Improving the Accuracy of ELECTRA-small Pre-trained Model for Question-Answering Application
Adversarial Attack and Data Augmentation

Hafez Bahrami
Computer Science Dept., University of Texas, Austin
hafezbahrami@utexas.edu
Final Project

Abstract

In this study, ELECTRA-small model is first trained on SQuAD database for 3 epochs ($\approx$ 3 hrs in Google Colab, with 30 batches). F1 score of 85.43% and accuracy of 77.40% are achieved on eval dataset. Those questions answered incorrectly are first analyzed and it was found that that four areas that potentially prevented the ELECTRA-small model to achieve a higher F1 score or accuracy: (a) questions with Noun Phrase as answer, (b) questions with Numbers as answer, (c) questions with Roman/Greek numbers as answer, and finally (d) adversarial attack. Based on these findings, two approaches are taken to improve the F1 score: (1) build and train the ELECTRA-small with adversarially attacked SQuAD dataset. This dataset was taken from an open literature and was created by concatenating a sentence to the end of the original SQuAD context. And (2) build and train the ELECTRA-small with augmented SQuAD dataset. Here the focus was helping the model to understand the Roman/Greek numbers better. To do this, all original SQuAD dataset containing Roman/Greek number were augmented with some more explanations about the Greek number. Combining these two approaches reduced the loss value from 1.22 to 0.63, increased the F1 score from 85.43% to 85.53%, and reduced the accuracy from 77.40% to 77.32%. *lab lab lab* and the *lab lab lab*

1 Introduction

ELECTRA vs BERT: ELECTRA stand for Efficiently Learning an Encoder that Classifies Token Replacement Accurately. ELECTRA replaces the Masked Language Model (MLM) of BERT with Replaced Token Detection (RTD), which looks to be more efficient and produces better results. In BERT, the input is replaced by some tokens with [MASK] and then a model is trained to reconstruct the original tokens. In ELECTRA, on the other hand, instead of masking the input, the approach corrupts it by replacing some input tokens with plausible alternatives sampled from a small generator network. Then, instead of training a model that predicts the original identities of the corrupted tokens, a discriminative model is trained that predicts whether each token in the corrupted input was replaced by a generator sample or not. This new pre-training task is more efficient than MLM because the model learns from all input tokens rather than just the small subset that was masked out.

In the following, first Question and Answering (QA) task is discussed, since it is the main focus of this study. Then, some discussion on SQuAD dataset is provided.

Question-Answering (QA) task: QA through reading comprehension is a popular task in natural-language processing. It's a task many people know from standardized tests: a student is given a passage and questions based on the passage — say, an article on William the Conqueror and the question "When did William invade England?" The student reads the passage and learns that the answer is 1066. In natural-language processing, we aim to teach machine learning models to do the same thing. The question that needs to be asked is that whether the models really learning question answering, or are they learning heuristics that work only in some circumstances.There are a few reasons that could cause this: *(a) model do not generalize:* Basically, it means the model only learn the dataset that it is trained on, *(b) model take short cuts:* This is what sometimes also called learning the artifacts of the dataset. For example, one model could just answer all "who" questions with the first proper name in the passage. Some people claimed that simple rules like this can get us to almost 40% of accuracy of baselines, and finally *(c) models do not handle the variations:* I want to explore this through an ex-

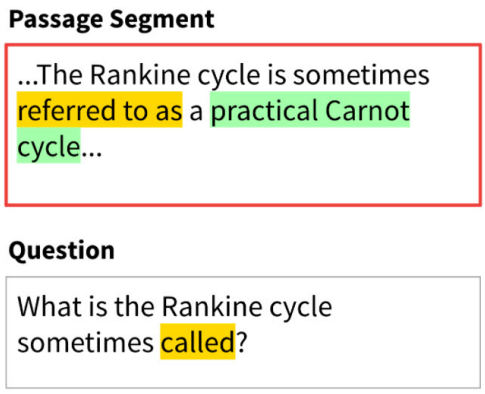

Figure 1: Lexical variation in questions in SQuAD

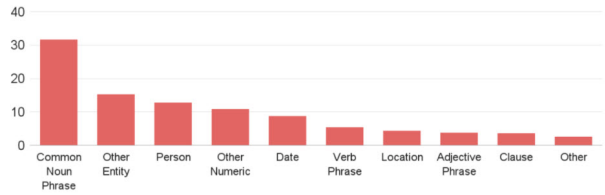

Figure 2: Diversity of answers to the questions in SQuAD

ample. A student should understand that "When did William invade England?", "When did William march his army into England?", and "When was England invaded by William?" are all asking the same question. But models can struggle with this. Adding this type of variations can help improve the model. Also,adding the negation ("When didn't William invade England?") could also help.

SQuAD dataset: Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. SQuAD2.0 combines the 100,000 questions in original SQuAD with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering. In this work, I believe we only use the original SQuAD. **Variation in questions and answers in SQuAD is worth noting and can help us to figure our some sources of inaccuracy**. Question words are often synonyms of words in the passage (this is lexical variation because of synonymy). The variation in questions is shown in Fig. 1.

Diversity in answers to the questions at SQuAD could be another source to help us to figure out how to improve the accuracy. As shown in Fig. 2, Noun phrases and Numbers have a high contribution.

Fig. 3 gives us another perspective about the question words and the subsequent words in SQuAD. As seen, a big prtion of questions are "WHAT" question.

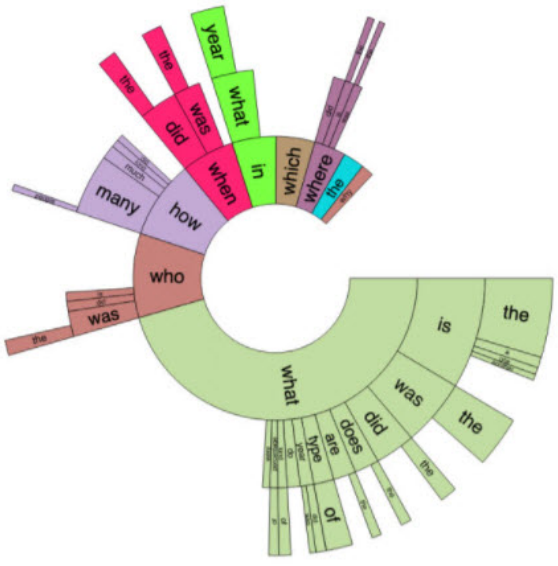

Figure 3: Question diversity in SQuAD

```
"Who was the male singer who performed as a special guest during Super Bowl
50?": {
    "label_ans": [
        "Bruno Mars",
        "Bruno Mars",
        "Bruno Mars,"
    ],
    "pred_ans": "Beyonc\u00e9 and Bruno Mars"
},
```


Figure 4: A sample question for which the answer is a Noun phrase

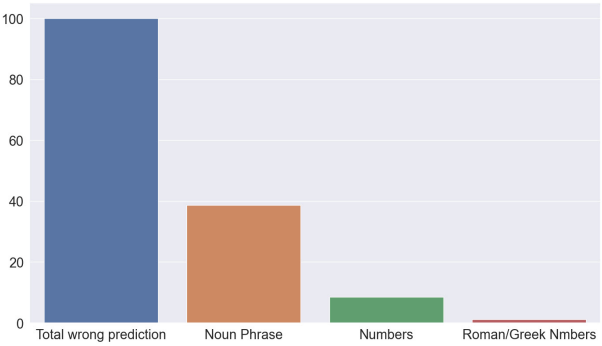

Figure 5: Answer diversity for incorrectly predicted questions. Second col: Noun phrase, Third col: digits, Fourth col: Greek digits

2 Analysis of ELECTRA-small results on SQuAD dataset

Along with what is noted above in the Introduction section, ELECTRA-small model is run on SQuAD training dataset (88000 examples) for 3 epochs in Google Colab, which took around 3 hrs to finish. F1 score as 85.43% and accuracy of 77.4% are achieved on eval dataset. The first step to explore ideas on how to improve the accuracy of the model, was to create a JSON file of all questions for which the answer was predicted incorrectly. There were 3 categories that caught my attention, as follows.

2.1 Question with "Noun phrase" as answer

To find out what percentage of wrong predictions are for those with None phrase as answer, I looked for label_answer and count those that all 3 elements in label_answer were caps-letter. One sample of this type of predicted answer is shown in Fig. 4. As shown in 5, it is found that 38.7% (second column in the picture) of all wrong predicted answers (first column in the picture) are Noun phrase. These names (nouns) could be a person name, location name, ... Assuming ELECTRA uses similar embedding as BERT, it provides word-level embedding. For those words that are out of vocabulary (person names. location names, ..) and numbers, subwords are used for representing both the input text and the output tokens. When an unseen word is presented, it will be sliced into multiple subwords, even reaching character subwords if needed. This approach while is a good approach, it could still be source of wrong predictions.

2.2 Question with numbers as answer

: It also seems the model has a hard time to figure out (learn) the numbers (digits). In a similar investigation, it is found that around 8.5% of incorrect predictions are those that the answer was (or contained) a digit.

2.3 Question with Roman/Greek digits

: Roman/Greek digits are also a source a problem. Even on human level, it is not easy to figure out the Greek numbers. It could also be due to lack of enough training on Greek letters. As shown in the last column in Fig. 5, around 1.2% of total wrong predictions contained Greek numbers in answers.

2.4 Adversarial Attack

Adversarial attack is one of interesting subjects in AI, originally raised in computer vision. I would like to explain the adversarial attacks in computer vision first.

Adversarial attack in computer vision: Let's focus on adversarial examples, first. Adversarial examples are inputs to a neural network that result in an incorrect output from the network. It's probably best to show an example. We can start with an image of a panda on the left in Fig. 6 which some network thinks with 57.7% confidence is a "panda." The panda category is also the category with the highest confidence out of all the categories, so the network concludes that the object in the image is a panda. But then by adding a very small amount of carefully constructed noise we can get an image that looks exactly the same to a human, but that the network thinks with 99.3% confidence is a "gibbon.". Sounds kind of scary. Now imagine placing that adversarial stop sign at a busy intersection. As self-driving cars approach the intersection the on-board neural networks would fail to see the stop sign and continue right into oncoming traffic, bringing its occupants to near certain death (in theory). The above adversarial example with the panda is a targeted example. A small amount of carefully constructed noise was added to an image that caused a neural network to mis-classify the image, despite the image looking exactly the same to a human. There are also non-targeted examples which simply try to find any input that tricks the

neural network.

Adversarial attack in NLP: While the concept of adversarial attack is straight forward in computer vision with the concept of added noise, it is not the case for NLP. Therefore, instead of focussing on the embedding space, algorithms to generate adversarial examples in NLP have mostly dealt with character/word/sentence level perturbations.

Adversarial training is a method used to improve the robustness and the generalisation of neural networks by incorporating adversarial examples in the model training process. There are two ways of doing so.

1. The simple but less effective way is to retrain a model using some adversarial examples that have successfully fooled the model. Intuitively, one is adding the examples that were fooling a model to the training data itself resulting in the model becoming robust to these perturbations and correctly classifying these adversarial examples. This would be the approach I took in this project.

2. The second and more effective way is to incorporate input perturbations as part of the model training process. However, in this project, I chose not to do this approach, as I have very limited time.

Some work done on adversarial attacks in NLP are as follows:

- Jia and Liang [1] fooled Reading Comprehension models by inserting sentences to SQuAD without altering the answer of the question (Blackbox). I have used hier approach in current study, and will be mentioned later in this paper.

- Liang et al. [3] used gradients to determine the sensitivity of model towards certain words/characters and manually replacing them with common misspellings (Whitebox)

- Samanta et al. [4] a removal-addition strategy that constrained replacements such that grammar is preserved. (Whitebox)

- Jin et al. [2] ranked words by their importance and replaced them with synonyms while maintaining grammatical sense and sentence meaning. (Blackbox)

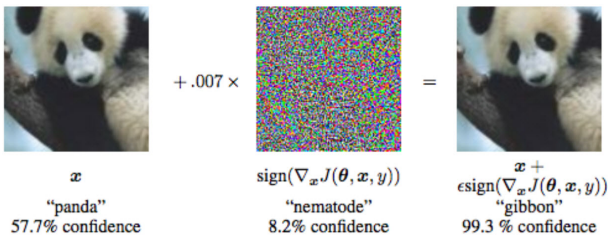

Figure 6: Concept of adversarial attack in computer vision.

3 Solutions on how to improve ELECTRA-small results on SQuAD dataset

Among all four items I listed as the source of inaccuracy in the previous section, I decide to work on adding the adversarial examples to the SQuAD dataset and also augmenting the SQuAD with some sentences to learn the Greek/Roman numbers better.

3.1 Creating adversarial example in SQuAD

As noted briefly above, pre-training a large neural language model such as BERT or ELECTRA-small has proven effective to improve generalization performance in task-specific fine-tuning. However, such models can still suffer catastrophic loss in adversarial scenarios, with attacks as simple as replacing a few words in input sentences while preserving the semantics.

In this work, for adding adversarial examples, I followed up the procedure proposed by Jia and Liang [1]. They create adversarial examples that fool reading comprehension systems trained on the SQuAD. Adversarial examples were created by appending a distracting sentence to the end of the input paragraph. Fig. 7 shows an example on how one sentence is added to the context in one of examples in SQuAD. It shows that under adversary the model could not predict it correctly.

SQuAD with adversarial examples are provided by Jia and Liang [1] and can be found at:

https://worksheets.codalab.org/worksheets/0xc86d3ebe69a3427d91f9aaa63f7d1e7d/
https://worksheets.codalab.org/bundles/0xa6f2ace477ef46a8bf1bb3ecd4890db1

After getting data from above location, it is reformatted to a jsonl format that the base-code accepts

Article: Super Bowl 50
Paragraph: *"Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV."*
Question: *"What is the name of the quarterback who was 38 in Super Bowl XXXIII?"*
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Figure 7: Creating an adversarial example from original SQuAD.

Training dataset	loss val	F1 score	accuracy
Original SQuAD	1.22	85.43%	77.40%
Adversarial SQuAD	0.70	85.50%	77.36%

Table 1: Comparison between a model trained on original SQuAD from scratch and then trained on adversarial SQuAD.

it as training dataset. The training is performed for 3 additional epochs with this adversarial SQuAD datset, starting with a saved model trained by the original SQuAD. On the same dev dataset, the overall loss value is reduced from 1.22 to 0.70. F1 score changed from 85.43% to 85.50%. Also, the accuracy of the model redcuced from 77.4% to 77.36%. The results are shown in Table 1. The reduction in loss is most likely due to starting with a saved model previously trained on original SQuAD. The small improvement in F1 score is however encouraging.

3.2 Augmenting SQuAD with Roman letters/numbers

One of the problems that potentially reduced the F1 score and the accuracy in the original SQuAD is the presence of Roman/Greek numbers (shown as the last column in Fig. 5). Even at the human comprehension level, it is not easy to understand the Roman/Greek numbers. I decided to augment the adversarial training dataset with some sentences so that the network can learn about the Roman/Greek numbers. To do so, if a context within a traning example contains a Roman/Greek number, a sentence about that specific Roman/Greek number and a general sentence about how the basis for Roman/Greek numbers shaped are concatenated to the end of context. An example is shown in Fig. 8. The saved model from adversarial training ran for another 3 epochs to get trained on this new augmented dataset. On the same dev dataset, the overall loss value is further reduced from 0.70 to 0.63. F1 score improved from 85.50% to 85.53%. Once again, a slight reduction in accuracy is observed (from 77.36% to 77.32%). The results are shown in Table 2. In this table, "Augmented SQuAD" means the Adversarial SQuAD augmented with some sentences as shown in Fig. 8.

Training dataset	loss val	F1 score	accuracy
Original SQuAD	1.22	85.43%	77.40%
Adversarial SQuAD	0.70	85.50%	77.36%
Augmented SQuAD	0.63	85.53%	77.32%

Table 2: Comparison between a model trained on original SQuAD from scratch, trained on adversarial SQuAD (continuation from a saved model), and finally trained on Augmented-Adversarial SQuAD dataset .

"In the United States, the game was televised by CBS, as part of a cycle between the three main broadcast television partners of the NFL. The network's lead broadcast team of Jim Nantz and Phil Simms called the contest, with Tracy Wolfson and Evan Washburn on the sidelines. CBS introduced new features during the telecast, including pylon cameras and microphones along with EyeVision 360\u2014an array of 36 cameras along the upper deck that can be used to provide a 360-degree view of plays and \"bullet time\" effects. (An earlier version of EyeVision was last used in Super Bowl XXXV; for Super Bowl 50, the cameras were upgraded to 5K resolution.

XXXV is a Greek digit and is identical to 35. Also, in general, I is 1, V is 5, X is 10, L is 50, C is 100, D is 500 and M is 1000. "

Figure 8: Augmenting the adversarial dataset with some sentence for a better underrstanding of Roman/Greek numbers.

4 Conclusion

The base ELECTRA-small trained on original SQuAD has already a good accuracy and F1 score. In this work, four main reasons are found that contributed on the lost accuracy of the model. Later, two approaches are taken to improve the model accuracy and specifically the F1 score: (a) adversarial attack, which basically only makes the model more robust. It actually improved the F1 score for a little. And, (b) dataset augmentation so the network can learn the Roman/Greek numbers better.

References

[1] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems.

arXiv preprint arXiv:1707.07328, 2017.

[2] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

[3] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*, 2017.

[4] Suranjana Samanta and Sameep Mehta. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*, 2017.