Robust Question-Answering with Data Augmentation and Self-Supervised Learning

Stanford CS224N Default Project
Track: RobustQA
TA mentor: **Allan Zhou**; No external collaborators nor external mentors; Not sharing project.

Arsh Zahed, Daniel Zeng, Arvind Sridhar
Department of Computer Science, Stanford University
{azahed, dazeng, arvind98}@stanford.edu

Abstract

Machine reading comprehension has made significant strides in recent years. However, Question-Answering (QA) systems have struggled to generalize beyond training data. In Robust QA, we tackle situations where the train and test distributions are different, with significantly fewer labeled out-of-domain (OOD) examples. We implement several data augmentation approaches to improve OOD robustness: data mixing, selective masking, Easy Data Augmentation (EDA), and Back-Translation (BT). We also propose and investigate a self-supervised (SS) learning approach, inspired by similar techniques in the vision domain. Combining our three best techniques (EDA, BT, SS) yields the best OOD validation performance. To maximally improve test set performance, we use the best technique for each OOD dataset to generate predictions; this strategy enables us to achieve third on the RobustQA test leaderboard by F1, with an F1 score of 62.99 & EM of 45.05.

1 Introduction

As natural language models grow to unprecedented scale and approach human-level performance on various tasks, research has focused on the ability of these neural systems to generalize beyond their training distributions and produce reliable results in various out-of-distribution settings. In particular, neural Question-Answering (QA) systems have become adept at reading comprehension, with popular datasets like SQuAD [1] enabling models to effectively identify spans of text in context paragraphs that answer posed questions. While state-of-the-art QA systems surpass even human-level performance on SQuAD [2], they often fall apart when applied to novel reading comprehension/QA tasks with out-of-domain (OOD) test distributions, such as the RACE [3], DuoRC [4], and Relation Extraction [5] QA datasets. The gap between the in-domain (ID) and OOD performance of neural QA systems showcases their *lack of robustness* and represents a major hurdle towards their deployment in the real world, where input distributions vary and unreliable results can have severe negative impact.

A straightforward approach to mitigate this lack of robustness is to simply train the system on more data, i.e. pool QA datasets like SQuAD, NewsQA [6], Natural Questions [7], RACE, DuoRC, etc. into a single large training dataset such that the model learns the nuances of each domain. However, this technique is not scalable and fails when the target domain has only a few datapoints (not enough to properly learn the distribution). In particular, in the ideal case, we desire models that are robust to *unknown* distributional shifts in the data, still able to produce reliable results in these settings (making them real-world ready). In this study, we investigate the ability of neural QA systems trained on SQuAD, NewsQA, and Natural Questions to adapt to OOD test distributions (RACE, DuoRC, Relation Extraction) given only limited labeled training data, but copious amounts of additional unlabeled data (i.e. context paragraphs and questions without answers), from the target domain.

To succeed in this low-resourced setting, we develop several novel data augmentation and self-supervised learning techniques to effectively learn the distribution from the limited available data.

With data augmentation, our goal is to generate additional training samples in the OOD distribution by leveraging the existing ID and OOD data. First, we propose **Data Mixing**: splicing the context paragraphs of ID (plentiful) and OOD (scarce) datapoints together to generate additional OOD augmented training samples. Next, we propose **Selective Masking**: selecting tokens at random in the non-answer sentences of OOD context paragraphs to drop and fill in using a pre-trained language model, generating augmented examples with slightly different phrasing. While these two techniques result in additional OOD data to train on, we find that they fail to convincingly outperform both our ID baseline (QA model trained on ID data only) and ID + OOD baseline (trained on ID + available OOD training data) on the OOD validation set, measured using the F1 and Exact Match (EM) scores. We hypothesize that these augmentations are not strong enough, i.e. the augmented examples are still too similar to existing OOD training points and thus do not provide much additional learning value.

To introduce stronger augmentations, we adapt **Easy Data Augmentation (EDA)** [8], a set of 4 data augmentation techniques for OOD training samples: random synonym replacement, random insertion, random swap, and random deletion. Further, we propose **Back Translation**: back-translate the question and non-answer sentences in OOD context paragraphs using a pivot language to again generate augmented OOD examples with similar phrasing. Finally, we propose **Self-Supervised Learning** (inspired by [9]), letting the QA model learn from the unlabeled OOD test data itself (which is plentiful) by constructing pseudo-QA tasks using the context paragraphs of these samples. With this approach, the model is able to learn the nuances of the OOD distribution before fine-tuning on the OOD training data. On its own, we find that self-supervised learning yields the best performance on the OOD validation set (F1 score of 51.87, EM score of 37.72). However, because these techniques are all modular and composable, we are able to combine all three and achieve even better performance: **F1 score of 54.29 and EM score of 40.11 on the OOD validation set**. On the OOD test set, our best-performing model combines all of our methods on each OOD test set separately, achieving an **F1 score of 62.99 and EM of 45.05, achieving 3rd place by F1 on the RobustQA test leaderboard.**

2 Related Work

Question-Answering The task of machine reading comprehension has been well-studied, with the pioneering SQuAD dataset [1] enabling models to approach human-level performance. Concretely, given a context paragraph and a question whose answer is contained within this passage, the model is tasked with predicting what the answer is. State-of-the-art QA systems today utilize massive transformers with beneficial scaling properties, i.e. BERT [10], in order to surpass even human-level performance on SQuAD [2]. However, these systems are often not robust, and fall apart when presented with out-of-domain inputs [11]. Because many domains only have limited training data available, and labeling data is laborious and costly, training QA systems to adapt to novel test distributions with only limited samples has garnered great interest in the NLP research community.

Data Augmentation Data augmentation is a commonly used technique to improve the size, scope, and diversity of training data. When augmenting natural language, a common approach is synonym replacement. Rule-based synonym replacement [12] has succeeded in generating natural examples on a token level. Easy Data Augmentation (EDA) [8] extends this by randomly swapping, inserting, and deleting words. Rule-based approaches excel at augmenting small datasets that need more examples of natural language, but have largely failed to account for sentence-level semantics. Recent work has instead used a BERT Masked Language Model to account for this [13]. This approach can utilize any information captured by the language model during training to account for sentence semantics, and the model can be finetuned to different domains. This method is much more flexible than rule-based synonyms and does not necessarily require human knowledge of the domains, as the BERT finetuning can be automated instead, but is limited by the performance of the BERT language model.

Back-Translation [14] is another approach to data augmentation. The training input is first translated to a pivot language, and is then translated back. Because the entire text can be translated together, back-translation can capture higher-level features in the sentences that synonym replacement cannot. Additionally, back-translation can use several different languages, allowing for multiple different augmentations for the same input. This approach does suffer a similar issue as the BERT language model method, as it is limited by the performance of the machine translation models used.

Self-Supervised Learning Self-supervised learning involves learning from unlabeled data. In the computer vision domain, Test-Time-Training [9] learns from unlabeled images by constructing dummy pseudo-tasks involving rotating an image and predicting the rotation angle. Broadly speaking, if the pseudo-task is well-defined, training the model on it helps it learn about the structure and distribution of data, which it can leverage to improve performance on the original task given a small labeled support. This approach can also be combined with data augmentation to improve robustness.

3 Setup and Approach

3.1 Defining Baselines

To analyze the robustness of our neural QA system, we first establish two baselines: an in-domain (ID) baseline and an in-domain + out-of-domain (ID+OOD) baseline. For the ID baseline, we take a pre-trained DistilBERT [15] language model from HuggingFace (provided by staff) and fine-tune the model on the SQuAD, NewsQA, and Natural Questions training datasets (considered ID datasets for this project). For the ID+OOD baseline, we additionally fine-tune the DistilBERT model on the RACE, DuoRC, and RelationExtraction training datasets (considered OOD datasets for this project), which are of limited size (only 127 labeled training examples per dataset, v.s. 50000 labeled examples per ID training dataset). We evaluate both baselines on the ID and OOD validation sets. For all of our experimentation, we use the staff-provided datasets and code for RobustQA (with logic for training and evaluation), and write our own helper functions to implement the following approaches.

3.2 Data Mixing

To improve the OOD performance of our model, we first experiment with various data augmentation techniques, with the goal of expanding the existing limited OOD training set by generating additional synthetic training samples within the OOD distribution (using the available OOD and ID training samples). For data mixing, we hope to leverage the large amount of ID data we have available in order to boost the size of our OOD training set and enable the model to more accurately represent that domain. Given an OOD training sample, we select an ID training sample at random, select some sentences from this ID sample's context, and insert them into our OOD sample's context paragraph, ensuring that the original answer is preserved. For example, if an ID sentence is "Lorem. Ipsum." and an OOD sentence is "Modelu Pronus." with answer "Modelu," the mixed sentence can be "Lorem. Modelu Pronus. Ipsum." We simply shift answer indices to generate a new OOD training point.

3.3 Selective Masking

First, we finetune a DistilBERT model with a Masked-LM head on the OOD training data. During QA training, we selectively mask random (non-answer) tokens of the context paragraph and use the fine-tuned LM to fill in the missing tokens. The intuition here is to modify non-essential parts of the context using alternate phrasing and vocabulary, which language models excel at. A robust model should be resilient to these spurious changes; thus, we hypothesize that these augmentations should teach the model to focus only on the important/relevant segments of the context. We fine-tune our ID-trained QA DistilBERT model (ID baseline) on all OOD training samples + generated samples.

3.4 Easy Data Augmentation (EDA)

We adapt EDA [8] as another data augmentation technique to improve OOD dataset distribution support. EDA consists of four simple text augmentation techniques: random synonym replacement, random insertion, random swap, and random deletion. To implement random synonym replacement, WordNet [16] is used to choose synonym words and replace then correspondingly. For random insertion, random words are chosen from WordNet and inserted with a given probability. For random swapping, a pair of words from the sentence are randomly chosen, and their positions in the sentence are swapped. For random deletion, a randomly chosen word is deleted with a given probability. We use the implementation written by the authors of EDA, at `github.com/jasonwei20/eda_nlp`.

3.5 Back-Translation

For this augmentation technique, we translate every non-answer sentence of a given OOD training sample context paragraph, as well as the question, into a pivot language (we use French, as it is data-rich). We then convert these passages back to English, generating a novel OOD training sample with the augmented context/question and original answer (with shifted indices). We use Helsinki-NLP's pre-trained machine translation pipeline from HuggingFace. The intuition here is similar to selective masking: back-translating a sentence is an excellent way to preserve meaning while modifying just the phrasing. By training on OOD samples augmented in non-essential ways, we teach the model to focus only on the important parts of the context, improving its robustness to spurious elements.

3.6 Self-Supervised Learning

This technique is inspired by and adapted from the concept of test-time training in computer vision [9], where vision models are trained on unlabeled data from target distributions by constructing dummy pseudo-tasks on these samples. In our case, we note that there is plentiful *un*-labeled data available from the OOD distribution, in the form of the OOD test set (i.e. contexts and questions, without answers). In order to enable the model to learn from this data, we construct **pseudo-QA tasks** consisting of fill-in-the-blank style questions in the following manner: 1) select a sentence in the context paragraph at random; 2) back-translate the sentence using the same approach as above in order to modify the phrasing; 3) select a phrase in the modified sentence at random to mask, repeating if the phrase does not show up verbatim in the original sentence; 4) construct the "question" as the modified sentence with the chosen phrase masked, and the "answer" as the masked phrase (making the question a fill-in-the-blank style, i.e. a cloze); 5) train the DistilBERT model on these pseudo-QA tasks constructed from the OOD test set. Note that we modify the phrasing of the chosen sentence before constructing the question/answer to prevent the model from simply learning to pattern-match the question text verbatim in the context (which will not work for real examples). After training on these pseudo-tasks, we fine-tune the model on the OOD training set, with real QA pairs. The intuition here is that the self-supervised pre-training allows the model to learn the distribution of the OOD dataset using a large support set of synthetically-constructed samples. Subsequent fine-tuning on the limited available labeled data from this domain is thus more successful, as the model has already learnt the OOD distribution and can leverage its knowledge to better optimize for the task at hand.

3.7 Combining Approaches

Because the above techniques (4 data augmentation techniques, and a self-supervised learning technique) all are modular and composable, we can combine them in order to train the model on as much OOD data (real and synthetic) as possible. In particular, we can combine the data augmentation techniques in any order: our approach is to simply take the best-val-performing checkpoint of one run/augmentation approach, and use that as the pre-initialization of the model for the subsequent run/augmentation approach. To combine self-supervised learning and data augmentation, we simply start by pre-training the model on the synthetically-generated pseudo-QA tasks from the OOD test set, and subsequently fine-tune on both real and augmented/generated OOD training data in any order.

3.8 Originality of Contributions

Data mixing and self-supervised learning are approaches that we came up with and implemented ourselves, inspired by similar approaches used in other domains to improve the robustness of models. Our selective masking and back-translation were taken from the provided reference handout for the RobustQA project; we implemented these ourselves. EDA was taken from [8], both idea and code.

4 Experiments

4.1 Data and Metrics

We use the staff-provided ID (SQuAD [1], NewsQA [6], Natural Questions [7]) and OOD (DuoRC [4], RACE [3], RelationExtraction [5]) datasets for QA, with training and validation splits for ID and train/val/test splits for OOD. All examples are of the form (context paragraph, question, answer), with the answer consisting of a start position and phrase. For evaluation, we use the standard F1 and Exact Match (EM) scores (implemented in staff-provided code) to measure the model's QA performance.

4.2 Training and Hyperparameter Details

For all of our experiments, we use the standard model configuration, i.e. a DistilBERT transformer model pre-trained for language modeling/re-purposed for QA, from HuggingFace. For ID baseline training, we use the default provided hyperparameters, i.e. learning rate 3×10^{-5}, batch size of 16, 3 epochs of training, AdamW optimizer, etc. For fine-tuning on the OOD training set, we perform 10 epochs of training with the same hyperparameters. This is also the case when training on additional OOD augmented data, as well as for self-supervised learning. For EDA, we use a lower learning rate in order to achieve the best results, at 3×10^{-6}. Whenever we use additional pre-trained language models (i.e. for selective masking, back-translation), we use the default hyperparameters provided by HuggingFace, modifying batch size as needed to fit each batch onto our GPU (NVIDIA Tesla V100).

Mixing data. We generate a new mixed dataset, which is the same size as the ID train dataset. For each new datapoint, we pick the ID question with half probability, and the OOD question with half probability. We use the corresponding answer to match the question. We then save this dataset to disk, and use this dataset in place of the ID dataset in fine-tuning the DistilBERT model for 3 epochs.

Selective Masking. We choose non-answer tokens to omit from the context paragraph independently with a probability of $p = 0.15$. The augmentations are generated on the fly during training.

Easy Data Augmentation (EDA). The probability for performing each type of text augmentation is $p = 0.1$. Because EDA operates on text sentences and not on the tokenized examples, the augmented copies are first saved to disk and then converted into the tokenized tensor. We augment each non-answer context sentence in the OOD train data 40 times, resulting in $41\times$ amount of data.

Back-translation. We use the Helsinki-NLP machine translation pipeline, trained using MarianNMT and OPUS data, from HuggingFace. The augmentations are also generated on the fly during training.

Self-Supervised Learning. For a given a context paragraph, we sample a sentence that is between 5 and 20 tokens long uniformly at random. We then back-translate this sentence using French as the pivot language, as above. Subsequently, we sample a start position uniformly at random in the modified sentence, check if the next two tokens are present verbatim in the original sentence, and select that as our answer (repeating until these conditions are met). We replace these answer tokens with underscores in order to create our question. Finally, we save our generated samples to disk.

Note: for data mixing only, we train the model on the generated dataset in lieu of the ID data. For all the others, we begin training/fine-tuning with the best-performing checkpoint of the ID baseline (on OOD val), and also train on the full OOD training data in parallel with the augmented training data.

5 Results and Analysis

5.1 Data Mixing

We train on the ID+OOD training samples and mixed OOD-ID augmented training samples, giving us an F1 score of 41.72 and EM score of 25.13 on the OOD val set. This technique was our first original method, as it is a form of mixup which combines the ID and OOD data and gives us sentences that come from the mixed distribution. However, our other techniques have since outperformed this.

We hypothesize several reasons why data mixing augmentation has not improved performance over the baseline. When ID and OOD data are mixed, the context texts become much longer, and there is the possibility that words get cut off. This may lead to important words being cut off, and thus hamper performance. While mixing in OOD data for training is likely beneficial, it may be that the mixed-in OOD data are still in the same text sentences, and thus the model does not obtain new information. Due to the memory limitation, the mixing dataset is only generated to be the same size as the ID dataset. Since we choose half for ID and half for OOD questions, this cuts out half the ID questions, which then reduces the quantity of available ID data. Finally, we hypothesize that data mixing is just not strong enough an augmentation/does not generate enough novel samples in the OOD distribution to learn from, as the distribution of the generated samples is mixed between ID

Method	F1 Score	EM Score
ID Baseline	47.96	31.94
ID+OOD Baseline	48.43	32.82
Data Mixing	41.72	25.13
Selective Masking	48.21	34.90
EDA	49.75	36.65
Back-Translation	48.98	35.08
Self-Supervised	**51.87**	**37.72**
EDA + Back-Translation	51.23	37.16
Self-Supervised + EDA	53.17	39.28
Self-Supervised + Back-Translation	52.91	38.75
Self-Supervised + EDA + Back-Translation	**54.29**	**40.11**

Table 1: Out-of-domain (OOD) validation set performance (F1, EM) across all methods experimented with, including combining our 3 most successful methods (EDA, Back-Translation, and Self-Supervised Learning). Combined techniques listed in order they were applied (i.e. first Self-Supervised, then EDA). Our best-performing technique combines all three of our most successful methods, and represents a significant improvement in performance over the ID and ID+OOD baseline.

Method	F1 Score	EM Score
Back-Translation	56.86	39.61
Self-Supervised + Back-Translation	60.12	42.96
Self-Supervised + EDA + Back-Translation	60.31	42.91
Self-Supervised (RACE), ID Baseline (DuoRC), EDA (RelExtract)	**62.99**	**45.05**

Table 2: Out-of-domain (OOD) test set performance for best-performing methods. Using a different one of our techniques for each OOD dataset (Self-Supervised Learning for RACE, ID Baseline for DuoRC, and EDA for RelationExtraction) results in our best-performing model, achieving 3rd on RobustQA leaderboard. Combining our methods also yields good performance (7th on leaderboard).

and OOD, and the generated samples are too similar to existing ID and OOD samples (since they are simply spliced together), meaning once again that the learning value of these samples is limited.

5.2 Selective Masking

After training our Masked-LM on the OOD train data for 6 epochs, we achieve a validation masked token prediction loss of 0.29 on OOD val data. The QA model is then trained on data modified by the Masked-LM, and achieves an Eval F1 of 48.21 and an Eval EM score of 34.90. While selective masking outperforms the ID and ID+OOD baselines with respect to EM, it falls short of the ID+OOD baseline with respect to F1 (Table 1). This is likely because we sample sentences from the ID dataset for training, and thus only complete masked tokens based on ID sentences. This may lead to generated sentences still strongly resembling ID data and not capturing the OOD data support.

5.3 Easy Data Augmentation (EDA)

After using EDA to augment the OOD training data and improve the OOD training support for the model, we were able to achieve significant outperformance on the OOD val set over the baselines. This confirms our hypothesis that data augmentation helps performance by enabling the model to better learn the OOD distribution; in fact, EDA was the best out of our data augmentation methods (outperforming back-translation). EDA achieves an OOD val F1 score of 49.75, EM score of 36.65.

We posit that EDA is able to generate a diverse set of example sentences, while still maintaining the key characteristics of the OOD distribution of which it models after. This differentiates it from selective masking and data mixing, since those techniques involve weaker augmentations that likely do not generate diverse enough samples even while accurately modeling the OOD distribution. As a

Method	All		RACE		DuoRC		RelExtract	
	F1	EM	F1	EM	F1	EM	F1	EM
ID Baseline	47.96	31.94	35.57	20.31	**43.38**	**35.71**	64.85	39.84
ID+OOD Baseline	48.43	32.82	34.36	20.31	36.53	27.78	75.07	51.25
EDA	49.75	36.65	32.25	18.75	40.22	32.54	**76.63**	**58.59**
Back-Translation	48.98	35.08	34.20	22.66	38.90	30.16	73.68	52.34
Self-Supervised	**51.87**	**37.72**	**38.78**	**27.12**	40.57	31.31	76.32	54.77

Table 3: Out-of-domain (OOD) individual dataset validation performance (F1, EM) for baselines and best-performing methods. All methods have the best performance on RelationExtraction compared to RACE and DuoRC. Compared to EDA, Self-Supervised Learning exhibits much better performance on RACE while maintaining roughly the same performance on both RelationExtraction and DuoRC.

result, EDA improves the OOD data support and provides novel samples with high learning value. Consider the following question from the OOD training data: "What will contribute to a satisfactory photo of a running lion in the wild?" EDA might augment this sentence to be as follows: "what will contribute to a satisfactory exposure of a running lion in the wild." This changes the text, while retaining similar structure and meaning as the original question and capturing the same semantics.

5.4 Back-Translation

Back-translating the context paragraphs and questions of OOD training samples also results in significant outperformance over the baselines, with the model achieving an F1 of 48.72 and EM of 34.82 on the OOD val data. Again, we hypothesize that back-translation generates diverse novel examples within the OOD data distribution, since the machine translation model is able to process large chunks of text (i.e. all non-answer sentences in the context) at once and thus has the freedom to introduce more variability and alternate phrasing through the 2 translation steps (as opposed to selective masking, where the LM only has limited flexibility and ability to introduce variations since it is only tasked with filling in a few tokens). Overall, back-translation, like EDA, also improves the OOD data support, enabling the model to better learn the distribution and achieve a higher val score.

The fact that EDA outperforms back-translation is a bit surprising, because back-translation uses a more expressive machine translation model to obtain the generated examples, whereas EDA uses very simple text manipulation to obtain additional text examples. This result may be due to the fact that there are not many variations of the back-translated text compared to the original text, or that the back-translation may contain very similar information already presented in the original text (thus, while the generated samples still exhibit good diversity, EDA achieves greater diversity within OOD).

For example, consider the following question from the OOD train data: "What will contribute to a satisfactory photo of a running lion in the wild?" Back-translating this sentence results in "What will contribute to a satisfactory photo of a lion being executed in nature?" While the phrasing at the end is different, most of the question remains the same. With EDA however, we get augmentations "running will contribute to a satisfactory photo of a what lion in the wild." The text and phrasing is much more different from the original, resulting in improved diversity of the augmented OOD support dataset.

All of our experiments thus far have utilized French as the pivot language in back-translation. We also experiment with other languages, and a combination of languages (to add even more data): German, Spanish, and Italian. Interestingly, adding back-translated samples from another language (i.e. in addition to the back-translated French examples) to further augment the OOD train set does not result in any significant performance improvement. This is surprising, as intuitively, back-translating using other languages should result in novel modifications to the phrasing given the differing idioms and conversational styles of each language. We hypothesize that additional back-translation does not result in any greater support diversity for the model: the additional training examples do not provide further learning value beyond the OOD training data + French-back-translated examples already present. We also experimented with using German, Spanish, and Italian back-translation on their own, and found only negligible changes in performance compared to French back-translation. This is a key observation: the specific back-translating language does not matter for performance, but combining data from using multiple languages does not necessarily improve performance.

Further, performing chained back-translation (i.e. English → French → German → English) actually results in a degradation in performance. Looking at the generated examples, we notice that the context paragraphs and questions become quite short compared to the original. This is interesting and expected, given that machine translation systems are incentivized to express concepts using the minimum number of tokens possible. Thus, chaining together these systems has the inadvertent affect of dramatically reducing the length of the passage, to the point where the original concepts are no longer captured. This, we hypothesize, causes performance to suffer, as the generated examples no longer have any learning value, and the model might even get confused by trying to optimize them.

5.5 Self-Supervised Learning

Self-supervised learning followed by fine-tuning on the OOD train data achieved the best performance out of all of our methods on the OOD validation set, with an F1 of 51.87 and EM of 37.72 (Table 1). This follows our hypothesis: self-supervised learning on the pseudo-QA tasks constructed from the large OOD test set enables the model to learn the OOD distribution and more effectively optimize on the OOD training data during fine-tuning. Like EDA and back-translation, self-supervised learning improves the size and diversity of the OOD data support, but to a greater degree since it enables the model to learn from large amounts of actual data from the OOD distribution (i.e. the test set) rather than synthetic examples. Thus, the model is able to learn the distribution better, rather than overfitting to the limited subset of the OOD distribution that the OOD training data + augmented data can cover.

We can visualize the types of pseudo-QA fill-in-the-blank training examples that our self-supervised algorithm generates. Consider the context paragraph in Appendix A.1, from the DuoRC test set. Our algorithm might select the sentence "One day as Kiara was playing, she fell into a small pond as Timon and Pumbba got her back onto land." Back-translating this sentence yields "Kiara was playing one day when she found herself in a small pond, so Timon and Pumbba got her back onto land." From here, we might select "small pond" as our answer, meaning that the question becomes "Kiara was playing one day when she found herself in a ___ ___, so Timon and Pumbba got her back onto land." Our algorithm then adds this context, question, and answer set to the self-supervised training set, and trains on all of these samples (nearly 10000 samples in total, as we repeat the question/answer sampling process twice per context) to learn the OOD distribution. We encourage future work to explore the limit of this improvement by sampling even more question and answer pairs per context.

In Table 3, we break down the OOD validation performance of each of our high-performing methods on each of the 3 OOD datasets (RACE, DuoRC, RelationExtraction, i.e. RE), to better understand the scores reported in Table 1. All methods (baselines, EDA, Back-Translation, Self-Supervised Learning) perform the best on RelationExtraction, followed by DuoRC, and worst performance on RACE. Comparing to the ID baseline, the ID+OOD baseline experiences a large jump in RE performance and slight drop in DuoRC performance; this indicates that the OOD training set well-covers the RE distribution and does not cover the RACE or DuoRC distributions well (especially DuoRC). In fact, the ID baseline, without training on any DuoRC data directly or indirectly, performs the best on DuoRC: perhaps the DuoRC training data distribution is quite divergent from the DuoRC test distribution to drive this phenomenon. Comparing to the baseline, EDA and Back-Translation both improve significantly RE and a bit on RACE, and achieve better DuoRC performance than the ID+OOD baseline. This once again supports our earlier hypotheses, and shows that data augmentation for DuoRC specifically can increase the diversity of the training set enough so as to improve val performance (i.e. bridge the divergence between the train and val distributions somewhat). Self-Supervised Learning exhibits a large improvement on RACE, while maintaining the performance of data augmentation on DuoRC and RE. This once again shows the power of self-supervised approach: the model is able to better learn all three of the data distributions by training on the pseudo-QA tasks, so the total gap between performance on the three val datasets is smaller than for any other method.

Our findings in Table 3 inspire our best-performing test leaderboard submission. To maximize our score, we use the best-performing technique (in terms of validation) on each OOD dataset in order to generate test predictions for that dataset. As a result, we use Self-Supervised + OOD fine-tuned model to predict RACE, ID Baseline to predict DuoRC (since any fine-tuning on actual DuoRC data seemed to perform worse), and EDA to predict RelationExtraction. Our test results are in Table 2.

5.6 Combining Approaches

We combine our most successful methods: Self-Supervised Learning, Back Translation, and EDA. We start with the ID baseline, use our self-supervised learning algorithm to fine-tune the model to the OOD distribution, and fine-tune the model on the OOD train data + augmented data (via back-translation and/or EDA). Table 1 shows the performance of every combination of these techniques, including all three together (i.e. ablation study). Using this, we can better understand the contributions and value-add of each technique towards the final performance. For example, Self-Supervised achieves F1 of 51.87, adding Back-Translation moves F1 up to 52.91, and adding EDA moves F1 further up to 54.29. After the model has learned the OOD characteristics via self-supervised learning, back-translation and EDA are more effective and allow the model to predict better given OOD questions. Figure 1 shows the pipeline of combining our three approaches, in terms of what happens when.

6 Conclusion

In this study, we develop 5 novel techniques to improve the robustness of neural QA systems to out-of-domain (OOD) inputs, given only a limited labeled OOD support. Our first four techniques involve various data augmentations, such as EDA and Back-Translation, which enhance the OOD training set by generating novel, diverse samples in the OOD distribution. Thus, with the increased OOD support that cover a greater portion of the distribution, the model is able to better fit the OOD characteristics and achieve better performance than our two baselines. Our final technique involves a novel Semi-Supervised Learning algorithm that enables the model to learn the OOD distribution from the copious amounts of unlabeled test data available. Subsequently, the model is able to better fit to the OOD training data and outperform data augmentation. Combining these three approaches yields an even greater improvement over the baseline, and breaking down the scores per OOD dataset reveals further insight into what techniques perform best on which distribution. Leveraging this insight, we utilize the best technique for each dataset and achieve third on the RobustQA test leaderboard.

We do note, however, several prominent limitations with our approaches. While Self-Supervised Learning works well when there is plentiful unlabeled data to learn from, this might not always be the case: many applications are data-constrained and might not even have enough unlabeled data. Further, both EDA and Back-Translation, while they succeed at improving dataset diversity, are still simply re-hashing existing examples and not generating truly novel, diverse samples. For an application where the available data only covers a small portion of the desired distribution, these techniques will still fall short, and the model will likely still not capture the full characteristics of the domain.

We encourage future research to further investigate the limits of the approaches we have proposed. In particular, we only tested Back-Translation with Romanic languages, whereas using more diverse languages (i.e. Asian/African languages) might result in even greater augmentation diversity. Further, we wonder if generating more question/answer pairs per context paragraph in our self-supervised learning algorithm, increasing the size of the pseudo-QA train dataset, might yield a further improvement.

References

[1] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

[2] Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. Reqa: An evaluation for end-to-end answer retrieval models. *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019.

[3] 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.

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

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

[6] 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.

[7] 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, 2019.

[8] Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196, 2019.

[9] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training for out-of-distribution generalization. *CoRR*, abs/1909.13231, 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] Wee Chung Gan and Hwee Tou Ng. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy, July 2019. Association for Computational Linguistics.

[12] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *CoRR*, abs/1804.07998, 2018.

[13] Siddhant Garg and Goutham Ramakrishnan. BAE: bert-based adversarial examples for text classification. *CoRR*, abs/2004.01970, 2020.

[14] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics.

[15] 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.

[16] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, nov 1995.

A Referenced Items

A.1 DuoRC Context Paragraph

DuoRC test set context paragraph example:

```
The film opens where the previous film ended, set a few years later, where Rafiki (Robert Guillaume)
gathers the animals of the Pride Lands together for the presentation of Simba (Matthew Broderick) and
Nalas (Moira Kelly) new daughter Kiara.  Mufasa's spirit (James Earl Jones) watches over the ceremony.
Later, Simba becomes very overprotective of an older Kiara (Michelle Horn), assigning Timon and Pumbaa
(Nathan Lane and Ernie Sabella) to watch her.  One day as Kiara was playing, she fell into a small pond as
Timon and Pumbba got her back onto land.  Kiara tells them only half of her is a princess.  Pumbba asks,
"Well, who's the other half?" While they wait for her to answer, they start having a snack.
```

A.2 Combined Training Pipeline

Combined training pipeline for Self-Supervised Learning + EDA + Back-Translation:

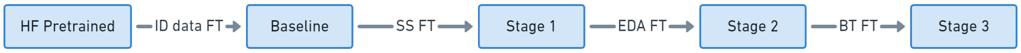

Figure 1: Training process pipeline for combined Self-Supervised Learning (SS), EDA, and Back-Translation (BT). FT means fine-tuning, and HF HuggingFace (where we get the pre-trained model).