Robust QA System with xEDA: Final Report

Stanford CS224N Default RobustQA Project

Kazuki Mogi
Department of Computer Science
Stanford University
kmogi@stanford.edu

Alexander Donovan
Department of Computer Science
Stanford University
adonovan@stanford.edu

Abstract

Data augmentation is often used in machine learning to reduce overfitting. However, data augmentation has not been thoroughly explored in NLP in the context of improving robustness on shifts in data domains. We present **xEDA**: extended easy data augmentation techniques for boosting performance on question answering tasks. xEDA extends existing data augmentation techniques by drawing inspirations from techniques in computer vision. We evaluate its performance on out-of-domain question answering tasks and show that xEDA can improve performance and robustness to domain shifts when a small subset of the out-of-domain data is available at train time.

1 Introduction

Question answering is a challenging natural language processing (NLP) task with much significance to real-world applications such as improving search queries and building virtual assistants. However, one area of concern when training question answering systems is that they often learn brittle correlations and overfit to their training data (in-domain data). Consequently, question answering systems can perform poorly on data from different distributions besides the distribution from which the training data is taken (out-of-domain data). This can pose problems in real-world applications where the distributions of the training data and the test data may be different. Therefore, it is integral that we can train question answering systems that can generalize well to out-of-domain data even when only a few out-of-domain training samples are available.

As noted in previous literature, data augmentation can reduce overfitting and help train machine learning models that perform well on unseen data. In computer vision, multiple data augmentation techniques have been proposed and shown to improve the performance of machine learning models in tasks including but not limited to: image classification, weakly-supervised localization, image captioning, and out-of-domain data detection [1] [2] [3]. However, data augmentation has not been thoroughly explored in NLP [4]. Hence, in this paper, we propose **xEDA**: extended easy data augmentation techniques for NLP. xEDA extends easy data augmentation techniques (EDA) [4] by drawing inspirations from proven data augmentation techniques in computer vision. In particular, xEDA alters techniques from EDA such that it reduces corruption in grammar and meaning when augmenting the data. The motivation behind this strategy is that complex NLP tasks such as question answering may depend heavily on grammar and meaning, and corrupted data may incur a drop in performance.

We apply xEDA to train a question answering system that performs well on out-of-domain data even when only a limited amount of out-of-domain data is available at train time. We further evaluate the effectiveness of xEDA by analyzing our results from this task. Our experiments show that xEDA can boost the out-of-domain performance of question answering systems: our best data augmentation techniques improve scores used as the evaluation metric (EM and F1) by more than 6%. Our analysis also suggests that an augmented data of smaller size may lead to better performance than non-augmented data of larger size in some cases.

2 Related Work

2.1 Data Augmentation in NLP

Despite the frequent use of data augmentation in certain areas such as computer vision, universal data augmentation techniques in natural language processing (NLP) have not been thoroughly explored. This is partly due to the difficulty in devising generalized rules for language transformation [4]. Data augmentation techniques developed in NLP include back translation [5], data noising as smoothing [6], and synonym replacement [7]. Although these data augmentation techniques are valid, they are not often used in practice due to the high cost of implementation relative to performance gain [4].

One existing framework for universal data augmentation in NLP is EDA as mentioned earlier [4]. The authors demonstrate the utility of these data augmentation techniques by reporting performance gains on text classification tasks. The four data augmentation techniques in EDA are

- **synonym replacement** (SR): randomly replace n non-stop words in the sentence with corresponding synonyms
- **random insertion** (RI): insert a synonym of a random non-stop word at a random position in the sentence, repeat this n times
- **random swap** (RS): swap the positions of randomly selected words in the sentence, repeat this n times
- **random deletion** (RD): randomly remove each word in the sentence with probability p

Please refer to Table 6 in the appendix for examples. The authors point out that EDA may not yield substantial improvements when using pre-trained models [4]. Although they do not articulate the reasons why, our conjecture is that EDA can corrupt the grammar and/or meaning of the data, which may be harmful to models that capture language representation well including many of the common pre-trained models. Therefore, in tasks that require the model to capture grammar and meaning such as machine translation or question answering, EDA may introduce excessive noise in the input data during training.

2.2 Data Augmentation in Computer Vision

Data augmentation techniques from computer vision include CutOut [1], Mixup [2], and CutMix [3]. These are three data augmentation techniques typically applied to image input during training. CutOut [1] blacks out a contiguous part of an image. Mixup [2] overlays two images during training and adjusts the ground truth label of the image. CutMix [3] combines CutOut and Mixup by replacing a continuous part of an image with another image during training and adjusting the ground truth label of the image. These techniques are known to provide performance gains on image classification, weakly-supervised localization, image captioning, and out-of-domain data detection, and help train a robust model that attends to greater parts of the input [1] [3] [8]. We use high-level ideas from these techniques to design xEDA.

3 Approach

We use pre-trained DistilBERT [9], a distilled version of the BERT transformer model [10], to build a question answering system. We finetune this pre-trained DistilBERT using the in-domain datasets and further finetune it using the out-of-domain datasets. Our goal is to augment the in-domain and/or out-of-domain train datasets to boost performance on the out-of-domain test datasets.

3.1 xEDA

As discussed in Section 2.1, one possible shortcoming of EDA is its tendency to corrupt the grammar and/or the meaning of the data, potentially resulting in a performance drop for complex NLP tasks. Therefore, we propose xEDA: an extension of EDA consisting of techniques that are simple and easy to implement while maintaining the grammatical structure and/or meaning of the data.

We have implemented and evaluated the following techniques: **masking**, extended random deletion (**xRD**), extended random insertion (**xRI**), and simple extended random insertion (**simple xRI**).

masking
Definition: *Randomly choose a word in a sentence from a uniform distribution and mask all following words in the sentence using a special masking token with probability α. Repeat this for every sentence in the context paragraph except for the sentence containing the answer.*
Masking is an extension of random deletion in EDA: instead of removing words at random, we mask words at random. This retains the grammatical structure and prevents changes in the meaning (although it can reduce the amount of information). The idea to mask words, in particular mask contiguous words, is inspired by Cutout [1].

extended random deletion (xRD)
Definition: *Randomly choose a sentence from the dataset that does not contain the answer. Delete this sentence with probability α. Repeat this for every context paragraph.*
Extended random deletion (xRD) is an extension of random deletion in EDA: instead of deleting a random word, we delete a random sentence in the context paragraph. This retains the grammatical structure and minimally changes the meaning of the context paragraph. Similar to masking, the idea of xRD is inspired by Cutout [1].

extended random insertion (xRI)
Definition: *Insert a random sentence from the dataset before/after a randomly chosen sentence in the context paragraph with probability α. Repeat this n times for every context paragraph.*
Extended random insertion (xRI) is an extension of random insertion in EDA: instead of inserting a synonym of a random word into a random position, we insert a random sentence from the data between sentences at random. This prevents changes in the meaning (although there are rare occasions when the meaning can change) and minimally corrupts the grammar. The idea of integrating two samples is inspired by Mixup [2]. (During our experiments, we fixed $n = 3$.)

simple extended random insertion (simple xRI)
Definition: *Randomly choose a context paragraph from the dataset. Insert this paragraph to the beginning or end of the context paragraph with probability α. Repeat this for every context paragraph.*
Simple extended random insertion (simple xRI) is an extension of random insertion in EDA: it is similar to xRI but instead of inserting a random sentence, this technique inserts an entire context paragraph. Simple xRI is even less prone to changes in meaning and sentence structure because the original context paragraph is intact. This technique is similarly inspired by Mixup [2].

For examples of these techniques, please refer to Table 1.

Operation	Example
none	I played soccer on a hot day. I scored two goals.
masking	I played soccer on [MASK] [MASK] [MASK]. I scored two goals.
xRD	I scored two goals.
xRI	I played soccer on a hot day. I like pizza. I scored two goals.

Table 1: Examples of xEDA

4 Experiments

4.1 Data

We have three in-domain train and dev datasets (SQuAD 1.0 [11], NewsQA [12], and Natural Questions [13]) and three out-of-domain train and dev datasets (DuoRC [14], RACE[15], and RelationExtraction[16]). We use the in-domain train datasets to finetune the pre-trained DistilBERT. We use the out-of-domain train datasets to further finetune DistilBERT. The corresponding dev datasets are used to measure performance and save the best model during each finetuning process. We also have limited access to three out-of-domain test datasets. We use this to evaluate the performance of our best models. For details about the datasets, please refer to Table 7 and Figure 2 in the appendix.

4.2 Evaluation method

One numerical evaluation metric is the **F1** score, which is the harmonic mean of precision and recall. Another evaluation metric is the **Exact Match** (EM) score which counts a prediction as correct only if there is an exact match with the correct answer. We use the average of these two metrics to measure performance. The goal is to train a model that obtains higher F1 and EM scores. In a larger context, we are also interested in comparing the performance of different data augmentation techniques under different settings to evaluate their usefulness. Please refer to section 4.3 more details.

4.3 Experiment Details

Experiment 1 – Augmenting In-Domain Train Datasets
In this experiment, we applied xEDA to in-domain train datasets and measured its performance on out-of-domain dev datasets. The intention behind this experiment is to evaluate the effectiveness of applying xEDA to in-domain train datasets. It is possible that xEDA removes certain features from the datasets, and prevent our question answering system from learning superficial correlations and characteristics specific to the in-domain train datasets. In such case, we may see an improvement in the performance on out-of-domain datasets.
This experiment involved 3 models. The first model used the in-domain test datasets with no data augmentation. The second model augmented the context attribute of the in-domain train datasets with masking ($\alpha = 0.5$). The third model augmented the context attribute of the in-domain train datasets with xRI ($\alpha = 0.5$). We finetuned the pre-trained DistilBERT on these datasets and evaluated performance on the out-of-domain dev datasets. The hyperparameters are summarized in Table 8.

Experiment 2 – Augmenting Out-of-Domain Train Datasets
In this experiment, we applied xEDA to out-of-domain train datasets and measured its performance on out-of-domain dev datasets (and out-of-domain test datasets for some). Since data augmentation often prevents overfitting, xEDA may be particularly useful in preventing our question answering system from fitting excessively to the out-of-domain train datasets which are small in size. Hence, it is likely that we will observe performance gains.
This experiment involved multiple models, all finetuning the control DistilBERT model from Experiment 1 on the out-of-domain train datasets. The first model was the control that simply finetuned on the out-of-domain datasets without augmentation. The remaining models used out-of-domain training datasets where the context attribute was augmented via xEDA. Since certain techniques in xEDA can complement each other, we also applied multiple data augmentation techniques simultaneously. Additionally, we utilized emsembling via majority voting and combined all augmented models. As each augmentation may make our question answering system more robust in a different way, this may increase the robustness. The augmentation techniques are summarized in Table 3. The hyperparameters are summarized in Table 9.

Experiment 3 – Even Smaller Out-of-Domain Datasets
This experiment differs from the aforementioned two experiments because it does not aim to build a model that achieves better F1 and EM scores. Instead, this experiment attempts to uncover the relationship between the effectiveness of xEDA and the size of the out-of-domain train datasets. In addition to this, this experiment allows us to gain an insight into the question "How much data is sufficient to achieve the baseline performance from Experiment 2 if we use xEDA?".
For each data augmentation technique in xEDA, namely masking, xRD, and xRI, we selected the best performing setting from Experiment 2: masking ($\alpha = 1.0$), xRD ($\alpha = 0.5$), and simple xRI ($\alpha = 0.5$). For each of these three data augmentation techniques, we re-ran Experiment 2 using 20%, 40%, 60%, and 80% of the data. The hyperparameters are summarized in Table 10.

Note: On each epoch, we used a different randomized augmented dataset to gain better results.

4.4 Results

Experiment 1
Table 2 summarizes the results of Experiment 1. Contrary to our expectation, we observe that both data augmentation techniques performed worse than our baseline model. Masking decreased EM and F1 by 1.57 and 0.27 points respectively. xRI decreased EM and F1 by 1.05 and 1.30 points respectively. One possible explanation mentioned in previous research with EDA is that there are only marginal performance gains from data augmentation when data is sufficient, which may be the case

with large amounts of in-domain train datasets [4]. It is also possible that xEDA did not introduce enough noise in the datasets to prevent our question answering system from learning superficial correlations that are present in the in-domain datasets but not in the out-of-domain datasets.

Model	Dev EM	Dev F1
baseline	**33.25**	**48.43**
masking	31.68	48.16
xRI	32.20	47.03

Table 2: Results of Augmenting In-Domain Datasets

Experiment 2
Table 3 summarizes the results of Experiment 2. In this experiment, the results are close to what we expected: in general data augmentation boosts overall performance. This suggests that xEDA can be useful in boosting the performance of question answering systems on out-of-domain datasets if few out-of-domain samples are available at train time. However, there does not seem to be a data augmentation technique that stands out in performance. Moreover, we observe that the hyperparameters of the data augmentation has a decent amount of impact on performance. For instance, masking with $\alpha = 0.5$ performs worse than the baseline model whereas masking with $\alpha = 1.0$ performs better than the baseline model by a non-trivial amount. Similarly, xRD with $\alpha = 0.5$ performs better than the baseline model whereas xRD with $\alpha = 1.0$ performs worse than the baseline model. This implies that when applying xEAD, we should try different hyperparameters to obtain the greatest performance gain possible.

Model	Dev EM	Dev F1	Test EM	Test F1
baseline	34.29	49.03	–	–
masking ($\alpha = 0.5$)	34.03	48.89	–	–
masking ($\alpha = 1.0$)	**35.86**	49.59	–	–
xRD ($\alpha = 0.5$)	34.82	49.35	–	–
xRD ($\alpha = 1.0$)	34.29	48.72	–	–
xRI ($\alpha = 0.5$)	34.03	49.78	–	–
xRI ($\alpha = 1.0$)	34.03	49.60	–	–
simple xRI ($\alpha = 0.5$)	34.55	49.80	40.826	58.029
simple xRI ($\alpha = 1.0$)	33.51	49.02	–	–
masking ($\alpha = 0.5$) + xRI ($\alpha = 1.0$)	33.51	49.04	–	–
masking ($\alpha = 1.0$) + xRI ($\alpha = 1.0$)	34.55	49.78	–	–
masking ($\alpha = 0.5$) + simple xRI ($\alpha = 0.5$)	34.03	49.95	–	–
masking ($\alpha = 1.0$) + simple xRI ($\alpha = 0.5$)	34.03	49.38	–	–
Ensembling all 12 augmented models	34.55	**50.41**	-	-

Table 3: Results of Augmenting Out-of-Domain Datasets

Experiment 3
Figure 1 (and Table 11 in the appendix) summarizes the results of Experiment 3. We see that, in general, larger out-of-domain datasets lead to greater performance gains. This is expected because larger datasets help machine learning models perform better in general. We also see that for all three data augmentation techniques, using 40% or more of the out-of-domain train datasets achieves the same performance gain as using 100% of the out-of-domain train datasets without augmentation. This is another indication that xEDA can help boost the performance of question answering systems on out-of-domain datasets if few out-of-domain samples are available at train time.

5 Analysis

5.1 Validity of xEDA

We examine the augmented datasets and check if masking, xRD, and xRI have preserved the meaning and grammar of the texts as intended such that the questions are still answerable. (Note that some lengthy context paragraphs are trimmed.)

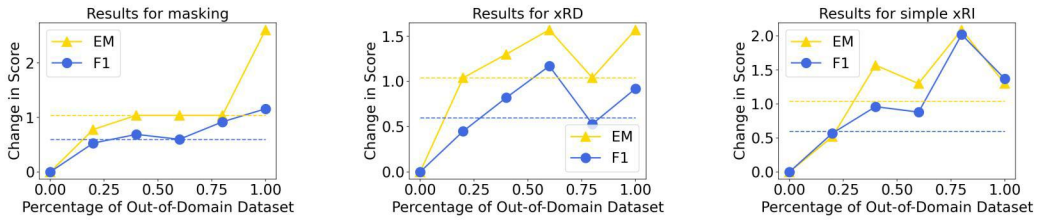

Figure 1: Results of Experiment 3. The change in score is relative to the score when 0% of the out-of-domain train datasets are used for finetuning. The dotted lines indicate the change in score when using 100% of out-of-domain train datasets for finetuning without data augmentation.

masking
Here is an example of a sample from the out-of-domain training dataset where masking is applied to the context paragraph.

> **Context (masking):** Kitty meets Charles [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK]. When Walter discovers his wife's infidelity, he seeks to punish her by threatening to divorce her on the grounds of adultery, if she doesn't accompany him to a small village in a remote area of China.
> **Question:** "Where does Walter force Kitty to go with him?"
> **Answer Text:** "to a small village in a remote area of China."

The grammar and meaning are almost intact as intended. We can also observe one potential merit of masking. One difficulty in successfully answering this question is that the sentence that contains the word "Kitty" does not contain relevant information to the question. Masking can potentially train a model that can tackle this type of situation well: since the sentence containing the word "Kitty" is heavily masked, the model is encouraged to "pay attention" to other sentences.

xRD and xRI
Here are examples of a sample from the out-of-domain training dataset where xRD and xRI are applied to the context paragraph respectively.

> **Context (xRD):** [Deleted sentence] We may wear collars and ties instead of war-paint, but our instincts remain basically unchanged. The whole of the recorded history of the human race, that tedious documentation of violence, has taught us absolutely nothing.
> **Context (xRI):** The pair fall asleep, but are awakened when Holly has a nightmare about Fred. What is really frightening, what really fills you with despair, is the realization that when it comes to the crunch, we have made no actual progress at all. We may wear collars and ties instead of war-paint, but our instincts remain basically unchanged. The whole of the recorded history of the human race, that tedious documentation of violence, has taught us absolutely nothing.
> **Question:** "Recorded history has taught us"
> **Answer Text:** "nothing"

The grammar and meaning are almost intact as intended. We can also gain an insight into the ways in which applying xRD or xRI helps train a more robust model. Notice that while these augmentation techniques do not change the grammar or meaning, they change the relative position of the answer. These variations can help the question answering system from overfitting to the original data especially when iterating over the same datasets multiple times (provided that each iteration produces different augmentations like our implementation). Another possible advantage is the same as masking: since there are variations, a model that focuses on a small set of features or places may perform poorly, which encourages the question answering system to attend to greater parts of the input. This may be helpful in reducing the model from picking up superficial correlations or features that are specific to the in-domain datasets.

5.2 Mistakes by Our Model

Data augmentation improved the robustness of the trained models. However, the models continue to err in certain scenarios.

Masking and Dispersed Attention
Here is an example of an error the model trained with masking makes.

Question :"Who returns to his home first?"
Correct Answer :"Ryu"
Predicted Answer by Model :"Dong-jin had previously set up an electric booby trap on his doorknob, which renders Ryu unconscious. Dong-jin"

The context paragraph describes that Ryu returns home first, but the model focuses on Dong-jin harming Ryu. The model's answer does not seem to be related to returning home, and the word "home" appears many times in the paragraph, which may have led to the incorrect prediction due to the model not understanding where to focus attention.

Occasionally, there are examples where the baseline model answers a question correctly, but the augmented model answers incorrectly. One such example deals with a passage about Brazil in the Race dataset.

Question : "Who might block the development of Brazil?"
Correct Answer : "Getulo Vargas"
Predicted Answer by Augmented Model : "Princess Isabel"
Predicted Answer by Baseline Model : "Getulo Vargas"

One reason this may have occurred is that masking expanded the attention of our question answering system. Gutelio Vargas is identified as the cause of several failures in Brazil at the end of the context, whereas Princess Isabel is associated with the abolishing of slavery in Brazil in the middle of the context. Another possible reason is the ambiguity of the context. It explains "The Brazilian Emperor, Pedro I, abolished slavery in 1888 in the face of Princess Isabel" where the phrase "in the face of" could mean "despite". With this interpretation, the model's answer may be correct.

Penalized Synonym
The models often return synonyms to the correct answer which decrease the EM score even though they are correct answers. One such example deals with a passage about Salt Flat, Utah, a western town.

Question : "Where does Stubby go hoping to get a job in the local casino?"
Correct Answer : "Salt Flat Utah"
Predicted Answer by Model : "small Wild West town"

This answer is not as exact as Salt Flat, Utah, but the model also refers to this place as a small Wild West town. Hence, it is a correct answer.

5.3 Analysis by Context Length

For each of the 12 augmentation models as well as an ensemble model that predicts the most frequent response of the other 12 models, the performance on the dev set for different word counts in the context pertaining to the question was assessed. The ranges were chosen so that each range shares roughly 20% of the total context examples in the dev set. As can be seen from Table 4, certain models are particularly strong in certain word count ranges. This perhaps suggests that different augmentation techniques help train models that are robust in different ways. For instance, we see that xRD ($\alpha = 0.5$) is especially strong in the 0 - 23 word range. This may be because that xRD ($\alpha = 0.5$) produces many training samples of short word range and the model learns to perform well on these samples. On the contrary, we observe that xRI models in general perform relatively well in the > 655 word range. This may be because of the opposite reason: xRI produces many training

Model	≤ 23		$24-224$		$225-311$		$312-655$		>655	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
masking ($\alpha = 0.5$)	62.03	79.76	32.00	50.98	25.97	39.76	34.21	45.14	14.67	27.48
masking ($\alpha = 1.0$)	64.56	81.62	30.67	49.91	27.27	39.27	38.16	49.13	**17.33**	26.15
xRD ($\alpha = 0.5$)	**68.35**	82.81	33.33	52.03	27.27	42.26	31.58	42.25	12.00	25.91
xRD ($\alpha = 1.0$)	56.96	77.18	29.33	47.70	27.27	40.96	**40.79**	**50.41**	16.00	26.00
xRI ($\alpha = 0.5$)	64.56	82.16	**34.67**	53.60	23.38	29.91	31.58	43.24	14.67	28.71
xRI ($\alpha = 1.0$)	63.29	81.24	33.33	52.55	25.97	40.05	32.89	44.90	13.33	27.88
simple xRI ($\alpha = 0.5$)	63.29	80.79	32.00	51.61	27.27	41.40	35.53	45.75	13.33	28.10
simple xRI ($\alpha = 1.0$)	67.09	**83.33**	30.67	49.04	25.97	40.93	31.58	44.89	10.67	25.37
masking (0.5) + xRI (1.0)	65.82	83.24	**34.67**	**53.68**	25.97	38.73	31.58	43.78	13.33	28.08
masking (1.0) + xRI (1.0)	65.82	82.96	30.67	51.77	24.68	39.73	32.89	44.19	12.0	25.04
masking (0.5) + simple xRI (0.5)	63.29	80.00	32.00	51.75	25.97	40.80	32.89	45.63	14.67	**30.26**
masking (1.0) + simple xRI (0.5)	65.82	83.24	**34.67**	**53.68**	25.97	38.73	31.58	43.78	13.33	28.08
Ensembling all 12 models	63.29	81.30	32.00	52.39	**28.57**	**42.54**	32.89	45.21	14.67	29.26

Table 4: Performance of Each Model for Different Word Counts of Context. The ranges were chosen such that the number of samples in each range is roughly equivalent.

samples of longer word range and the model leans to perform well on these samples. xRD also performs very well in the $312-655$ word range. Ensembling the 12 models performed the best in the $225-311$ word count range, which suggests that combining the models in this range is effective. More experimentation is required to understand the reasons for these relationships.

Given that each model has its own strength, we hypothesized that adopting an adaptive prediction in which the model making the prediction is chosen by the length of the context may improve performance. Under this hypothesis, we implemented the following prediction procedure:

- For each question, get the length (in terms of word) of the corresponding context.
- Identify the range that this context belongs to in Table 4.
- Find the best model for that range and take the prediction from this best model.

The result of this procedure is summarized in Table 5. We see that this procedure, adaptive prediction, improves both F1 and EM by a significant amount.

Model	Dev EM	Dev F1	Test EM	Test F1
baseline	33.25	48.43	–	–
control from Exp. 1	34.29	49.03	–	–
adaptive prediction	**38.22**	**51.40**	**41.835**	**59.012**

Table 5: Results of Augmenting In-Domain Datasets

6 Conclusion

We discovered that xEDA can help build a question answering system that is robust to shifts in domain distributions if few samples of out-of-domain datasets are available at train time. In particular, by applying xEDA to out-of-domain datasets during training, we increased the performance of our question answering system by 6.1% in terms of F1 and by 14.9% in terms of EM when compared to the provided baseline on the dev set. Moreover, using 40% of the out-of-domain train datasets augmented via xEDA achieved the same performance as using 100% of the out-of-domain train datasets.

As xEDA is simple and easy to implement, it has the potential to be applied to many other complex NLP tasks. However, in this paper we were only able to evaluate the effectiveness of xEDA on question answering task with domain shifts. Therefore, it may be illuminating for practitioners and researchers in NLP to either apply xEDA to other NLP tasks or explore data augmentation techniques catered for NLP. Moreover, there are other variants of EDA that we were not able to include in this paper, such as extended random swapping (swap entire sentences at random), generative random swapping (swap a sentence with a sentence generated by a language model at random), generative random insertion (insert a sentence generated by a language model at random). Implementing and evaluating these may be an extension of this paper.

References

[1] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017.

[2] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

[3] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019.

[4] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019.

[5] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension, 2018.

[6] Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. Data noising as smoothing in neural network language models, 2017.

[7] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[8] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization, 2017.

[9] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[11] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[12] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830, 2016.

[13] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019.

[14] Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. *CoRR*, abs/1804.07927, 2018.

[15] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. *CoRR*, abs/1704.04683, 2017.

[16] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *CoRR*, abs/1706.04115, 2017.

A Appendix (optional)

Augmentation	Definition	Example
None	n/a	I played soccer on a hot day.
SR	replace random words with synonyms	I played soccer on a **burning** day.
RI	insert synonyms into random positions	I played **warm** soccer on a **hot** day.
RS	swap positions of random words	I played **hot** on a **soccer** day.
RD	delete random words	I played soccer on a day.

Table 6: Brief Summary of EDA

Dataset	Question Source	Passage Source	Train	Dev	Test
`in-domain datasets`					
SQuAD [11]	Crowdsourced	Wikipedia	50,000	10,507	–
NewsQA [12]	Crowdsourced	News articles	50,000	4,212	–
Natural Questions [13]	Search logs	Wikipedia	50,000	12,836	–
`out-of-domain datasets`					
DuoRC [14]	Crowdsourced	Movie reviews	127	126	1,248
RACE [15]	Teachers	Examinations	127	128	419
RelationExtraction [16]	Synthetic	Wikipedia	127	128	2,693

Table 7: Details about Datasets

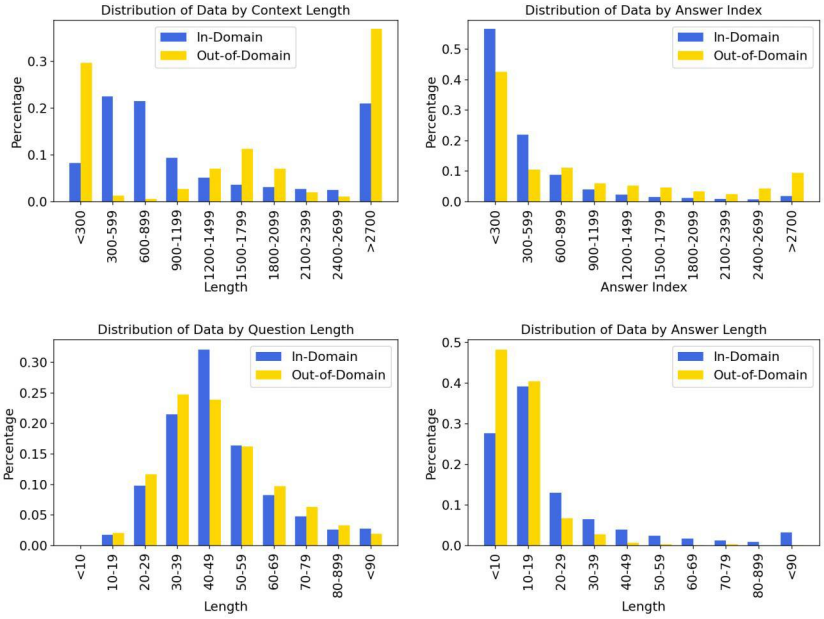

Figure 2: Comparison of Datasets by Different Metrics

Setting	Value
base model	pre-trained DistilBERT
batch size	16
epoch size	3
learning rate	3e-5

Table 8: Settings for Experiment 1

Setting	Value
base model	finetuned DistilBERT from Exp. 1
batch size	16
epoch size	15
learning rate	1e-5

Table 9: Settings for Experiment 2

Setting	Value
base model	finetuned DistilBERT from Exp. 1
batch size	16
epoch size	$\lfloor \frac{15}{\text{fraction of data used}} \rfloor$
learning rate	1e-5

Table 10: Settings for Experiment 3

Model		Dev EM	Dev F1
baseline	0%	33.25	48.43
	100%	34.29	49.03
masking ($\alpha = 1.0$)	20%	34.03	48.96
	40%	34.29	49.12
	60%	34.29	49.03
	80%	34.29	49.35
	100%	35.86	49.59
xRD ($\alpha = 0.5$)	20%	34.29	48.88
	40%	34.55	49.25
	60%	34.82	49.60
	80%	34.29	48.96
	100%	34.82	49.35
simple xRI ($\alpha = 0.5$)	20%	33.77	49.00
	40%	34.82	49.39
	60%	34.55	49.31
	80%	35.34	50.45
	100%	34.55	49.80

Table 11: Results of Experiment 3