorrect answers. We then added the filtered 9174 examples and 40000 examples from SQuAD, for a total training size of 50000 examples. The baseline model was fine-tuned on this data using the LoRA adapter optimization above with one modification, we used 5 epochs instead of 1. Each checkpoint was analyzed and the best one was selected, epoch 3.9.

5 Results

Section	Context + Question	Answer	Prediction
Vocabulary Comparison	Kevin is cleaner than Alice. Who is less clean?	Alice	Kevin
Vocabulary Comparison	Peter is weaker than Andrew. Who is less weak?	Andrew	Peter
Coreference (His/Her)	Erica and Paul are friends.Her mom is an academic.Whose mom is an academic?	Erica	Paul
Coreference (His/Her)	Megan and Nathan are friends. Her mom is an accountant.	Megan	Megan and Nathan
Semantic Role Labeling	Anna is followed by Sarah. Anna follows Amanda. Who follows Amanda?	Anna	Sarah. Anna

Table 3: Baseline Model CheckList Failed Examples

Robustness Question	Prediction 1	Prediction 2
What's a connection identifier	packets include a connection identifier rather than address information	The packets include a connection identifier rather than address information
What do red algal chloroplasts have that green chloroplasts don't?	phycobilisomes	chlorophyll b

Table 4: Baseline Model CheckList Failed Examples for Robustness

Score / Dataset	SQuAD	Fine-Tuned w/AdvSQuAD	Fine-Tuned with 80/20 SQuAD/CheckList
EM	78.2592	78.0984	78.2592
F1	85.8674	85.7767	86.1644

Table 5: In-Domain Model Performance with Different Datasets

ELECTRA is a discriminative model and not a reasoning model, we expected relational questions to regularly under perform. This behavior compounds harder when working through CheckList's problem set which consists predominantly on comparative questions which requires multi-step reasoning. The base model was fine-tuned on three different datasets: AdvSquad, pure CheckList (PC), and a 80/20 ratio of SQuAD and CheckList (combined).

CheckList / Dataset	SQuAD	Fine-Tuned w/AdvSQuAD	Fine-Tuned with 80/20 SQuAD/CheckList
Vocab + POS	99.9	99.9	23.2
Taxonomy	56.98	70.457	29.9
Robustness	13.47	21.93	23.3
Named Entity Recognition (NER)	6.6	13.1	11.6
Temporal Understanding	62.2	50	49.2
Negation	83.75	97.8	49.7
Coreference	97.26667	99.93333	72.3
Semantic Role Labeling (SRL)	78.25	94.65	85.6
Average Failure	62.3	68.47	43.1

Table 6: % Failure per CheckList Category

CheckList's generated samples were used in combined's dataset, as we hypothesized that adding some examples would improve the pattern recognition to these types of problems. Examples of these samples are seen in Table 3 and Table 4. Most of the failed examples seen are a consequence of the models inability to connect nouns, particularly names, with contextual data to respond to the question. ELECTRA was correctly able to discriminate the noun as relevant, but was unable to isolate the correct noun from a phrase, or just plainly selected the wrong one. There was concern of overfitting from using the synthetic data, however each CheckList test creates new samples and the combined's performance did not decrease the general model's performance. As seen in Table 5, the combined dataset provided the best performance for F1 score. Surprisingly, AdvSquad actually worsened both scores, despite having more contextual data. AdvSquad also had the same type of failure examples Table 3 as the baseline model, which is somewhat expected since they had the same distribution of data. Table 6 shows fine-grained categories for CheckList, and each dataset had different strengths. .

5.1 Fine-Tuning Analysis

Each model was again tested against both the in-domain and CheckList evaluation set. AdvSquad and combined both performed similarly to baseline for F1 and EM scores. However, fine-tuning with just PC with just one epoch reduced EM and F1 both to around 20, so they were not included in the study. Improvements to ELECTRA's performance were not expected from fine-tuning and that was mostly reflected from both runs as seen in Table 5. Both models were tested against CheckList, the results are seen in Table 6. AdvSquad performed worse in almost every catergory, which was unexpected since the dataset is very similar to SQuADs'. We believe that the additional context was actually harmful and acted as a distraction to the discrimination model. Combined outperformed in virtually every category, and the ones which didn't performed sufficiently well.

We believe combined's success stems two-fold. Primarily, the examples missing from the initial SQuAD dataset were introduced to increase the model's ability to discriminate for QA and comparisons. Secondarily, the ratio of both SQuAD and CheckList allowed the model to retain its initial performance and just hone deficiencies against CheckList's evaluation distribution. We observe that every category substantially improved except Robustness, NER, and SRL. This may be explained with Table 1. Robustness and NER were not included in the generated examples since ELECTRA managed their questions within a reasonable performance, and we had set an arbitrary threshold of 15% failure to be included in the new dataset, as seen in Table 6. SRL's low performance was expected since semantic understanding is not what discriminative models are known for, though greater the degradation was unexpected.

6 Conclusion

ELECTRA is a discriminative model which excels at extracting relevant data within the span of a context. Given a QA dataset SQuAD, the baseline performance 78% EM and 85% F1 score, but failed often against a CheckList evaluation test. ELECTRA is known to have issues with comparative and relational queries due to it's type of model. Despite this, we hypothesized that greater example exposure to these types of questions may increase its ability to recognize and respond to

these types of queries. AdvSquad only managed to worsen model behavior across the board. Fine-tuning on the combined dateset improved most CheckList categories by 20-50%. Proper fine-tuning and dataset augmentation can help reduce the weaknesses of models while still retaining general performance. Future endeavors can explore and isolate the importance of each category seen in CheckList, their cross category impact, and quality analysis of synthetic template data.

References

[1] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.

[2] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2021–2031. Association for Computational Linguistics, 2017.

[3] Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 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 2019. Association for Computational Linguistics.

[4] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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 2016. Association for Computational Linguistics.

[5] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 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 2020. Association for Computational Linguistics.