# Typographical Errors in Question Answering

Tom Fu
tomfu21@utexas.edu
zf2778

Faraz Heravi
farazh@utexas.edu
fm9793

## Abstract

This paper focuses on the real-world Question Answering scenarios, where sometimes answers are not found, and typos or misspelled words are common. Our goal is to improve a pre-trained model trained on the SQuAD 2.0 dataset by modifying the data such that it includes typographical spelling errors. Using Electra Model pre-trained weights from Deepset.ai, we tested the model's performance on the original unaltered SQuAD 2.0 dataset. We then altered the SQuAD 2.0 dataset and created an augmented dataset, which contains more than 100 thousand questions with misspelled words and typos. When evaluating the same model on the adversarial dataset, the F1 score dropped significantly. This paper further analyzes the two separate challenge tasks, word deletion and typos, to compare the differences in the model's performance decline. Lastly, this paper proposes and discusses a few solutions to increase model performance against datasets with spelling errors, and it finds out that focus training with the adversarial dataset is in fact the simplest solution.

## 1 Introduction

The previous work on Question Answering (QA) by Jia and Liang suggested that recent improvements on language systems require more scrutiny and understanding [Jia and Liang, 2017]. When adding some trivially distracting sentences to the question's context, even the State of Art models seemingly failed to find the right answers, and model performances dropped drastically. Perhaps, the QA models are finding spurious language patterns or artifacts within the context to achieve a high F1 score. Subsequently, other related literature also aimed to produce some Universal Adversarial Triggers, such as irrelevant sentence or grammatically incorrect answer triggering words, added to the context that would cause QA models to break [Wallace et al., 2019].

However, in real life scenarios when asking a question or searching on a search engine, humans usually do not have the access to the full context. Instead, humans would come up with the questions that QA models would use to query and search for the answer from the context. Besides modifying the context given to the QA models, questions are also another input of the models which have less research on, and the augmented data comes from the questions more often than from the context. The questions that humans type into a search engine are not necessarily perfect questions, whereas the context knowledge database is often carefully maintained and updated regularly to remove misleading or irrelevant information.

This paper aims to test whether modifying the QA questions with typographical errors would break the language pattern and the artifact that QA models link between questions and the context. We also aim to mitigate the issue of typographical errors by training a model with an augmented SQuAD 2.0 dataset.

## 2 Model

Due to the constraints of our resources, we decided to use a pre-trained model for our paper. We used Deepset.ai's electra-base-squad2[1] model which is specifically trained on SQuAD 2.0 for Extractive QA. Electra-base-squad2 is trained using Google's electra-base-discriminator on 5 epochs, a batch size of 5, a max sequence length of 384, max query length of 64, and a learning rate of 1e-4. This model utilizes a linear warm up learning rate scheduler with a warm up proportion of 0.1. The model's official evaluation has a performance with F1 81.35 on the SQuAD 2.0 dev set using the official evaluation script.

[1]You can find more information about Deepset's model here: https://huggingface.co/deepset/electra-base-squad2

## 3 Data

We use the SQuAD 2.0 dataset[2] to evaluate and create challenge sets. According to the authors, this dataset is different than SQuAD 1.1 because it contains over 50,000 unanswerable questions. We chose SQuAD 2.0, because our paper wants to mimic real-life scenarios where not all questions are answerable. The SQuAD 2.0 dataset comes with an official evaluation script which evaluates answers of a given model [Rajpurkar et al., 2018].

### 3.1 Dataset

We created a parallel SQuAD 2.0 data for the train and dev sets using data augmentation. We utilized the TextAttack Python framework[3] to create these datasets [Morris et al., 2020], and modified only the questions of the SQuAD 2.0 while the contexts and answers remained the same. For these datasets, two types of word transformations were used from the TextAttack framework: random character swap and random character deletion. The swaps are equivalent to typos, and a character would be swapped with a surrounding character on a computer keyboard. One important parameter of this data modification was the rate of typographical error. We chose this to be 30% of all words. This means that approximately, three out of ten words will have at least one character swapped or deleted. The frequency of spelling errors in human typed text varies from 0.05% in a carefully edited newswire to 38% in difficult applications such as telephone directory lookup. We selected a relatively higher percentage because accuracy is not an important factor while typing questions into search engines quickly. But also, at 30% the question is still easily identifiable by a human in most cases.

## 4 Pre-Trained Analysis

This section explores the analysis of the data augmentation and the performance of the Deepset's electra-base-squad2 on augmented and unaltered SQuAD 2.0's dataset.

### 4.1 Evaluation

We created three different datasets by sampling 800 QA examples from the original SQuAD 2.0 dev set.

- SQuAD 2.0: unaltered sample of the SQuAD 2.0 dev set
- Augmented: applied TextAttack to the same sample of the SQuAD 2.0 dev set
- Random: randomly selected samples from SQuAD 2.0 and augmented with a probability of 50% for each

We used the pre-trained model to generated answers for the 800 questions given their contexts and ran the official SQuAD 2.0 evaluation script to determine the F1 and exact scores of our predicted answers.

### 4.2 Results

After running the official evaluation script on the three prediction sets, we reached the following results:

| | **F1** | **Exact** |
|---|---|---|
| **SQuAD 2.0** | 87.22 | 80.00 |
| **Augmented** | 68.82 | 56.88 |
| **Random** | 79.55 | 70.50 |

Although Deepset's model is trained on unanswerable questions as well, based on our observation, unanswerable questions still caused issues and confusions for the model. We believe that the higher F1 score for our subset compared to Deepset's official F1 score comes from the fact the percentage of unanswerable questions in our evaluation sample was lower than the percentage of unanswerable questions in the entire dev set. This does not affect the analysis because we will be comparing the F1 scores of the same subset relatively to each other.

The augmented data which contains typed spelling errors has a much lower F1 score than the original dataset, and the random data resulted in an F1 score in between. Also, the exact scores from the table correlate with the F1 score. The higher the exact score, the higher the F1 score we get for all of the datasets.

Many times, the pre-trained model confused the answers for questions with typographical errors. For example, the question *French Church Street is in what Irish town?* was modified to *French Chhrch Etreet is in what Irish town?*. The model was able to recognize that we are looking for something that has to do with Church and it is located in an Irish town, thus it gave the answer *Portarlington*. This answer comes from the fact that Portarlington was a French Church built in the town according to part of the context.[4] But the answer we are looking for is a town called

[2]The Stanford Question Answering Dataset: https://rajpurkar.github.io/SQuAD-explorer/

[3]TextAttack Python Framework: https://textattack.readthedocs.io/en/latest/

[4]Due to the contexts being long, we do not mention them in this paper, but the questions and their contexts can be found from the dev SQuAD 2.0 dataset.

Cork City, and due to the word *Street* having errors, the model could not identify this.

## 5 Approach

Following the suggested approach by relevant literature [Liu et al., 2019] [Zhou and Bansal, 2020] [Morris et al., 2020], we first conduct focus-training on the augmented data directly. We train an additional fine-tuned model to specially handle questions with spelling errors, whereas the original model handles normal questions.

We have also considered an ensemble approach by labeling adversarial data and normal data and training a shallow classification network. Then we could put this classification network on top of the original model and the augmentation fine-tuned model to sort questions into augmented questions or normal questions, and then send these questions to the matching fine-tuned models. However, after fine-tuning with spelling error data, the augmented fine-tuned model performs better than the original model on both sets of questions. Hence the specialized models handling different questions is not as effective as one single fine-tuned model.

### 5.1 Fine-tuning

In order to fine-tune the pre-trained model on the augmented data, we tokenized the augmented questions and their context with truncation and padding using the hugging face transformers library. We also added the start and end positions of the answers as the label. Sometimes the SQuAD 2.0's answers were off by one or two positions, thus we had to adjust for that as well. Then we fine-tuned the Deepset model using a batch size of 8 for 1 epoch. The optimizer used was AdamW with a learning rate 5e-5. Although we had the entire data set augmented, only 25000 sample questions were selected randomly from the SQuAD training dataset and used for training due to the computational power required.

## 6 Fine-tuned Analysis

This section displays and explains the results obtained from the fine-tuned and pre-trained model applied to different datasets.

### 6.1 Results

The fine-tuned model was evaluated on the same 800 subset of the SQuAD 2.0 dev set. The training with the previously mentioned hyperparameters was effective enough to achieve better results instantly. Below are the results obtained:

| | **F1** | **Exact** |
|---|---|---|
| **SQuAD 2.0** | 87.59 | 81.25 |
| **Augmented** | 80.36 | 73.13 |
| **Random** | 85.16 | 78.25 |

As the results in the table show, the F1 scores improved significantly on the random and augmented set in comparison to the results in section 4.2, and it interestingly improved slightly (87.59 vs 87.22) on the original SQuAD 2.0. We observe that the SQuAD 2.0 dataset contains a rare amount of typed spelling errors, and the fine-tuned model can pick up those spelling errors, hence increasing the performance on even the original set.

Also not surprisingly, the augmented fine-tuned model did not perform better on the augmented data compared to the original data, suggesting that adding noise and typographical errors into questions still hinders the model performance.

### 6.2 Evaluation

To test and analyze the root causes of the improvements, we had to take a deeper look at more data augmentations. We created five more datasets with different augmentation parameters. The following datasets were created:

- SQuAD 2.0 with only character deletion, and with augmentation probability of 30%
- SQuAD 2.0 with only character swap, and with augmentation probability of 30%
- SQuAD 2.0 with character swap and deletion but an augmentation probability of 50%
- SQuAD 2.0 with character swap and deletion but an augmentation probability of 70%
- SQuAD 2.0 with character swap and deletion but an augmentation probability of 90%

The same procedure of creating a prediction set using both models was performed.

### 6.3 Discussion

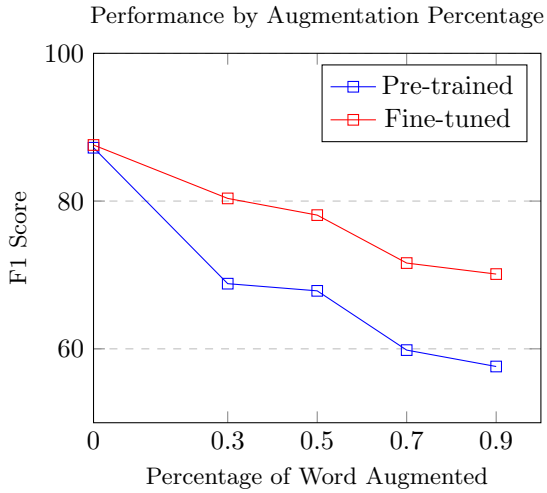
First, let us take a look at the change in augmentation transformations (deletion and swap).

| | **Fine-** | **Tuned** | **Pre-** | **Trained** |
|---|---|---|---|---|
| | **F1** | **Exact** | **F1** | **Exact** |
| **Swap** | 82.90 | 75.75 | 77.13 | 68.34 |
| **Deletion** | 84.48 | 77.50 | 77.98 | 68.88 |

The Fine-tuned model has higher F1 scores for both only swap and only deletion compared to a mixed augmentation. A mixed transformation of character swap and deletion leads to a higher confusion of the model because some words can lose a character and swap a character and completely lose their shape. The deletion augmentation gained a higher F1 score. Based on the data, character deletion usually keeps the same structure of the word while swaps can introduce a new word which would not exist in the context. For example, if you take the word *Normandy* and swap a character, you can end up with *Norcandy*. The model is not able to detect such a word in the context, but if the same character was deleted, *Norandy* is still a subset of the same word.

As mentioned, the fine-tuned model's performance is higher on the only deletion and swapping compared to mixed augmentation. This performance improvement is displayed drastically in the pre-trained model. The F1 scores jump from 68.82 on the augmented set with both swap and deletion to 77.13 and 77.98 on the swap only and deletion only sets respectively. But both deletion and swap have similar F1 scores because these transformations are equally unseen to the pre-trained model.

Now, let us take a look at a graph for the models' performance on different augmentation probabilities.




Looking at the graph, the difference between the fine-tuned and the pre-trained model is visible once the data is augmented.

At 0% augmentation (original SQuAD data), the fine-tuned model performed slightly better by a 0.37 F1. We believe that the cause of this performance improvement lies within the answers which the fine-tuned model provides. Taking a look at the question and answers, we see that Deepset's pre-trainend model provides longer and more detailed answers. While the answers are not necessarily wrong from a human's point of view, but they are not the gold answer. The fine-tuned model answers the questions more concisely. This corresponds better to the gold answers provided by the SQuAD 2.0 dataset. Below is an example of questions and answers provided by both models.

| |
|---|
| **Question**: Who did BSkyB compete with initially? |
| **Original Answer**: ONdigital |
| **Pre-Trained Model's Answer** |
| ONdigital (later ITV Digital) terrestrial offering and cable services |
| **Fine-Tuned Model's Answer** |
| ONdigital |

The fine-tuned model provided a concise response to the question which the pre-trained model could not. This occurred on many instances. For example, Sometimes the pre-trained model included a period which was not part of the gold answer.

After augmentation and adding typographical errors, the pre-trained model's performance dropped lower than the fine-tuned model as expected. The difference in the F1 scores between the pre-trained and fine-tuned models were 11.54, 10.25, 11.78, and 12.51 for the 30%, 50%, 70%, and 90% augmentations respectively. These differences remained consistent. Looking at each individual graph, there is a steep curve from 0% to 30% augmentation, and then from 50% to 70%. It is obvious that from 0% to 30% the data starts introducing errors and thus causing a sharper decrease in F1 score. After introducing more than 50% typographical errors, the question becomes non-legible and hard to grasp. Take a look at the same question with different augmentations:

| | |
|---|---|
| 0% | When did Herve go up against the Turks? |
| 30% | Wen did Herve to up against the Turks? |
| 50% | Wen did Nerve bo up against the Trks? |
| 70% | Whn did Hrve go up aginst the Trks? |
| 90% | Whe did erve gi up against the Turkw? |

After 50% of augmentation, even humans will pause and think about what this sentence could mean. This would explain the drastic change when the model is faced with such questions.

Taking a step back, let us also look at the behavior of the pre-trained model on the 30% augmented data which the model was fine-tuned on. Below is an example of a question and answer from both models to a question with spelling errors:

| |
|---|
| **Augmented Question:** |
| Wen did Huguenots colonize in orth America? |
| **Original Question:** |
| When did Huguenots colonize in North America? |
| **Pre-Trained Model's Answer** |
| New Netherland |
| **Fine-Tuned Model's Answer** |
| 1624 |

The pre-trained model is not able to recognize the word *Wen*, which is a major part of the question, while the fine-tuned model recognizes this word given the context of the question. It has learned to bypass different spelling errors.

Upon our further analysis, it seems that the model which is trained on spelling errors looks more at the context of the questions and tries to verify the missing or wrong characters from the question. This dependence on context causes the model to look for more concise answers excluding unnecessary information. Not only this improves the answers of augmented questions containing spelling errors, it also improves upon the original SQuAD 2.0 answers.

## 7 Conclusion

Motivated by the real world Question Answering situation where human errors are common in constructing the question, our paper analyzes the effect of adding typographical errors into the questions of QA model inputs. The result is not surprising — when introducing errors to model inputs, the model's F1 score dropped by 21%. Overall, once the technique of fine-tuning the model with typographical errors was applied, we saw improvement on the original data, challenge data, and random data. Our analysis found out that the model fine-tuned with augmented dataset outputs more concise and accurate answers, and it is able to recognize the key answer triggering misspelled words such as when, how, why. These words identify the question or include hints from the context. As a result, the fine-tuned model can mitigate the impact of typographical errors in Question Answering.

## 8 Acknowledgements

We thank Dr. Durrett and the TAs of the NLP class at The University of Texas at Austin for providing us with the opportunity of writing this paper.

## 9 References

[Jia and Liang, 2017] Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328.

[Liu et al., 2019] Liu, N. F., Schwartz, R., and Smith, N. A. (2019). Inoculation by fine-tuning: A method for analyzing challenge datasets. *CoRR*, abs/1904.02668.

[Morris et al., 2020] Morris, J. X., Lifland, E., Yoo, J. Y., and Qi, Y. (2020). Textattack: A framework for adversarial attacks in natural language processing. *CoRR*, abs/2005.05909.

[Rajpurkar et al., 2018] Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822.

[Wallace et al., 2019] Wallace, E., Feng, S., Kandpal, N., Gardner, M., and Singh, S. (2019). Universal adversarial triggers for NLP. *CoRR*, abs/1908.07125.

[Zhou and Bansal, 2020] Zhou, X. and Bansal, M. (2020). Towards robustifying NLI models against lexical dataset biases. *CoRR*, abs/2005.04732.