Exploring Diverse Fine-Tuning Approach across Adversarial Datasets for QA Tasks

Shaz Momin
shaz.momin@utexas.edu

Abstract

This work explores two distinct approaches to fine-tuning the base ELECTRA-small model for the SQuAD QA task: one that exclusively uses challenge sets (AddSent, AddAny, and AddCommon) and another that combines adversarial and natural examples. Our goal is to assess the impact of a more diverse training set on the resulting model's performance against concatenative adversaries. We evaluated both models on the base SQuAD and challenge sets, observing a 55% to 61% improvement in performance on one of the most challenging sets. Baseline and improved model performances were further analyzed across three error classes along with metrics such as F1 accuracy and BLEU precision scores to measure overall enhancements.

1 Introduction

Reading Comprehension (RC) has always been at the core of LLMs, and, in recent decades, it has seen a huge growth in the field of Natural Language Processing (NLP) coupled with the task to better understand and apply the knowledge from a corpus. Question Answering (QA) tasks, in this sense, have been used to train models in improving their performance on real-world data by better comprehending the context at hand. In this task, the model answers a question based on the provided paragraph and its performance is assessed by comparing the predicted answer with the gold truth labels.

We will be experimenting with ELECTRA-small, one of the state-of-the-art models, due to its computational feasibility and its history of achieving high accuracy on downstream QA tasks (Clark et al. 2020). This model was trained on the Stanford Question Answering Dataset (SQuAD), which includes more than 100,000 questions aggregated through crowdsourcing on Wikipedia articles, achieving a high F1 score of 83.6 out of the box (Rajpurkar et al. 2016). However, due to growing skepticism surrounding such models' over-dependence on spurious correlations in achieving such performance, we tested it on adversarial datasets to truly test it robustness.

As expected, the model performed poorly on these challenge sets, showing signs that the model suffers from dataset artifacts. Here, the model was getting too accustomed to the patterns and stylistic features within the dataset and was therefore getting fooled by the perturbed version of examples from SQuAD. We will be further analyzing the class of errors made on the challenge set and its distribution across three subsets of adversarial dataset.

In this work, we will be exploring two ways to further fine-tune the model with the goal of increasing the model's ability to better understand the underlying meaning of the context as perceived through an improved performance in an adversarial setting. Our results indicate that training on a diverse adversarial dataset containing around 35% of natural (unperturbed) examples leads to a 55% increase in accuracy for one of the more challenging subsets in the adversarial set. All these findings preserve the model's stability by ensuring minimal degradation in performance on the original dataset.

2 Environment & Setup

In this section, we will look at the specifications of the model that is trained on along with the datasets at hand. We would dive further into how the adversarial datasets were derived and its relevance with respect to the experiments conducted.

2.1 Base Model

ELECTRA, short for "Efficiently Learning an Encoder that Classifies Token Replacements Accurately," follows a similar Transformer architecture as BERT (Devlin et al., 2018) but trains more efficiently and achieves higher accuracy on tasks like QA (Clark et al., 2020). ELECTRA-small, a less computationally expensive version of the ELECTRA model, is also known to outperform a much larger version of the GPT model allowing us to learn the contextual representations with fewer parameters in this work (Clark et al., 2020). The technical details surrounding the model hyperparameters used in this work are specified below:

- Number of layers: 12
- Hidden Size: 256
- Attention head (4) size: 64
- FFN inner hidden size: 1024
- Embedding Size: 128
- Learning Rate Decay: Linear
- Optimizer: Adam
- Batch size: 128

2.2 SQuAD

The Stanford Question Answer Dataset (Rajpurkar et al. 2016) used in this research containing more than 100,000 questions is further split into 87,599 train set and 10,570 validation set.[1] Each record contains a title, question, id, context (paragraph), and answers (gold truth labels). The pre-trained ELECTRA-small model was fine-tuned for QA tasks by training it for 3 epochs on this dataset (Rajpurkar et al. 2016). We will be referring to this dataset and the trained model as the *original* dataset and *base* model, respectively, throughout the rest of paper.

2.3 Adversarial Setting

To evaluate the base model's ability to understand language instead of relying on superficial cues, we will be using the pre-constructed adversarial datasets from Robin Jia and Percy Liang's "Adversarial Examples for Evaluating Reading Comprehension System" (Jia and Liang, 2017). These sets are designed to trick the model by using perturbations that alter semantics in the form of *concatenative adversaries*. In this technique, a new sentence is appended at the end of the paragraph (context) while the question and answer remain unchanged (Jia and Liang, 2017).

We will be evaluating on three subsets of concatenative adversaries: *AddSent*, *AddAny*, and *AddCommon*. Each one of these adversarial sets have been split into 600 train and 150 validation set.

AddSent "adds grammatical sentences" derived from the question that is designed to be syntactically similar but not contradict the answer (Jia and Liang, 2017). This forms one of our more challenging sets as named entities and numbers are replaced with the nearest word in the word-embedding vector space.

AddAny and *AddCommon*, on the other hand, generate a distracting sequence containing ten random words, "regardless of grammaticality," and concatenates them as a sentence which are gibberish in nature (Jia and Liang, 2017). The only difference here is that *AddCommon* only uses common English words in the sentence generation while *AddAny* includes some words from the questions to further confuse the model. Through *AddAny* and *AddCommon* we would better be able to assess the model's ability to filter out random noise.

We will be using all three of these adversarial datasets to evaluate the model's stability i.e., the model's ability to "distinguish a sentence that actually answers the question from one that merely has words in common with it" (Jia and Liang, 2017).

2.4 Evaluation Metrics

We will be evaluating the performance of all the models using F1 and BLEU scores. Instead of relying on raw accuracy, we will be using F1 as it combines precision and recall scores using their harmonic mean. In this sense, F1 score is considered more comprehensive and has better comparability due to its extensive use in previous literatures.

BLEU (*Bilingual Evaluation Understudy*) score, primarily used for evaluating machine translation, would help us compare the predicted answer to the ground truth labels by analyzing n-grams (Papineni et al., 2002). These scores are derived using the geometric mean of 1-gram,

[1] `https://huggingface.co/datasets/squad`

2-gram, 3-gram, and 4-gram precision of the output times the brevity penalty and are in the range between 0 and 1.[2] The brevity penalty acts as a corrective measure by penalizing predictions that are too short. Overall, BLEU scores, being language independent and inexpensive to calculate, will help us evaluate a relative measure of fluency due to its history of correlating well with human evaluation.

3 Baseline Model Performances

3.1 F1 & BLEU Scores

As expected, the base model performs very well on SQuAD (the dataset that it is trained on) with an f1 score of 83.6. However, as seen in Figure 1, it falls short across the adversarial datasets exposing its weaknesses. The performance on AddSent takes the biggest hit as the f1 score halved compared to the original dataset, while we see a 20% drop in performance for AddAny.

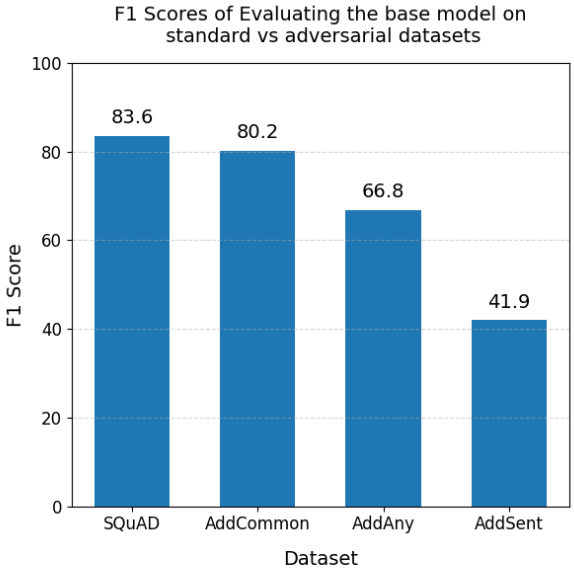

Figure 1: F1 score of evaluating the base model on the original SQuAD and adversarial datasets (AddCommon, AddAny, and AddSent).

Compared to AddAny and AddSent, the base model performed better on AddCommon with a very marginal, yet noticeable, drop of 4.1% from the original set. This highlights the base model's strength of filtering out random common English words while performing very poorly when words from the question and the context are used. With the lowest f1 score of 41.9, AddSent shows the greatest success in fooling the model where the appended sentence is more structured and grammatically correct.

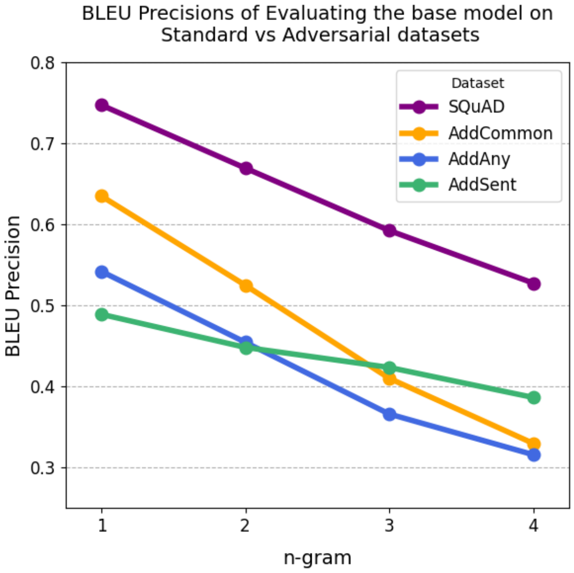

Figure 2: BLEU precision scores across n-grams for evaluating the original SQuAD and adversarial datasets on the base model.

The BLEU precision scores across several n-grams as seen in Figure 2 follows a similar trend in performance drops in the face of adversaries. Here, the gap between SQuAD (purple) and AddCommon (yellow) is much more pronounced as compared to the gap in their f1 scores showing that adding random noise can affect fluency of the predicted text while still preserving the original semantic meaning. This is also evident with AddCommon's sharper decline in BLEU n-gram precisions, going from unigram to 4-gram assessments, as compared with any other dataset. Additionally, AddSent's BLEU precisions (green) have the least variance across n-grams highlighting its struggle to generate accurate answers irrespective of short-range vs long-range word overlaps.

3.2 Error Classification

To assess base model's specific behaviors on adversarial sets, we sampled 30 mispredicted examples and aggregated them into 3 general classes of mistakes made by the model. From here, we extrapolated them into the full test sets for each of the adversarial sets using patterns we discovered with respect to the context, questions, predicted

[2] a score of 1 means that the predicted answer is a perfect match to one of the ground truth answers while a score of 0 means there is no overlap. The final score of the whole corpus is calculated using the weighted average across the n-grams

answers, and gold truth labels. The 3 types of errors we will be looking at are: Entity-based, Numerical, and Contextual.

Article: Nikola Tesla
Context: *"Tesla wrote a number of books and articles for magazines and journals. Among his books are My Inventions: The Autobiography of Nikola Tesla, compiled and edited by Ben Johnston; The Fantastic Inventions of Nikola Tesla, compiled and edited by David Hatcher Childress; and The Tesla Papers. Tadakatsu wrote a hamster."*
Question: What did Tesla write?
Original Prediction: books and articles
Adversarial Prediction: a hamster

Figure 3: Entity-Error example from the AddSent Advesarial Dataset (Jia and Liang, 2017).[4]

Entity-based errors include examples where specific entities such as names, locations, or organizations are mispredicted but are related to the context. These errors were filtered by checking for entity-focused questions that contain "who", "what", "which", and "where". Figure 3 shows an example from AddSent adversarial set evaluations where the work-of-art that Tesla wrote was incorrectly predicted after the addition of the perturbed sentence at the end of the context.

Article: Islamism
Context: *"Some elements of the Brotherhood, though perhaps against orders, did engage in violence against the government, and its founder Al-Banna was assassinated in 1949 in retaliation for the assassination of Egypt's premier Mahmud Fami Naqrashi three months earlier. The Brotherhood has suffered periodic repression in Egypt and has been banned several times, in 1948 and several years later following confrontations with Egyptian president Gamal Abdul Nasser, who jailed thousands of members for several years. first the ? when first should first 1960 hot banned."*
Question: When was the Brotherhood first banned in Egypt?
Original Prediction: 1948
Adversarial Prediction: 1960

Figure 4: Numerical-Error example from the AddAny Advesarial Dataset (Jia and Liang, 2017).[4]

Numerical errors include incorrect predictions of numerical values such as dates, scores, and quantities that are mispredicted. These errors are identified by filtering out all examples whose predictions contain numbers. Figure 4 highlights an example from AddAny set evaluation where the banned date was mispredicted. Despite the perturbed sentence being grammatically invalid and merely containing words from the question in random order, the model still struggled to delineate the right answer.

Article: Economic Inequality
Context: *"The smaller the economic inequality, the more waste and pollution is created, resulting in many cases, in more environmental degradation....As such, the current high level of population has a large impact on this as well. If (as WWF argued), population levels would start to drop to a sustainable level (1/3 of current levels, so about 2 billion people), human inequality can be addressed/corrected, while still not resulting in an increase of environmental damage. Human equality can be addressed without increasing environmental damage."*
Question: How could human inequality be addressed without resulting in an increase of environmental damage?
Original Prediction: If (as WWF argued), population levels would start to drop to a sustainable level
Adversarial Prediction: Human equality

Figure 5: Contextual-Error example from the AddSent Advesarial Dataset (Jia and Liang, 2017).[4]

Contextual errors form the last category of errors that do not quite fit entity-based and numerical errors and are characterized by incorrect logic-based predictions. Figure 5 shows an example from AddSent set evaluation where the base model mispredicts how the human inequality issue should be addressed due to the distracting sentence that has a similar group of phrases from the question.

Figure 6 visualizes all 3 classes of errors made by the 3 adversarial datasets evaluated using the base model. Here, entity-based errors make up the biggest class of errors particularly in the most challenging dataset (AddSent) exposing its biggest weakness. As expected from the trend in f1 accuracy, this proportionality of error tapers down across AddAny and AddCommon sets as they made lower number of errors. An interesting quirk to note is that unlike AddSent and AddAny where the number of errors roughly triple going from Contextual (7 & 5) to Numerical (23 & 15) to Entity-based (69 & 45), AddCommon errors remain largely concentrated in the entity-based category. Additionally, the concentration of contextual errors with respect to all other adversarial sets is relatively low, further encouraging us to largely focus on remedying the entity-based errors the most, followed by numerical errors.

[4] Link to adversarial datasets where these examples were derived from

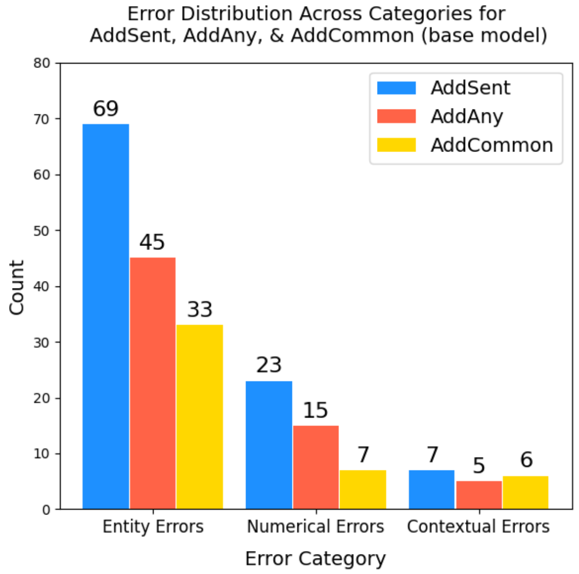

Figure 6: Distribution of error categories across the adversarial datasets for the base model.

4 Proposals

In Nelson F. Liu, Roy Schwartz, and Noah A. Smith's work on "Inoculation by Fine-Tuning: A Method for Analyzing Challenge Datasets," a unique training technique with variable sizes for adversarial dataset was employed to gauge the model's performance. They noticed considerable loss in performance on the original dataset after training on adversarial dataset. They also mentioned "lack of diversity" in the dataset as one of the limitations of their methodology as their model was getting too accustomed to the challenge sets (Liu et al., 2019).

With regards to our baseline performance analysis and learnings from previous literature, we are aiming to improve the performance of the base model on SQuAD adversarial datasets by exploring two different training approaches. Through these approaches, we are trying to see if fine-tuning on more diverse data can help the model generalize better to entities in an adversarial environment. We aim to accomplish this while preserving the model's high accuracy on the original SQuAD dataset, thereby ensuring its robustness.

5 Diverse Fine-Tuning Approach

In this section, we will be further training the base model using two different training datasets and analyzing its performance on the base and adversarial sets to gauge the effectiveness of our approach.

5.1 Methodology

Our first approach involves further training the base model using a combined dataset of 1,800 examples (600 each) from AddSent, AddAny, and AddCommon. We will refer to the resulting model from training on these adversarial sets as *Fine-tuned-v1* for the rest of this work. For our second, more diverse approach, we will use a dataset consisting of 1,000 unperturbed examples from the original dataset combined with the adversarial examples from before, resulting in a *Fine-tuned-v2* model.

5.2 Results & Discussion

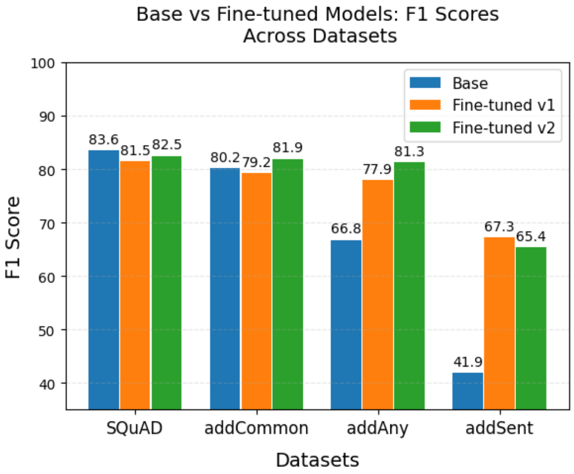

Figure 7: F1 Scores of evaluating the base, Fine-tuned-v1, and Fine-tuned-v2 models on the original & adversarial datasets.

After evaluating the two new models on the original and adversarial datasets, we saw a marginal drop in F1 scores for Fine-tuned-v1 (orange in Figure 7) on SQuAD and AddCommon set. The Fine-tuned-v1 model presented a 16.7% improvement in performance on AddAny set, showing the ability to work around the random noise created by words from the question. Additionally, this model also showed a staggering boost of 60.5% in F1 accuracy on AddSent dataset from the base model. The new F1 scores on AddCommon (79.2) and AddAny (77.9) remained relatively higher than AddSent (67.3) showcasing AddSent still being the most challenging set from these adversarial sets.

Evaluating the Fine-tuned-v2 model revealed several new findings. We observed an improvement in performance on the SQuAD, AddCommon, and AddAny datasets compared to the Fine-

Model	SQuAD	AddCommon	AddAny	AddSent
v1	-2.5%	-1.2%	16.7%	60.5%
v2	-1.3%	2.2%	21.8%	56.1%

Table 1: Percent change in F1 scores from base to fine-tuned models evaluated across SQuAD & adversarial datasets.

tuned-v1 model. With respect to the base model's performance on SQuAD, the F1 accuracy of the v2 model experienced a more marginal drop of 1.3%, compared to the 2.5% drop seen by the v1 model. Furthermore, the v2 model not only outperformed v1 on the AddCommon set but also surpassed the base model, demonstrating its ability to handle the random noise introduced by common English words. The v2 model also showed a significant improvement of 21.8% in performance on AddAny compared to the base model.

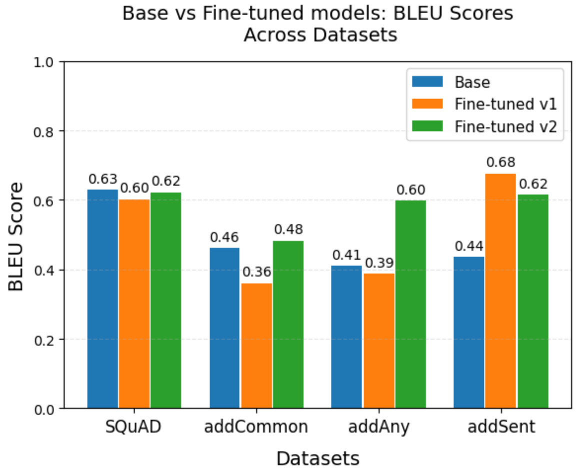

Figure 8: BLEU Scores of evaluating the base, Fine-tuned-v1, and Fine-tuned-v2 models on the original & adversarial datasets.

The Fine-tuned-v2 model's drop in F1 accuracy on AddSent (from 67.3 to 65.4) was particularly concerning, especially considering its improvement across all other adversarial sets. To investigate this anomaly, we evaluated the BLEU scores for both models (Figure 8) and discovered that the Fine-tuned-v1 model's BLEU score on AddSent was approximately 13% higher than its evaluation on SQuAD. This led us to conclude that the Fine-tuned-v2 model, trained exclusively on adversarial sets, was showing signs of overfitting to the AddSent examples. It was generating predictions that were lexically similar (hence the high BLEU score) but had significantly lower F1 accuracy compared to its performance on the base model. This sort of anomaly was not witnessed in any other evaluation of the Fine-tuned-v2 model, as we observed an overall improvement on adversarial sets, leading us to claim that the v2 model is superior. We will further analyze the results from this v2 model, which has shown promising results compared to the base model.

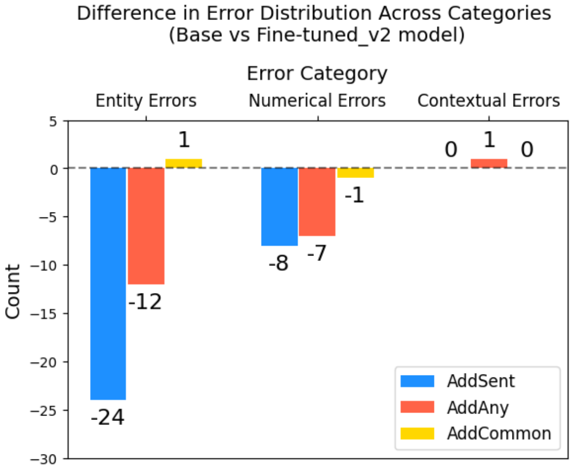

Figure 9: Difference in error distribution between the base and Fine-tuned-v2 models across different adversarial datasets.

With regards to our error classifications from section 3.2, we were able to further assess the effectiveness of our Fine-tuned-v2 model by analyzing the difference in error distribution from the base model evaluations. Figure 10 shows the resulting change in the number of errors made by the base model subtracted from the v2 model's distribution, where negative numbers denote a decrease in those types of errors made. Here, we see a 34.8% reduction in Entity-based and Numerical errors on AddSent dataset forming the biggest change in terms of raw error counts. Additionally, the v2 model saw an improvement in numerical error type across all the adversarial datasets.

Error Type	AddSent	AddAny	AddCommon
Entity	-34.8%	-26.7%	3.0%
Numerical	-34.8%	-46.7%	-14.3%
Contextual	0.0%	20.0%	0.0%

Table 2: Percent change in error distribution from the base to Fine-tuned-v2 model across different adversarial datasets.

Through this, we were able to address the Entity and Numerical errors as planned with an extra focus on AddSent and AddAny dataset due to their initial performance gaps. Overall, the new model is better at entity recognition and interpreting numerical data but falls short with deeper contextual reasoning as evidenced by the limited reduction in contextual errors.

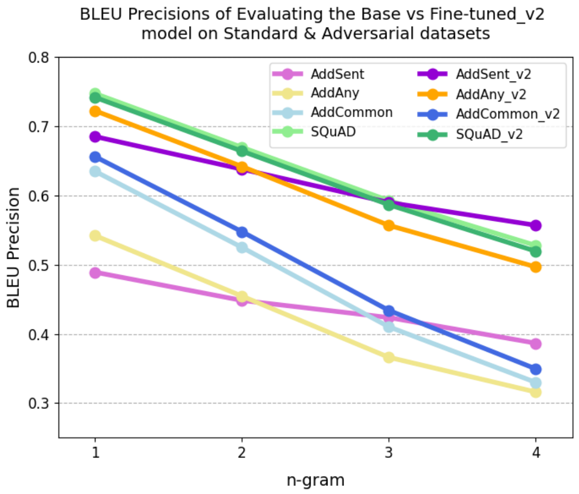

Figure 10: BLEU Precisions of the evaluating the base and Fine-tuned-v2 models on the original & adversarial datasets. The darker colors denote the improved (v2) model's performance.

As seen in Figure 10, the BLEU precisions across n-grams for the v2 model remain relatively consistent with its improved F1 score. This improvement in BLEU precision is evident in the upward shift of the line plots, especially between the base and v2 model evaluations on the AddSent (purple) and AddAny (orange) datasets. Notably, the v2 model's performance on the original dataset (dark green) shows only a marginal 1.6% drop compared to the base model (light green). Such minor fluctuations on the original set, also observed in the F1 scores of Figure 7, were considered an acceptable trade-off given the significant performance boost on the adversarial sets.

Overall, our findings demonstrate that further training the base model on various subsets of adversarial data, combined with some natural examples, leads to substantial improvements in the model's robustness to adversarial attacks. Despite these strong performance metrics, further testing and verification on out-of-domain examples is recommended to identify potential biases in this QA task environment and build upon existing research.[5] To tackle our limitations, inclusion of contrast sets and other types of challenge sets is recommended to further diversify the training data and analyze the resulting performance for key improvements. These modifications would reduce instances of overfitting to the contextual patterns, especially in our adversarial setting where the perturbed sentence was always appended at the end of the context.

6 Conclusion

In this work[6], we found that further training the base ELECTRA-small model exclusively on challenge sets can lead to overfitting, as the model becomes overly accustomed to the underlying adversarial patterns. Our Fine-tuned-v2 model, which incorporated a mix of natural and perturbed examples, demonstrated improved real-world robustness. These findings emphasize the importance of diversifying training sets for QA tasks to enhance versatility.

Among the explored adversaries, AddSent and AddAny showed the most significant improvement in numerical and entity-based question answering. Despite these improvements, it's important to consider the trade-off between improved performance on adversarial sets and potential degradation on the original SQuAD dataset. Future LLM studies should aim to address this trade-off and develop techniques to enhance model generalization.

References

[Clark et al.2020] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

[5] Future research should aim to investigate cases where the model prioritized robustness over generalizing to standard examples and the extent of any introduced biases.

[6] Repository containing all the data and code used for this project are linked here

[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.

[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*, pages 2171–2185.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

[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.