CS388 Natural Language Processing - Final Project

Enhancing Question Answering Performance with Adversarial Training: A Case Study on the ELECTRA-small Model and SQuAD Dataset

Abstract

This project explores strategies to enhance the performance of the ELECTRA-small model on the Stanford Question Answering Dataset (SQuAD). The study identifies two predominant error types: Misinterpretation Errors and Numeric Errors. To address these, we incorporate adversarial examples into the training data, specifically focusing on synonym and numeric-based adversarial data. The model is retrained and reevaluated using four distinct methods, each designed to assess the model's performance in different aspects. The results show that retraining the model on the original SQuAD data in addition to the new adversarial datasets surpasses the original SQuAD performance. The study concludes with a discussion on potential future research directions, including the exploration of other models and the implementation of additional adversarial data types.

1 Introduction

In this project, we aim to address the challenges associated with the complexity and syntactic divergence in question-answering tasks using Natural Language Processing (NLP). Our approach involves the use of the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016), a reading comprehension dataset built by Stanford University researchers in 2016. The SQuAD dataset was constructed to facilitate the training and evaluation of question-answering models in a real-world context.

The SQuAD dataset comprises 107,785 question-answer pairs on 536 Wikipedia articles, with no answer choices provided. These articles were randomly sampled from the top 10,000 articles of English Wikipedia, as ranked by Project Nayuki's Wikipedia's internal PageRanks. The questions and answers in the dataset were generated by humans, resulting in a diverse range of linguistic structures and real-world contexts.

The evaluation of models trained on the SQuAD dataset is typically based on F1 scores, a measure of test accuracy that considers both precision and recall. However, despite its strengths, the SQuAD dataset has some weaknesses. For instance, model performance tends to worsen with increasing complexity of answer types and syntactic divergence between the question and the sentence containing the answer.

In this project, we first establish a baseline by running the vanilla ELECTRA-small model on the SQuAD dataset. This provides us with a base result to compare subsequent modifications against. We then perform an error analysis to identify areas for improvement. This analysis focuses on the types of errors the model is making and how these errors can be reduced. Recognizing the limitations of the F1 score as the sole evaluation metric, we explore alternative metrics. These alternatives could provide a more comprehensive understanding of model performance. To enhance the model itself, we incorporate adversarial data into our training set. This aims to make the model more robust and capable of handling complex examples. Finally, we evaluate the impact of these modifications on the model's performance. We provide a detailed analysis of the results and discuss potential future directions for this work.

2 Baseline Result

For our initial experiment, we utilized the ELECTRA-small pre-trained model (Clark et al., 2020) from the Hugging Face's transformers library. ELECTRA models are efficient transformer models that pre-train by discriminating replaced tokens in a sentence, rather than predicting masked tokens like in BERT. The 'small' variant of ELECTRA is a smaller, faster version that still delivers competitive performance on many tasks.

We used this pre-trained model directly for the QA task on the SQuAD dataset. The results were promising, with an Exact Match score of 78.7% and an F1 score of 86.4%. These scores are significantly higher than the results reported in the original SQuAD paper by Rajpurkar et al., 2016, where a Logistic Regression model achieved an Exact Match score of 40.4% and an F1 score of 51.0%. Our results are also very close to the human performance scores reported in the same paper, which

were 77.0% and 86.8% respectively. We present a comparison of these results in Table 1.

	Exact Match	F1
Logistic Regression	40.4%	51.0%
Human	77.0%	86.8%
ELECTRA-small	78.7%	86.4%

Table 1: Comparison of Exact Match and F1 scores

3 Analysis of Baseline Result

In our pursuit to refine the performance of the ELECTRA-small model, we meticulously analyzed the errors it produced. Among the 10,570 predictions made on the test set, we found 869 examples where the F1 score was less than 0.5.

3.1 Error Type Analysis

In this section, we will explore the different types of errors that emerged in the baseline results. For each error type, we provide a brief description, an example, and discussion of any noticeable trends or patterns.

Misinterpretation Error

A Misinterpretation Error is a type of mistake where the model's predicted answer makes sense in a general context but is incorrect in the specific context of the question. This can occur when the model picks up on the wrong aspect of the question or evidence.

> **Context:** When rock units are placed under horizontal compression, they shorten and become thicker. [...] These folds can either be those where the material in the center of the fold buckles upwards, creating "antiforms", or where it buckles downwards, creating "synforms".
>
> **Question:** When rock folds deep in the Earth it can fold one of two ways, when it buckles downwards it creates what?
>
> **Model Prediction:** antiforms
>
> **Gold Label:** synforms

In this example, the model misinterprets the context and predicts the incorrect term. In our analysis, we found 548 instances of Misinterpretation Errors, which constitute 63.1% of the total mistakes.

Numeric Error

A Numeric Error is a type of mistake where the model's predicted answer is incorrect due to misinterpretation of numeric information in the question or context. This can occur when the model understands that a numeric answer is required but selects the wrong numeric value or uses the incorrect unit.

> **Context:** The revived series has received recognition from critics and the public, across various awards ceremonies. It was very popular at the BAFTA Cymru Awards, with 25 wins overall including Best Drama Series (twice), Best Screenplay/Screenwriter (thrice) and Best Actor.
>
> **Question:** How many BAFTA Cymru Awards has Doctor Who received?
>
> **Model Prediction:** five
>
> **Gold Label:** 25

In this example, the model misinterprets the numeric information in the context and predicts the incorrect number. In our analysis, we found 119 instances of Numeric Errors, which constitute 13.7% of the total mistakes.

Irrelevant Answer Error

An Irrelevant Answer Error is a type of mistake where the model's predicted answer is not relevant or related to the question or context. This can occur when the model completely misinterprets the question or context, resulting in an answer that is off the point or does not belong to the category of the expected answer.

> **Context:** Oxygen storage methods include high pressure oxygen tanks, cryogenics and chemical compounds. For reasons of economy, oxygen is often transported in bulk as a liquid in specially insulated tankers, since one liter of liquefied oxygen is equivalent to 840 liters of gaseous oxygen at atmospheric pressure and 20 °C (68 °F).
>
> **Question:** By what means is bulk oxygen shipped?
>
> **Model Prediction:** reasons of economy
>
> **Gold Label:** insulated tankers

In this example, the model's prediction is completely irrelevant to the question and context. In our analysis, we found 110 instances of Irrelevant Answer Errors, which constitute 12.7% of the total mistakes.

Question Ambiguity Error

A Question Ambiguity Error is a type of mistake where the model's predicted answer is incorrect due to the ambiguity of the question. This can occur when the question is not clear or specific enough, causing the model to misinterpret what is being asked.

> **Context:** Like many cities in Central and Eastern Europe, infrastructure in Warsaw suffered considerably during its time as an Eastern Bloc economy. [...] In particular, the city's metro, roads, sidewalks, health care facilities and sanitation facilities have improved markedly.
>
> **Question:** Warsaw's sidewalks and sanitation facilities are some examples of things which have what?
>
> **Model Prediction:** health care
>
> **Gold Label:** improved markedly

In this example, the ambiguity of the question leads the model to predict an incorrect answer. In our analysis, we found 34 instances of Question Ambiguity Errors, which constitute 3.9% of the total mistakes.

Generalization Error

A Generalization Error is a type of mistake where the model's predicted answer either includes too much detail (overgeneralization) or not enough detail (undergeneralization) compared to the correct answer. This can occur when the model either omits necessary details or includes unnecessary ones.

> **Context:** Historically, Victoria has been the base for the manufacturing plants of the major car brands Ford, Toyota and Holden; however, closure announcements by all three companies in the 21st century will mean that Australia will no longer be a base for the global car industry.
>
> **Question:** What type of manufacturing plant is Victoria soon losing?
>
> **Model Prediction:** Ford, Toyota and Holden
>
> **Gold Label:** major car brands

In this example, the model's prediction includes too much detail, naming the specific car brands instead of generalizing to "major car brand" as in the correct answer. In our analysis, we found 24 instances of Generalization Errors, which constitute 2.8% of the total mistakes.

Alternative Correct Answer Error

An Alternative Correct Answer Error is a type of mistake where the model's predicted answer is correct but differs from the ground truth answer, even though both are valid. This can occur when multiple correct answers exist but the evaluation metric requires an exact match with the ground truth.

> **Context:** CBS set the base rate for a 30-second advertisement at $5,000,000, a record high price for a Super Bowl ad. [...] Nintendo and The Pokémon Company also made their Super Bowl debut, promoting the 20th anniversary of the Pokémon video game and media franchise.
>
> **Question:** Which video gaming company debuted their ad for the first time during Super Bowl 50?
>
> **Model Prediction:** Pokémon
>
> **Gold Label:** Nintendo

In this example, both Nintendo and The Pokémon Company made their Super Bowl debut, so both answers are correct. However, the evaluation metric requires an exact match with the ground truth, leading to an Alternative Correct Answer Error. In our analysis, we found 18 instances of Alternative Correct Answer Errors, which constitute 2.1% of the total mistakes.

Representation Error

A Representation Error is a type of mistake where the model's predicted answer and the correct answer represent the same concept but in different forms or notations.

> **Context:** In addition, there are $2 million worth of other ancillary events, including a week-long event at the Santa Clara Convention Center, a beer, wine and food festival at Bellomy Field at Santa Clara University, and a pep rally. [...] Additional funding will be provided by the city council, which has announced plans to set aside seed funding for the event.
>
> **Question:** How long will the event at Santa Clara Convention Center last?
>
> **Model Prediction:** week-long
>
> **Gold Label:** a week

In this example, the model's predicted answer and the correct answer represent the same concept (the duration of the event), but in different forms. In our analysis, we found 11 instances of Representation Errors, which constitute 1.3% of the total mistakes.

Semantic Error

A Semantic Error is a type of mistake where the model's predicted answer is semantically related to the correct answer but not exactly the same. This includes cases where the predicted word is a different form or transformation of the correct word or when crucial components like negation are missed.

Context: Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. [...] 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 L"), so that the logo could prominently feature the Arabic numerals 50.

Question: What color was used to emphasize the 50th anniversary of the Super Bowl?

Model Prediction: golden

Gold Label: gold

In this example, the model's predicted answer is semantically related to the correct answer but is a different form of the correct word. In our analysis, we found 4 instances of Semantic Errors, which constitute 0.5% of the total mistakes.

Incorrect Ground Truth Error

An Incorrect Ground Truth Error is a type of mistake where the model's predicted answer is marked as incorrect due to an error in the ground truth label. This can occur when the provided answer in the dataset is incorrect, causing the model's correct answer to be marked as incorrect.

Context: Luther and his wife moved into a former monastery, "The Black Cloister," a wedding present from the new elector John the Steadfast (1525–32). They embarked on what appeared to have been a happy and successful marriage, though money was often short.

Question: When did Luther and his wife live?

Model Prediction: 1525

Gold Label: The Black Cloister

In this example, the model's prediction is correct according to the context, but it is marked as incorrect due to an error in the ground truth label. In our analysis, we found 1 instance of Incorrect Ground Truth Errors, which constitute 0.1% of the total mistakes.

3.2 Summary of Error Type Analysis

After analyzing the different types of errors, we found that Misinterpretation Errors were the most common, making up 63.1% of the total mistakes. Numeric Misinterpretation Errors and Irrelevant Answer Errors also made up a significant portion of the errors, at 13.7% and 12.7% respectively.

Further analysis of the Misinterpretation Errors revealed several subcategories. For instance, the model often got confused when handling person's names, locations, or other proper nouns.

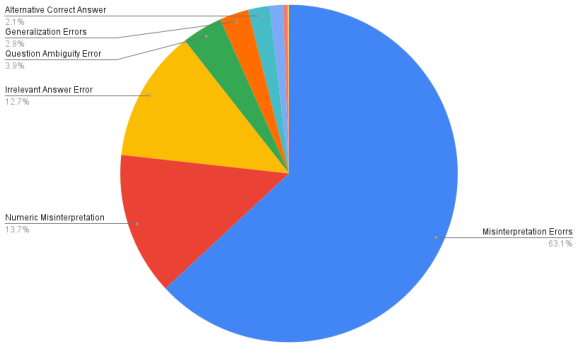

Figure 1: The breakdown of error types

The pie chart in Figure 1 illustrates the proportion of each error type among the total errors in our baseline results.

Based on these findings, we propose two main enhancements to improve the model's performance and robustness. These enhancements are designed to address the specific error types identified in our analysis.

4 Alternative Evaluation Metrics

In our analysis of the 869 erroneous predictions, we observed that most of them had an F1 score of 0, indicating no overlap between the model's prediction and the gold label. The F1 score measures the harmonic mean of precision and recall on the token (word) level (Yacouby and Axman, 2020). It is calculated based on the number of true positives (matches between the model's prediction and the gold label), false positives (words predicted by the model that are not in the gold label), and false negatives (words in the gold label that the model did not predict).

However, as demonstrated in the examples of Representation Error and Semantic Error, the model often produces answers that are very close to the correct answer, but these are considered complete mistakes based on their F1 score. This suggests that while the F1 score works in many cases, it may not necessarily be the best measure of model performance for this task.

Aside from Irrelevant Answer Errors, many of the mistakes made by the model are somewhat sensible, suggesting that a metric that can account for near-misses might provide a more balanced evaluation of the model's performance. To address this, we have considered several alternative metrics that

could potentially provide a more nuanced evaluation, including ROUGE and METEOR scores (Lin, 2004).

Among these, the metric that we have decided to explore further is METEOR. This metric has been widely used in the field of Natural Language Processing and could potentially provide a more nuanced evaluation of the model's performance. In the following section, we delve deeper into the METEOR metric, its components, and how it can provide a more nuanced evaluation of our model's performance.

4.1 METEOR

The Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Banerjee and Lavie, 2005) combines precision, recall, and alignment with synonyms. It includes stemming and synonym matching to provide a more nuanced evaluation. METEOR is based on a generalized concept of unigram matching between the machine-produced translation and human-produced reference translations. Unigrams can be matched based on their surface forms, stemmed forms, and meanings. METEOR computes a score for this matching using a combination of unigram-precision, unigram-recall, and a measure of fragmentation that is designed to directly capture how well-ordered the matched words in the machine translation are in relation to the reference.

To illustrate how METEOR works, consider an example from the Semantic Error category discussed in the previous section. Suppose the model prediction is "golden", but the gold label is "gold". In this case, the METEOR score is 0.5, indicating that the prediction and reference are somewhat similar. This is because METEOR uses stemming, and the stemmed forms of "golden" and "gold" are the same. On the other hand, the F1 score is 0, indicating that the prediction and reference are completely dissimilar. This is because F1 only considers exact word matches and does not account for stemming.

Despite these differences in scoring methods, our evaluation found that the F1 score still prevailed when applied to a larger dataset.

4.2 Evaluation Results

The scores were calculated based on a test set of 10,570 instances. The results for the entire evaluation set were as follows:

	Score
F1	86.4%
METEOR	61.9%

Table 2: Comparison of F1 and METEOR scores

In our evaluation of the METEOR score, we found two significant issues that impact its effectiveness.

Issue 1: Exact Match Caught by F1 but Not by Meteor

The first issue is related to the internal tokenization methods of the METEOR score. These methods prevent it from awarding a full 100% score even when there is an exact match. This issue is evident in our data, where none of the instances with a 100% F1 score received a 100% METEOR score.

Description	**Rate**	**Counts**
METEOR = 100%	0.0%	0
90% ≤ METEOR < 100%	48.1%	4,005
50% ≤ METEOR < 90%	38.6%	3,216
METEOR < 50%*[1]	13.2%	1,102
Total	100.0%	8,323

Table 3: METEOR score Distribution for Exact Matches

Issue 2: Failure of F1 caught by METEOR

The second issue arises when the F1 score is 0%. In these cases, the METEOR score is often too low to compensate for the decrements and uncertainty that the algorithm incorporates for other examples. This issue is particularly noticeable in our data, where a majority (95.24%) of the instances with a non-scoring F1 score received a METEOR score of less than 50%.

Description	**Rate**	**Counts**
METEOR ≥ 50%	4.8%	2
METEOR < 50%	95.2%	40
Total	100.0%	42

Table 4: METEOR score Distribution for F1 Failures

While the METEOR score has proven effective in identifying certain types of errors, it may not be the perfect metric for the entire dataset. Our goal is to find a model that aligns with human judgment, and this quest may require further refinement of our evaluation metrics.

In addition to refining our evaluation metrics, we also propose enhancements to the training data itself.

[1]For examples with a 0% F1 score, METEOR scores below 50% generally indicate that the predicted answer/model is incorrect, not that the predicted answer is misclassified as incorrect.

5 Adversarial Data Augmentation

Another enhancement we propose is to augment the training data with adversarial examples, an area that has garnered considerable attention in recent NLP research (Jia and Liang, 2017; Liu et al., 2020; Wallace et al., 2019). Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake. By training the model on adversarial examples, we can expect it to learn to generalize better and be more robust to errors.

In our investigation of adversarial datasets, our principal objective was the mitigation of two predominant error types: Misinterpretation Errors (63.1% of the errors) and Numeric Errors (13.7% of the errors). After investigating multiple adversarial datasets, we identified the SQuAD Adversarial Dataset sourced from Hugging Face from Jia and Liang (2017) as a larger source from which to subset data from to address the aforementioned error categories. This choice was driven by the dataset's advantageous alignment with our desired error categories and its parallel formatting, ensuring a consistent learning environment for our analytical processes.

5.1 Extracting Data

From the SQuAD Adversarial Dataset, we focused on extracting two subsets of additional training data to mitigate Misinterpretation and Numeric Errors, respectively.

Misinterpretation Errors In the context of addressing misinterpretation errors, we identified that enhancing the model's grasp of the intrinsic meaning of sentences could significantly bolster its predictive accuracy. To achieve this objective, we adopted a methodology centered around the incorporation of synonyms into the answer inputs the model is trained on.

First, we shuffled the SQuAD Adversarial Dataset, aiming to introduce variability and diversify the sentence structures encountered during training. Then, we selectively filtered examples where the answer length exceeded 10 words, limiting the dataset to 500 examples. The answer length capping mechanism was implemented to ensure the model was not excessively challenged and retained its foundational learned policies.

The data extracted contained three parts: the question, the context from which the model predicted an answer, and the answer. The answer can further be divided into both the text portion of the answer and the answer start index within the context. Following the original data extraction, we manually substituted 1-3 words per answer with synonyms while preserving the inherent semantic meaning. We abstained from altering key words, such as proper nouns and critical details.

See the below examples of data collected. Only the question, the original answer, and the modified answer are shown for conciseness.

Question:
How many people think that Valencian is different than Catalan?
Original Answer:
the majority of the Valencian **people**
Modified Answer:
the majority of the Valencian **citizens**

Question:
What is the point of the right?
Original Answer:
This **enables** the members of the parliament to observe the activities of the executive power and the above-mentioned high officials of the state
Modified Answer:
This **allows for** the members of the parliament to observe the activities of the executive power and the above-mentioned high officials of the state

Question: When was the Holocaust?
Original Answer:
prior to and during World War II
Modified Answer:
before and during World War II

Numeric Errors In the context of addressing numeric errors, we identified that adding additional examples with numeric answers could enhance the model's predictive accuracy. In many cases, examples with numeric answers had contexts which contained multiple numbers from which the model could choose. Adding numeric-centric data enabled the model to discern not only the presence of numerical values but also to consider contextual nuances surrounding such values.

Due to the fast searching conditions and the absence of manual augmentation following data extraction, we were able to iterate through the entire SQuAD Adversarial Dataset of 30,000 training examples. During this process, we identified and extracted all instances where the answers contained numeric values, yielding a total of 2,707 examples. The data extracted contained the same three parts as detailed above: the question, the context from which the model predicted an answer, and the answer. The answer can further be divided into both the text portion of the answer and the answer start index within the context.

See the below examples of data collected. Only the question and answer are shown for conciseness.

Question:
Which election was the Contract with America unveiled closest to?
Answer: the 1994 midterm elections

Question:
When was the most crowded election held according to the text?
Answer: 1855

Question:
At what point did Florida finally have fairly drawn congressional districts?
Answer: December 2015

After extracting these data subsets, we retrained our original ELECTRA-small pre-trained model and reevaluated against the test data via four distinct methods. Note that in the second two methods, we reincorporate unmodified data in an effort to prevent the model from exclusively training on challenging examples, thereby promoting a more diverse and balanced training set.

5.2 Retraining and Reevaluating Data

Following the data extraction and augmentation, we proceeded to retrain our ELECTRA-small model. The retraining was conducted using the same parameters as the initial training, ensuring a fair comparison of the results.

Upon completion of the retraining, we reevaluated the model against the test data. This was done using four distinct methods, each designed to assess the model's performance in different aspects and to validate the effectiveness of our data augmentation strategies.

In the following sections, we will discuss the results of this reevaluation and the impact of our enhancements on the model's performance.

Synonym Data In this method, we trained our model using the synonym-based adversarial data (500 examples). This training was conducted on top of our initial pre-trained model, which had an accuracy of 86.4%.

Numeric Data In this method, we trained our model using the numeric-based adversarial data (2,707 examples). This training was also conducted on top of our initial pre-trained model, which had an accuracy of 86.4%.

Partial SQuAD (11.5%), Synonym, and Numeric Data In this method, we shuffled our original SQuAD dataset and subset 10,000 out of the existing 87,599 training examples, equivalent to roughly 11.49% of the original set. We then added our synonym-based adversarial data (500 examples) and numeric-based adversarial data (2,707 examples) to create a new dataset with 13,207 total training examples. We trained this data on top of our initial pre-trained model, which had an accuracy of 86.4%.

Partial SQuAD (37.5%), Synonym, and Numeric Data In this method, we shuffled our original SQuAD dataset and subset 30,000 out of the existing 87,599 training examples, equivalent to 37.5% of the original set. We then added our synonym-based adversarial data (500 examples) and numeric-based adversarial data (2,707 examples) to create a new dataset with 33,207 total training examples. We trained this data on top of our initial pre-trained model, which had an accuracy of 86.4%.

5.3 Results and Comparison

Of the four retraining methods, retraining our model on 30,000 examples from the original SQuAD data in addition to our new adversarial datasets surpassed the original SQuAD performance, with an exact match score of 78.91% and an F1 score of 87.29%. A summary of the three results is shown below.

Method	Exact Match	F1
Synonym Data	27.23%	42.58%
Numeric Data	57.89%	69.63%
Partial SQuAD*[2]	75.43%	84.09%
Partial SQuAD*[3]	78.91%	87.29%

Table 5: Summary of results for the four retraining methods

5.4 Analysis of Results

The initial findings indicate that the first two implementations (Synonym Data and Numeric Data) yielded notably lower scores compared to the original model. This outcome aligns with expectations, as these implementations exclusively involved training on challenging or outlying examples. For the synonym data, there are possibly instances in which the substitution of a crucial keyword for the model negatively impacted the model's ability to identify the correct answer. Similarly, in the case of numeric data, the substantial decrease can be attributed to the training of all examples of one specific kind without the opportunity for subsequent model normalizations through additional data training. The last two implementations (Partial SQuAD (11.5% and 37.5%), Synonym, and Numeric Data) demonstrated performance metrics closely aligned with the original model, with the 37.5% reincorporation method surpassing the original model's exact match and F1 score.

Due to the performance improvement using the Partial SQuAD (37.5%), Synonym, and Numeric

[2]Trained on 11.5% of SQuAD data, with synonym-based and numeric-based adversarial data. The total training examples used were 13,207.

[3]Trained on 37.5% of SQuAD data, with synonym-based and numeric-based adversarial data. The total training examples used were 33,207.

Data, we focused our analysis on this dataset. Among the 10,570 predictions made on the test set, we found 779 examples where the F1 score was less than 0.5. From these, we analyzed only 500 examples to get a summary of results. From these examples, we applied the same error classification criteria outlined in Section 3.1 for the initial ELECTRA-small model trained on the original SQuAD dataset. See Figure # for a breakdown of the results.

As detailed in Section 3.2, the initial ELECTRA-small model exhibited two prevalent error types on the original SQuAD dataset: Misinterpretation Errors (63.1% of errors) and Numeric Errors (13.7% of errors). After introduction of targeted adversarial data, our analysis revealed a notable reduction in error rates to 29.2% and 12.2%, respectively. This observed decrease underscores the efficacy of training on a more diverse and resilient dataset, tailored to address identified weaknesses within both the model and the dataset itself. The decrease in Misinterpretation Errors was more substantial compared to Numeric Errors, suggesting that the adversarial examples introduced may have been particularly effective in mitigating misinterpretation challenges.

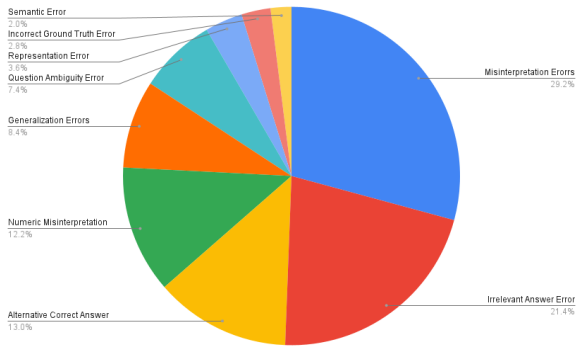

Figure 2: The breakdown of error types

In this updated training iteration, apart from observing reduced Misinterpretation Errors (29.2%) and Numeric Misinterpretation Errors (12.2%), notable occurrences of Irrelevant Answer Errors (21.4%) and Alternative Correct Answer Errors (13.0%) have emerged. The introduction of synonym data may have contributed to the model's diminished precision in identifying the correct word or exact answer. Instead, the model appears inclined to output alternative correct answers or other segments of the sentence, possibly interpreted as synonyms, reflecting the nuanced impact of synonym data on the model's response generation.

6 Conclusion

In this study, we undertook a comprehensive exploration of strategies to enhance the performance of the SQuAD dataset. We focused on the challenges posed by limitations in computing power and time constraints. Despite these constraints, our findings illuminate several avenues for further investigation and improvement.

Model Exploration Another key element of model performance is the model architecture itself. While our study utilized the ELECTRA-Small model, models such as BERT, GPT, and RoBERTa have shown efficacy in many related natural language processing tasks (Wallace et al., 2019; Liu et al., 2019). Investigating their performance on the SQuAD dataset could offer valuable insights and potentially uncover models better suited to the nuances of our dataset.

Comprehensive Adversarial Data Integration A significant limitation of our study was the inability to retrain the entire model due to computing power constraints. In our analysis, retraining using 30,000 examples from the original SQuAD data yielded markedly better results compared to retraining in the same method with only 10,000 examples. Future efforts could explore the implementation of adversarial data from the start of model training, allowing for a more comprehensive integration of adversarial examples.

Addressing Other Error Types Our study primarily focused on synonym and numeric-based adversarial data to target specific error types. However, Figure 1 shows the presence of additional error types, such as irrelevant answers, question ambiguity, and generalization errors, that could be targeted with tailored adversarial datasets or similar strategies.

In conclusion, while our study represents a step towards augmenting SQuAD dataset performance, the identified next steps underscore the ongoing nature of research in natural language processing.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

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 International Conference on Learning Representations (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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Reda Yacouby and Dustin Axman. 2020. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the first workshop on evaluation and comparison of NLP systems*, pages 79–91.