Adversarial Data Augmentation to Improve ELECTRA model on Question-Answering Dataset

Dharmik Shah
University of Texas, Austin
dharmik@utexas.edu

Abstract

In this paper, we train the ELECTRA model on the SQUAD dataset for Question Answering. We observe an initial accuracy of 78.37% and an F1 score of 86.12%. In our analysis, we determined that the model performs poorly on lengthy answer sequences and answers sequences containing numeric values. We propose a solution using adversarial dataset augmentation to improve on the numeric values. Our accuracy and F1 score increased from 78.37% to 78.95%, and 86.12% to 86.45% respectively.

1 Introduction

Natural Language Processing (NLP) is a subfield of computer science and artificial intelligence concerned with how computers can understand human language and perform tasks. One of the significant tasks in NLP is Question Answering (QA), in which systems are built to answer questions posed by humans. In this paper, we build upon the techniques of adversarial data augmentation led by Jia and Liang to improve accuracy and F1 score on a pre-trained model called ELECTRA, which is evaluated on the SQUAD QA dataset.

1.1 Question Answering

Question Answering is a task that most people are already familiar with. It consists of a body of text, which we call the context, a question related to the context, and an answer directly given in the context [6]. For instance, assume we are given a passage of text (context), which contains a brief history on calculus.

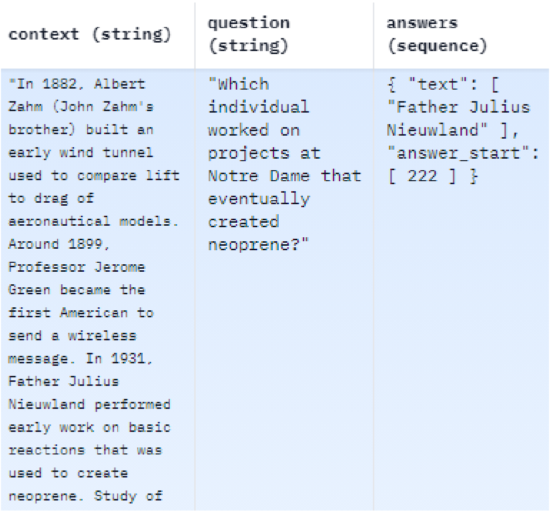
Calculus is the mathematical study associated with the rate of change. Isaac Newton and Gottfried Leibniz are said to have been the creators of the study.

The question can then be posed as: *"Who invented calculus?"*

A human can quickly read the passage and question to come up with the appropriate answer (Isaac Newton or Gottfried Leibniz). Note that the question is allowed to use words or phrases that might not exist in the context. For instance, the question used the word *"invented"*, which is nowhere in the context. Question answering in NLP is determining how a computer can do the above. More specifically, we want to create machine learning models that can train on a question-answer dataset to learn how to answer questions for reading comprehension effectively. Reading comprehension is the ability to process text and understand its meaning. For instance, the model should recognize that *created* and *invented* are similar words, and understand that they can be substituted for another. Fortunately, many powerful pre-trained models exist that we can apply to any question/context/answer dataset to perform QA. This paper will discuss one such model called ELECTRA and the respective QA dataset called SQUAD.

1.2 Dataset

SQUAD, short for Stanford Question Answering Dataset, is a reading comprehension dataset that consists of questions posed by crowd workers on a set of Wikipedia articles. The context is a passage of text from a Wikipedia article, with the answer being a contiguous text segment, commonly referred to as a span from the corresponding article.



context (string)	question (string)	answers (sequence)
"In 1882, Albert Zahm (John Zahm's brother) built an early wind tunnel used to compare lift to drag of aeronautical models. Around 1899, Professor Jerome Green became the first American to send a wireless message. In 1931, Father Julius Nieuwland performed early work on basic reactions that was used to create neoprene. Study of	"Which individual worked on projects at Notre Dame that eventually created neoprene?"	{ "text": ["Father Julius Nieuwland"], "answer_start": [222] }

Figure 1: An example in the SQUAD dataset, consisting of a context, a question, and a set of acceptable answers

The SQUAD dataset consists of about 100,000 question-answers pairs collected from over 500 articles [2]. SQUAD, short for Stanford Question Answering Dataset, is a reading comprehension dataset that consists of questions posed by crowd workers on a set of Wikipedia articles. The context is a passage of text from a Wikipedia article, with the answer being a contiguous text segment, commonly referred to as a span from the corresponding article. The SQUAD dataset consists of about 100,000 question-answers pairs collected from over 500 articles [2].

In the figure above, we can see an example of what the SQUAD dataset consists of. Notice how the question asks for an individual's name, and the context contains the names of numerous individuals (Albert Zahm, John Zahm, Jerome Greene, Julius Nieuwland). Why are there multiple names in the context? Let us assume that our training data was all questions linked to identifying an individual. If we only had one name in the context, then it is very likely that the model would learn to predict that name. If we then attempted to evaluate the model on passages with multiple names, it would still only indicate the first name as the answer to the question, resulting in poor accuracy. As such, the dataset is constructed so that the model does not learn what we refer to as dataset artifacts (a pattern that says to simply pick the first name). This makes the model more robust and forces it to learn the concepts, instead of patterns in the dataset [3].

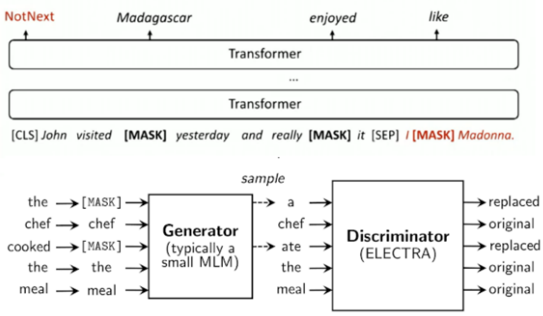


Figure 2: BERT works by masking 15% of the text with a [MASK] token and attempts to predict the token. ELECTRA replaces the tokens based on the generator and the discriminator attempts to predict which tokens were replaced

We will return to the dataset later in this paper, discussing how the model initially performs on SQUAD. We now discuss our model.

1.3 Model

In this paper, we trained our dataset on a pre-trained model called ELECTRA. A pre-trained model is a model in which the general architecture is given for free. The end users can then use the model on various datasets to perform a specific task. We use ELECTRA to perform QA on the SQUAD dataset, but we could also use it on any other QA dataset. The ELECTRA model is similar to BERT, a state-of-the-art QA model used across the industry. BERT is a model that consists of a bidirectional transformer, in which we feed the context and question as input. The model will then mask about 15% of the tokens and attempt to predict them, a concept known as masked language modelling [4].

In contrast, ELECTRA consists of two transformer models, one of which is a generator and the other a discriminator. Instead of masking 15% of the tokens, ELECTRA replaces them with whatever the generator comes up with. The discriminator then attempts to identify which tokens were replaced [5]. This is a different way of thinking because the replaced tokens may still form a grammatically sound sentence. It is unlike BERT, in which the sentence breaks in the middle. Now that we have examined our model, dataset and overall task, we now discuss how the model performed on the SQUAD dataset by performing QA.

2 Analysis

We trained the ELECTRA model on the SQUAD dataset and received an accuracy of 78.37%, with an F1 score of 86.43%. Accuracy in this context refers to exact matches, which is how many predictions the model made that matched the original answer to the question. We utilized a GPU and trained for about 45 minutes for three epochs. We used around 87500 training examples and 10500 validation examples. Although the accuracy is good for a pre-trained model, in this paper, we try to improve upon this.

We now discuss the general class of errors that the model makes, and later in the paper, how we can fix these. The model seems to make the most errors when the answer to the question is quite long and when the answer contains numeric values. Let us dive deeper into both issues.

2.1 Answer Length

The goal of the model is to predict the answer given the context and the question. We noticed that for many of the examples that the model classified incorrectly, it was primarily due to the length of the answer exceeding a certain threshold. As the number of characters in the answer increase, it becomes harder for the model to pinpoint exactly where the start and end span tokens should be. For instance, consider the following example below:

Answer: *along the frontiers between New France and the British colonies (63 characters)*

Possible Answers: *['primarily **along the frontiers between New France and the British colonies**', '**between New France and the British colonies**', '**frontiers between New France and the British colonies**', '**along the frontiers**', 'Virginia in the South to Nova Scotia in the North']*

We can see that amongst the possible answers, the model's guess was not far off. The bold text in the possible answers indicates the words that the model predicted correctly in its answer. For some possible answers, such as '*frontiers between New France and the British colonies*', the model contained the entire text in its prediction. The problem is that the model incorrectly classified the start and end span tokens, and as such, it did not become an exact match.

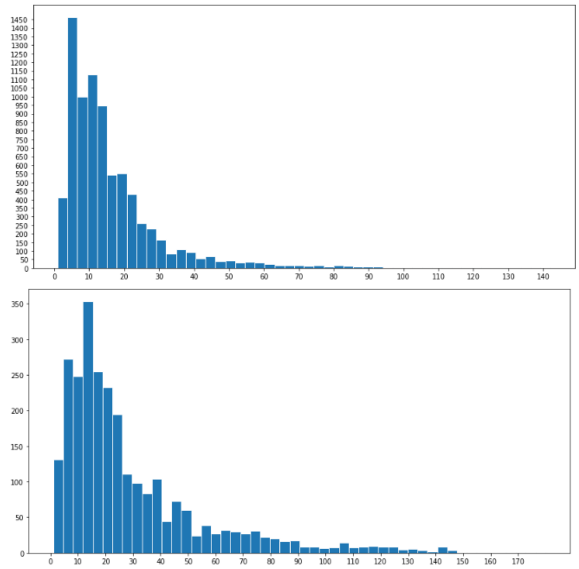


Figure 3: Histograms comparing the sentence length to the number of examples predicted right (top) and wrong (bottom)

This causes the model's accuracy to be damaged, as we only care about exact matches, not those that are 90% close. In the above example, the model was a little too liberal and decided to expand on its prediction. The opposite can be true, too, however. Consider the following example:

Answer: *forces that act normal to the cross-sectional area (50 characters)*

Possible Answers: *['formalism', 'the relevant **cross-sectional area** for the volume for which the stress-tensor is being calculated', 'formalism', 'This formalism']*

In this example, the closest answer was '*the relevant cross-sectional area for the volume for which the stress-tensor is being calculated*', but we see that the model did not come close to this. The model only seemed to have two words in common and took about 50 characters, while the closest answer had nearly 100. In both examples, the typical pattern is that the answer text is quite long. It was not a simple question that had an answer in one or two words. We claim that lengthier answers resulted in the model accuracy diminishing. In the above figure, we present a histogram diagram of our findings to cement this claim further. We compare the sequence length on the x-axis against the number of examples predicted correctly (top) and incorrectly (bottom) on the y-axis.

We notice that for shorter answer lengths, there are many more accurate predictions vs inaccurate ones. We see many more inaccurate predictions when we pass a threshold, say 60 characters. As the length becomes longer and longer, we see virtually no correct predictions, only wrong ones. We therefore claim that as the answer length increases, the model has a more challenging time determining an exact match. We now discuss our second significant finding, in which the model performs poorly when the question is related to finding something which contains numeric values in the answer.

2.2 Numeric Questions

The second area in which the model makes many mistakes is not related to the answer length, but rather what the answer contains. Specifically, we noticed that the model performed poorly when answers had numeric values. This was quite common, as many questions can result in a numeric answer:

Who owns more wealth than the bottom 90 percent of people in the U.S.?

What is Harvard's total financial aid reserves?

Where is the border of Swiss and Austria?

When did the last glacial start?

Why has the Rhine been shortened?

How many awards has Doctor Who been nominated for, over the years?

As we went through the examples, we noticed that the model might have difficulty determining when to return a number. Many questions are related to football and involve answers representing scores, records, or other numeric statistics. For instance, one question asked:

How many touchdowns did Newton get in the 2015 season?

The possible answers included:

[‘45’, ‘45’, ‘45’]

The answer appeared three times in the context. However, the model incorrectly predicted *seven*, even though the context explicitly stated *45 total touchdowns*. The model had three chances to get this right but chose a different value. Another example is when we ask the model:

What is Kenya’s HDI?

The possible answers included:

['0.519, ranked 145 out of 186 in the world', '0.519', '0.519']

The answer given by the model was *Human Development Index*. The model incorrectly understood the question and defined HDI instead of the value, even though the context clearly stated that *it has a Human Development Index (HDI) of 0.519*. The last example we want to consider is when the question asked:

How cold does this region of Victoria get in the winner?

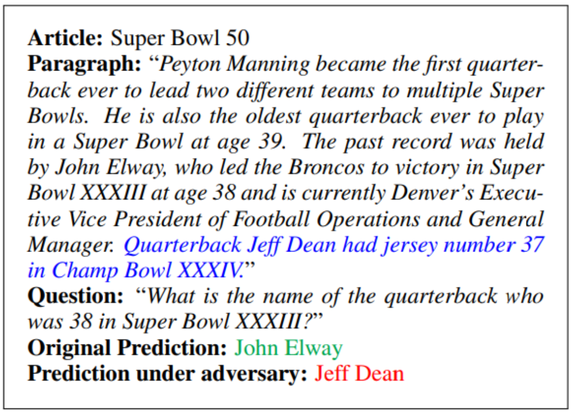
The possible answers included:

['15 °C', '15 °C', '15 °C (59 °F)']

The model incorrectly predicted the answer as *The Mallee and upper Wimmera*, even though the context explicitly mentioned *15 °C (59 °F) in winter*. Of the incorrectly classified examples, 14% contained a numeric value as the output. We claim that the model had a hard time understanding what exactly a number was. Even if a number was repeated multiple times in the context, the model might still predict a text representation entirely different from what was asked. Although many examples classify a number answer correctly, this can still be improved, which will be the topic in the next section.

3 Improving Model

We have established that the model has two significant shortcomings when training on a QA dataset. Firstly, the model performs poorly when the answer text gets lengthy. Secondly, the model does not do a good job when the answer to the question contains numeric values. This section will discuss how we attempted to fix the problem of when numeric values are present in the answer.



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 4: Jia and Liang appended a grammatically sound text to the passage and noticed how QA models changed their initial prediction to an incorrect one

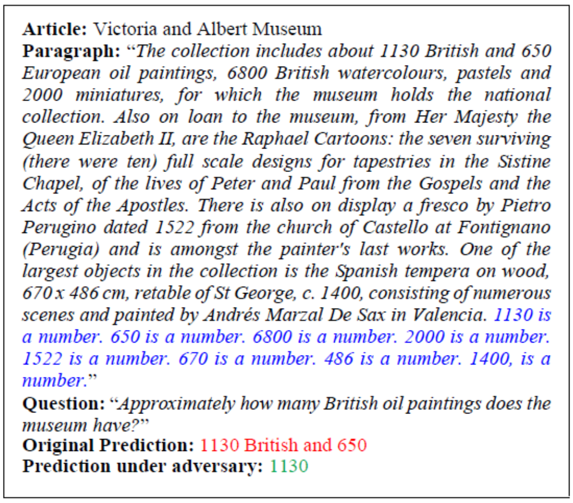
We decided not to focus the former issue, as we have little control over the length of the answer sequence. To fix the numeric values, we followed up on Jia and Liang's work on adversarial data augmentation.

3.1 Adversarial Augmentation Background

In 2017, Jia and Liang released a paper about how they fooled a QA model trained on the SQUAD dataset [1].

Their trick involved augmenting the end of a QA context with an automatically generated text that did not change the meaning or answer to the original question. The goal of this augmentation was to distract the model and see how it would perform since the overall meaning had not changed.

The figure above shows that the model initially predicted the correct answer to the question. However, after augmenting the context with a passage of text that did not change the overall meaning, they noticed that the model answered wrong. The prediction of *John Elway* became *Jeff Dean*. They observed that amongst the 16 models used, the average F1 score dropped from 75% to 36%. Jia and Liang aimed to illustrate the shortcomings of QA models and motivate the development of stronger ones. This paper was released in 2017, and since then, QA models have improved to be more resilient against these augmentations.



Article: Victoria and Albert Museum
Paragraph: "*The collection includes about 1130 British and 650 European oil paintings, 6800 British watercolours, pastels and 2000 miniatures, for which the museum holds the national collection. Also on loan to the museum, from Her Majesty the Queen Elizabeth II, are the Raphael Cartoons: the seven surviving (there were ten) full scale designs for tapestries in the Sistine Chapel, of the lives of Peter and Paul from the Gospels and the Acts of the Apostles. There is also on display a fresco by Pietro Perugino dated 1522 from the church of Castello at Fontignano (Perugia) and is amongst the painter's last works. One of the largest objects in the collection is the Spanish tempera on wood, 670 x 486 cm, retable of St George, c. 1400, consisting of numerous scenes and painted by Andrés Marzal De Sax in Valencia. 1130 is a number. 650 is a number. 6800 is a number. 2000 is a number. 1522 is a number. 670 is a number. 486 is a number. 1400, is a number.*"
Question: "*Approximately how many British oil paintings does the museum have?*"
Original Prediction: 1130 British and 650
Prediction under adversary: 1130

Figure 5: Adversarial data augmentation applied to numeric examples. The context is extended to include information about which numbers are present.

In some cases, the accuracy can improve slightly, which we will see next. ELECTRA was released in 2020, and we used adversarial data augmentation to slightly improve the accuracy of the SQUAD dataset instead of diminishing it.

3.2 Numeric Adversarial Augmentation

We decided to augment training examples from the SQUAD dataset with a numeric value in their answers list.

We followed the same procedure as Jia and Liang, in hopes that it would allow the model to recognize digits better.

In the figure above, we highlight our changes in blue. We first loop over the training data and find examples with a numeric value in their answer set. We then go through the context and find all occurrences of numbers. The format *x is a number* is appended to the end of the context for each number that exists inside. Note that if the number were presented with a suffix, for instance, *15%*, this would also be appended as *15% is a number*.

We finally joined all the augmented sentences with periods to complete the passage. This does not change the overall meaning of the context, but it does help the model recognize better what to consider as the answer.

Using this approach, the model was able to answer questions correctly like:

When did the Jin dynasty end?

from *1115-1234* to *1234*.

Another example is

What is Harvard's total financial aid reserves?

The model previously classified *$159 million for students*, and now correctly classifies as simply *$159 million*.

One more example is

How long was the first audio of a Doctor Who story?

The model previously thought the answer was *Ten years*, but now returns *21-minute*, which shows how it favours answers with numeric values.

Using this approach, the model not only selects numeric values as the answer, but also selects the proper span length too. Instead of extending the span, the model only finds the numeric value and returns that as the answer.

4 Results

We trained the dataset using adversarial augmented data over 3 epochs. Our accuracy (exact match) increased from 78.37% to 78.95%. The F1 score also increased from 86.12% to 86.45%. The increase in accuracy is because the model can answer questions with numeric outputs better than before. The model can better understand what a number is and return the appropriate length of the span. One area that we could have improved on was how we were determining the list of numbers available in the context passage. For instance, numbers in the context were represented as roman numerals, temperatures, feet, meters, cubic meters, etc. It was impossible to account for every case, and as such, we likely missed some examples that should have been augmented. In our augmentation, we determined if a piece of text was a number by discarding the characters *, % $* and seeing if what remains are only digits.

Dataset	**Accuracy**	**F1 score**
SQUAD	78.37%	86.12%
Augmented SQUAD	78.95%	86.45%

Table 1: Summary of results

For instance, *10,000* would be considered as the number 10000, and similarly, *25%* would be considered as 25. However, we would not have treated *15°* C as a number. Overall, we are happy with our results as we could not account for every case, and we leave this task for future readers.

5 Conclusion

In this paper, we examined the task of Question Answering in NLP and how it can be applied to the SQUAD dataset, which contained about 100,000 crowdsourced examples. We utilized a pre-trained model called ELECTRA, and analyzed the initial results of training on the SQUAD dataset. We noted that the model had an initial accuracy of 78.37% and an F1 score of 86.12%. In our analysis, we discovered that the model performed poorly on examples which had a long answer or contained numeric values in the answer. We followed up on the work by Jia and Liang on adversarial augmentation and applied the technique to our dataset. We augmented our context to contain numeric data at the end of the passage. After training on 3 epochs, our accuracy increased from 78.37% to 78.95%, and the F1 score increased from 86.12% to 86.45%. We ended by mentioning limitations in our approach and how future readers can follow up on it.

References

[1] Jia, R., & Liang, P. (2017, July 23). *Adversarial examples for evaluating reading comprehension systems*. arXiv.org. Retrieved from https://arxiv.org/abs/1707.07328

[2] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016, October 11). *Squad: 100,000+ questions for machine comprehension of text*. arXiv.org. Retrieved from https://arxiv.org/abs/1606.05250

[3] Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., & Smith, N. A. (2018, April 16). *Annotation artifacts in Natural Language Inference Data*. arXiv.org. Retrieved from https://arxiv.org/abs/1803.02324

[4] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May 24). *Bert: Pre-training of deep bidirectional Transformers for language understanding*. arXiv.org. Retrieved from https://arxiv.org/abs/1810.04805v2

[5] Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020, March 23). *Electra: Pre-training text encoders as discriminators rather than generators*. arXiv.org. Retrieved from https://arxiv.org/abs/2003.10555

[6] Wang, Z. (2022, June 30). *Modern question answering datasets and benchmarks: A survey*. arXiv.org. Retrieved from https://arxiv.org/abs/2206.15030