Building QA Robustness Through Data Augmentation

Stanford CS224N {Default RobustQA} Project

Jake Chao
Department of Computer Science
Stanford University
jakechao@stanford.edu

Ryan Tran
Department of Computer Science
Stanford University
rtran56@stanford.edu

Abstract

While question and answering (QA) models have achieved tremendous results on in-domain queries, recent research has brought into question the ability of these QA models to generalize well to unseen data in other domains. As such, we aim to build a robust question answering system, which trained on a set of in-domain data can then be adapted to unseen domains given few training samples. In doing so, we explore the field of data augmentation. In this work, we conduct a survey of existing data augmentation methods, including backtranslation, synonym replacement, and synonym insertion as well as introduce a mixed data augmentation method (MDA) combining the previous three. In particular, we explore the efficacy of data augmentation in the task of question answering. We find that data augmentation provides moderate gains on our out of domain validation and test sets and that certain methods such as backtranslation and synonym replacement provide larger improvements compared to others. Overall, we confirm that data augmentation is a simple, generalizable technique with a wide variety of different methods that can effectively aid in improving the robustness of QA models in the face of unseen domains with few training examples.

1 Key Information to include

- Mentor: Mandy Lu

2 Introduction

In our paper, we seek to build a robust Question and Answering model. Although question and answering models have been explored in the past, and have been demonstrated to work well on in domain test data for models that are allowed to train on a large amount of in-domain training data, research shows that question and answering models build frail correlations that prevent generalization to domains outside the training data, especially in the case of outside domains with few available examples [1, 2]. In terms of building models comparable to human language capability, generalization is an important measurement of a model's ability to understand complex linguistic relationships and find solutions to difficult problems. As such, the focus of our research project is exploring methods through which we can develop robust QA models that can generalize to domains outside the training distribution.

Specifically, our project places focus on the area of data augmentation. Given a small set of out-of-domain training data, we attempt to generate more augmented data examples via several different data augmentation techniques. We then finetune our model on the examples from the few-shot out-of-domain data as well as the additional examples we generated via data augmentation. The goal of this data augmentation is to utilize our small set of out-of-domain training data to produce new examples for further training while simultaneously generalizing the distribution of the data to

prevent overfitting. Overall, we conduct a survey of a variety of data augmentation methods such as backtranslation as in [3] and [4], as well as synonym replacement and synonym insertion as in [5], specifically in the context of question answering. Through our research, we show that finetuning our models on the data-augmentation techniques have led to an increased F1 and EM on the overall validation datasets, some methods working more effectively than others.

3 Related Work

Overall, lack of generalization due to over fitting is a common problem faced when building machine learning models, and an increasingly popular method attempting to help develop robust models is data augmentation. Within fields such as NLP and computer vision, data augmentation is particularly useful in the context of building deep learning networks [6].

Data augmentation is useful in preventing NLP models from overfitting to training datasets. By widening the distribution of the training dataset through data augmentation, data augmentation pushes models to learn significant aspects of languages rather than syntactic specifics. For example, [7] replaces words in the training data examples with synonyms, and [8] expands upon the idea of synonym replacement by introducing a novel data augmentation technique known as "contextual augmentation", the replacement of words with their paradigmatic-related counterparts. In particular, both [7] and [8] find that their respective data augmentation methods produced positive, albeit sometimes marginal, results. Additionally, several works have explored the ability of data augmentation techniques to generate additional training examples, which is especially useful in the case of low-resource languages [9] and domains with very few training examples. One popular method of data augmentation explored in [3], [4], and [10] is backtranslation, the augmentation of a data example by translating the example to and from a target language in order to create new, semantically similar examples. [3] describes the benefits gained through backtranslation for their QA architecture QANet. On the other hand, [4] did not see any improvements for in-domain or out-of-domain performance.

The wide variety of methods and levels of success for each method motivates further exploration of data augmentation, and particularly, while other research has investigated tasks such as text classification, to our knowledge there exists no wide survey of data augmentation methods specifically in relation to QA, a gap that our project attempts to fill.

4 Approach

In our approach to building a more robust question-and-answer system, we used data augmentation to generate new examples to add to our out of domain dataset. Using data augmentation, we train on similar but syntactically different examples as to avoid overfitting that would occur if we simply trained repeatedly on the original data. The data augmentation techniques we explore are backtranslation, synonym replacement, synonym insertion, and a mixture of these data augmentation methods. For each example in the out of domain training data, we create additional augmented examples on which to train. We only augment the context for each method as to avoid changing the meaning of the question.

We utilize Spacy [1] to conduct tasks such as sentence parsing. Our coding contributions include writing the script to parse and modify each provided training example in order to generate additional augmented out-of-domain training data. Unless mentioned otherwise, we implement the augmentation of the examples ourselves.

4.1 Back Translation (BT)

In this method of data augmentation, we conduct standard backtranslation as in [3] and [4] by translating to a chosen target language through use of an external machine translation model and then translating back to the source language, which in all datasets explored is English. In order to conduct translation, we utilized the Marian model[2] and tokenizer [3] provided by HuggingFace. For

[1] https://spacy.io/api/docs

[2] https://huggingface.co/transformers/model_doc/marian.html#marianmodel

[3] https://huggingface.co/transformers/model_doc/marian.html#mariantokenizer

Data Augmentation	Example
Original Sentence	The dog walked to its home.
Backtranslation	**The dog headed for his house.**
Synonym Replacement	The dog walked to its **abode**.
Synonym Insertion	The dog **abode** walked to its home.

Figure 1: Examples of each data augmentation method, save for Mixed Data Augmentation which is a combination of the above three.

each given out-of-domain example, we conducted backtranslation using the Marian library with the Helsinki-NLP models [4] for each language in a subset of the following set of languages: {Spanish (ESP), French (FRA), Portugese (POR), Italian (ITA), Romanian (RON), German (DE), Indonesian (ID) }. Backtranslation creates a paraphrased version of the original example adding noise due to semantic and syntactical changes caused by the translation process.

4.2 Synonym Replacement (SR)

In the synonym replacement method, as used in [5], for every sentence in a context of a given example, we choose n random non-stop words from the sentence where $n = \alpha\ell$. α is a selected hyperparameter representing the percent of words to be replaced, and ℓ is the length of the sentence. For each of these n random words in the sentence, we then replace each word with a random synonym for these words found using the nltk WordNet [5]. After running synonym replacement for each sentence in a context, we then have created a modified synonym-replaced context. Each context is augmented n_{aug} times to create n_{aug} new examples.

4.3 Synonym Insertion (SI)

Synonym insertion method is similar to synonym replacement and is taken from [5]. However, for every sentence in a context, we randomly choose n non-stop words in the sentence, find a random synonym for each of those n words, and then insert these synonyms at random locations in the sentence. We repeat this process and generate n_{aug} new examples composed of synonym-inserted contexts, questions and answers.

4.4 Mixed Data Augmentation (MDA)

For mixed data augmentation (MDA), for each context in the out-of-domain training dataset, we randomly choose a method out of back-translation, synonym-replacement, and synonym-insertion to apply to the context. We chose these three methods based off inspiration from [3] and [5]. We apply the randomly selected method to augment the given context. Through mixed methods, we apply an assortment of these data augmentation methods to create more examples for our dataset. We select backtranslation with probability p_{bt}, synonym replacement with probability p_{sr}, and synoynm insertion with probability p_{si} where $p_{bt} + p_{sr} + p_{si} = 1$.

4.5 Answer Heuristic

An obvious concern with data augmentation through the above methods in the context of question answering is that the original answer span for the example will be changed and may no longer exist in the new context. In the case that the original answer text is preserved, we simply find the start and end of this text in the augmented context and utilize these logits as our answer span for the new augmented example. If the answer is changed, we utilize the method described in [3] and [4] to generate an approximate answer. Let s' be the newly generated sentence and a be the original answer. For both the start word w and end word e of a, we compute the 2-gram Jaccard similarity score between the start/end word and each word in s'. We then consider each pair of candidate start and end words in s'. The pair with the highest total Jaccard score become our new start and end words of the answer phrase, so our new answer then becomes the phrase in s' starting with the new start w' and end e'.

[4] See appendix A.1 for a complete list of models used.

[5] https://www.nltk.org/howto/wordnet.html

4.6 Baseline

The baseline model is the pretrained model DistilBert [10] finetuned for the downstream task of question answering on our provided in-domain datasets. We will refer to this model as our baseline. As an additional baseline to measure the contribution of our data augmentation methods, we took the baseline model (DistilBert finetuned on in-domain data) and continued finetuning with the original out of domain datasets. We will refer to this additional baseline as the OOD-FT baseline.

5 Experiments

5.1 Data

For finetuning, we utilize three in-domain question answering datasetse and three out-of-domain datasets. The in-domain QA datasets we use include SQuAD [11], NewsQA [12], and Natural Questions [13]. The out-of-domain QA datasets include DuoRC[14], RACE [15], and RelationExtraction[16]. We then augment the out-of-domain data to generate additional out-of-domain examples (the total amount depending on which method and parameters are used).

5.2 Evaluation method

We utilize the Exact Match (EM) and F1 scores as evaluation metrics. We evaluate performance on our provided out of domain validation datasets and note our final results on an out-of-domain test set.

5.3 Experimental details

For our model, we used the provided baseline DistilBERT model and then finetuned this model on the in-domain datasets SQuAD, NewsQA, and Natural Questions. After finetuning on in-domain data, we then finetuned this model on our out-of-domain datasets. These out of domain datasets included the originally provided DuoRC, RACE, and RelationExtraction datasets as well as additional augmented data produced by the data augmentation methods introduced in Section 4.

We selected specific hyperparameters for each method by experimenting with different learning rates and number of epochs. For each method, we experimented with learning rates ranging between $0.5e-5$ and $3e-5$ and with number of epochs from 3 to 7. In each section below, we denote the selected hyperparameters that provided the highest scores for each respective method. Across all methods, we utilize a batch size of 16 with a max sequence length of 384.

5.3.1 Backtranslation

For backtranslation, we used a learning rate of $1e-5$, and trained for 3 epochs. We selected these hyperparameters as we noticed that with higher learning rates and number of epochs, our model began to overfit on the data, leading to regression in F1 and EM scores. We ran several experiments varying the subset of languages translated to and from for each original data example. We experimented with subsets of languages from the following list: {Spanish (ESP), French (FRA), Portugese (POR), Italian (ITA), Romanian (RON), German (DE), Indonesian (ID) }.

5.3.2 Synonym Replacement

For synonym replacement, we used a learning rate of $1e-5$, and trained for 5 epochs. For α, the percentage of words which we replaced with synonyms in each sentence, we used a default value of 0.1 for all experiments, since this value generally worked well among all datasets and dataset sizes according to [5]. For n_{aug}, the number of new examples we generated per original example, we tested values of $n_{aug} = \{1, 2, 4, 8, 16\}$, which were also tested in the paper.

5.3.3 Synonym Insertion

For synonym insertion, we used a learning rate of $1e-5$, and trained for 5 epochs. For α, the percentage of words for which we found synyonyms to insert in each sentence, we used a default value of 0.1 for all experiments. $\alpha = 0.1$ generally worked well among all datasets and dataset sizes

Languages	DuoRC	RACE	RelationExtraction	Overall
Baseline	F1: 38.59 EM: 29.37	**F1: 40.04** **EM: 28.12**	F1: 66.51 EM: 42.19	F1: 48.432 EM: 33.246
OOD-FT Baseline	F1: 40.63 EM: 30.16	F1: 36.83 EM: 22.66	F1: 70.08 EM: 48.44	F1:49.23 EM: 33.77
{FRA}	F1: 39.79 EM: 29.37	F1: 36.97 EM: 24.22	F1: 71.14 EM: 50.78	F1: 49.35 EM: 34.82
{ESP, FRA, POR}	F1: 42.63 EM: 32.54	F1: 35.10 EM: 21.88	F1: 73.05 EM: 54.69	F1: 50.30 EM: 36.39
{ESP, FRA, POR, ITA, RON}	**F1 43.31** **EM: 32.54**	F1: 35.57 EM: 19.53	**F1: 75.45** **EM: 57.81**	**F1: 51.37** **EM: 36.39**
{ESP, FRA, POR, ITA, RON, DE, ID}	F1: 41.69 EM: 30.16	F1: 36.33 EM: 23.44	F1: 72.60 EM: 52.34	F1: 50.25 EM: 35.34

Table 1: F1 and EM scores on out-of-domain validation set for baseline models and models finetuned with data augmented using backtranslation to a different subset of languages.

according to the paper [5]. For n_{aug}, the number of new examples we generated per original example, we tested values of $n_{aug} = \{1, 2, 4, 8, 16\}$.

5.3.4 Mixed Data Augmentation (MDA)

For mixed data augmentation (MDA), we used a learning rate of 1e-5, and trained for 5 epochs. We utilize the top-performing parameters from each individual method of backtranslation, synonym replacement, and synonym insertion as shown in A.3. For the probability distributions p_{bt}, p_{sr}, and p_{si} from which we select the three methods backtranslation, synonym replacement, and synonym insertion respectively to apply to a given example, we tested the following set of probabilities for $\{p_{bt}, p_{sr}, p_{si}\}$: $\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}, \{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\}, \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}, \{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\}$.

5.4 Results

Below are the results we received from the experiments described in Section 5. We will first analyze each method's results individually and then discuss them holistically afterwards. For brevity, we have included data where deemed most relevant and necessary. For further experimental results not shown here, please see the Appendix for a list of results that are not included in the main report. We utilize our validation set to analyze our experiments (see Table 3 for an overview of our results) and comment on our performance on the test set later.

5.4.1 Backtranslation

As expected, we find that backtranslation provides significant gains on both F1 and EM scores as the number of languages used in the translation process increases, at least up until 5 languages used (those being Spanish, French, Portuguese, Italian, and Romanian). We have our highest scores on the validation set at 5 languages, achieving gains of approximately +2 and +3 points for F1 and EM scores respectively over the OOD-FT baseline, as displayed in Table 1. After adding German and Indonesian to reach a total of 7 languages, however, despite a continued increase in RACE, we witness a slight overall regression in F1 and EM scores. We hypothesize that this regression is due to either 1) the difference between German and Indonesian compared to the Romance languages used previously or 2) the model beginning to overfit on the augmented data.

5.4.2 Synonym Replacement

In Table 2, we find that all experiments for synonym replacement (regardless of n_{aug}) generally have a higher F1 and EM score than the baseline OOD-FT model. Increasing n_{aug} at first does not provide substantial gains to overall F1 and EM on our out-of-domain validation datasets. However, as we increase n_{aug} for values 8 and beyond, significant gains are continuously made to both F1 and EM. When $n_{aug} = 16$, F1 and EM reach their max scores (+3 F1, +4 EM compared to OOD-FT baseline). Based on our results, finetuning on synonym replacement augmented data and increasing n_{aug} (for values 8 and beyond) seems to improve F1 and EM scores as expected.

n_{avg}	DuoRC	RACE	RelationExtraction	Overall
Baseline	F1: 38.59 EM: 29.37	**F1: 40.04** **EM: 28.12**	F1: 66.51 EM: 42.19	F1: 48.432 EM: 33.246
OOD-FT Baseline	F1: 40.63 EM: 30.16	F1: 36.83 EM: 22.66	F1: 70.08, EM: 48.44	F1:49.23 EM: 33.77
1	F1: 41.71 EM: 30.16	F1: 33.17 EM: 20.31	F1: 76.10 EM: 57.03	F1: 50.37 EM: 35.86
2	F1: 40.67 EM: 28.57	F1: 35.84 EM: 23.44	F1: 73.78 EM: 53.12	F1: 50.15 EM: 35.08
4	F1: 39.92 EM: 29.37	F1: 35.99 EM: 21.88	F1: 71.37 EM: 51.56	F1: 49.14 EM: 34.29
8	**F1: 44.32** **EM: 32.54**	F1: 38.21 EM: 25.78	F1: 71.53 EM: 50.78	F1: 51.39 EM: 36.39
16	F1: 40.55 EM: 30.16	F1: 39.77 EM: 25.78	**F1: 75.17** **EM: 56.25**	**F1: 51.89** **EM: 37.43**

Table 2: F1 and EM scores on out-of-domain validation set for baseline models and models finetuned on augmented data using synonym replacement. n_{avg} represents the number of augmented examples generated per original data example.

5.4.3 Synonym Insertion

We find that overall for synonym insertion, regardless of n_{aug} used, F1 and EM scores are generally higher than the OOD-FT models'. For n_{aug} values from 1 to 8, increasing n_{aug} does not lead to an increase in F1 and EM. F1 and EM for these values are not significantly higher than that of OOD-FT baseline. This surprised us as previous data augmentation techniques seemed to show increases in F1 and EM as n_{aug} increased. We find, however, that $n_{aug} = 16$ still achieves the highest F1 and EM scores of 50.32 and 35.60 respectively (+2 F1, +2 EM higher than the OOD-FT baseline). Our results suggest that training on synonym insertion augmented data may improve the baseline's out-of-domain performance, especially if n_{aug} is large enough. See Appendix A.2 for full method results.

5.4.4 MDA

Regardless of what probabilities were set for p_{bt}, p_{sr}, p_{si}, all experiments for MDA produced moderate gains compared to both the baseline and the OOD-FT baseline. Analyzing the results, we see that the largest increase for MDA came when $p_{bt} = 0.5$, $p_{sr} = 0.25$, and $p_{si} = 0.25$, achieving F1 of 50.64 and EM of 36.39 with a gain of approximately +1 F1 and +3 EM on the OOD-FT baseline. From these results, we see that even utilizing a mixture of backtranslation, synonym replacement, and synonym insertion still, at least partially, carries the gains provided by the individual methods. What is interesting, however, is that MDA is unable to outperform the individual methods of backtranslation and synonym replacement. As MDA was inspired by EDA in [5], we expected that a mixture of methods would provide additional gains, but we hypothesize that our lack of gains relative to the individual methods stem from the random nature of MDA. See Appendix A.3 for full method results.

5.4.5 Test Set Results

In Table 4, we see the results of some of our data augmentation methods on our out-of-domain test set. Due to limited ability to test on the out-of-domain test set, we experimented with our top three performing data augmentation methods on the out-of-domain test set: backtranslation, synonym replacement, and MDA. Surprisingly, we found that compared to the results on the validation set in which SR performed the best, SR performed poorer than backtranslation, which performed best, and MDA. We attribute this to backtranslation providing the most noise (paraphrasing of an entire sentence compared to simple synonym replacement) and therefore, our model was less prone to overfitting on the training and validation data, leading to better results on the test set. Overall, we see that our best performing method of backtranslation achieved an F1 score of 59.278 and an EM score of 41.628, minimal gains in F1 and a moderate +1.5 increase in EM compared to the baseline. We were surprised to see the gap between the baseline model and our data augmentation methods narrow on the test set. We hypothesize that a number of factors led to this occurrence including overfitting of hyperparameters on the validation set and perhaps noise due to the similarities and/or

Augmentation Method	Parameters	DuoRC	RACE	RelationExtraction	Overall
Baseline	–	F1: 38.59 EM: 29.37	**F1: 40.04** **EM: 28.12**	F1: 66.51 EM: 42.19	F1: 48.432 EM: 33.246
OOD-FT Baseline	–	F1: 40.63 EM: 30.16	F1: 36.83 EM: 22.66	F1: 70.08, EM: 48.44	F1:49.23 EM: 33.77
Backtranslation	Languages: {ESP, FRA, POR, ITA, RON}	F1: 43.31 EM: 32.54	F1: 35.57 EM: 19.53	**F1: 75.45** **EM: 57.81**	F1: 51.37 EM: 36.39
Synonym Replacement	$n_{avg} = 16$	F1: 40.55 EM: 30.16	F1: 39.77 EM: 25.78	F1: 75.17 EM: 56.25	**F1: 51.89** **EM: 37.43**
Synonym Insertion	$n_{avg} = 16$	F1: 39.36 EM: 28.57	F1: 36.52 EM: 22.66	F1: 74.91 EM: 55.47	F1: 50.32 EM: 35.60
MDA	$p_{bt} = \frac{1}{2}, p_{sr} = \frac{1}{4}$, $p_{si} = \frac{1}{4}$	**F1: 43.67** **EM: 32.54**	F1: 34.94 EM: 21.88	**F1: 74.09** **EM: 54.69**	F1: 50.94 EM: 36.39

Table 3: The top results for F1 and EM scores on our validation set drawn from each of the individual methods. The parameters for each method are displayed above.

Data Augmentation Method	Test Scores
Baseline	F1: 59.187 EM: 40.275
Backtranslation	**F1: 59.278** **EM: 41.628**
Synonym Replacement	F1: 58.272 EM: 40.94
MDA	F1: 58.467 EM: 41.078

Table 4: EM and F1 Scores of different data augmenation methods on the out-of-domain test set.

differences between our training, validation, and test sets. Due to the random nature of many of our data augmentation techniques, it could be the case that the augmentation that did occur added noise helpful for the validation set but not necessarily so for the test set.

6 Analysis

6.1 Analysis of Individual Datasets

We notice some interesting trends in F1 and EM scores of the individual out-of-domain datasets: DuoRC, RACE, and RelationExtraction. We saw that the individual dataset which experienced the greatest improvements in F1 and EM was RelationExtraction followed by DuoRC. We further saw that data augmentation actually generally led to decreases in F1 and EM scores for the RACE dataset. We hypothesize that this may be because RelationExtraction contexts are often only one or two sentences. With single-sentence contexts, it can be easy to learn superficial aspects of language that overfit to the training set, so generalizing this distribution makes an overall larger impact on RelationExtraction than datasets with longer contexts such as DuoRC and RACE. Further, we also hypothesize that increases in F1 and EM scores for RelationExtraction and decreases for RACE may be due to the amount of noise introduced in the data augmentation process. Too much noise may be unhelpful. Since RACE contexts are usually longer than RelationExtraction and thus RACE contexts are usually modified many more times than RelationExtraction, there is more of a chance that erroneous noisy data augmentations, such as poor choices of synonyms, poor choices of synonym insertions, or bad translations, are more likely to be introduced with RACE than RelationExtraction.

6.2 Prediction Comparison

We observed the predictions of our baseline model and compared it to the the predictions of models finetuned on backtranslation, synonym replacement, or synonym insertion augmented data.

6.2.1 More Direct Answer

We find that data augmentation helps the model better understand and answer some questions. This is a general trend we notice and have included one example below which demonstrates this.

- **Context**: "The human TBR1 gene is located on the q arm of the positive strand of chromosome 2."
- **Question**: On what chromosome is TBR1 found?
- **Truth**: "chromosome 2"
- **Baseline Prediction**: "q arm of the positive strand of chromosome 2"
- **SR/BT/SI/MDA Prediction**: "chromosome 2"
- **Analysis**: Our baseline model does not directly answer what the question asked for, namely the chromosome where TBR1 is found, and includes unnecessary information. After finetuning on augmented data, the model is able to more directly answer the question with the correct answer, omitting the irrelevant information. We hypothesize that this may be because out-of-domain data-augmented examples helps the model better understand word semantics (e.g. different syntactic ways to phrase the same meaning). Thus, the model better understands context semantics and relationships between words within the context, helping it answer questions directly and correctly.

6.2.2 Failures on More Semantically Complex Contexts

However, finetuning on additional augmented data does not help the model answer more difficult questions, which may require much more insight into language than learnt from this finetuning.

- **Context**: "... The whole idea started when young Trey was called to come outside. He didn't because he was busy playing on the iPad. That's why his dad thought of the idea of living "in 1982" for a year. ..."
- **Question**: Who made the family have the idea of living "in 1982"?
- **Truth**:"young Trey"
- **Baseline Prediction**: "his dad"
- **SR/BT/SI/MDA Prediction**: "his dad"
- **Analysis**: We see that in this case despite finetuning on backtranslation, synonym replacement, or synonym insertion augmented data, the model is still not able to fully understand the semantics of the given context, particularly complex relationships between sentences and phrases, and produces the same, incorrect answer as the baseline model.

7 Conclusion

In our project, we experimented with applying various data augmentation techniques to our out of domain training data in order to generate more data to train our baseline model on. The data augmentation techniques we experimented included backtranslation (as described in [3] and [4]), synonym replacement, and synonym insertion (as described in [5]). We generally found that additional finetuning on these original data combined with these augmented datasets generally led to increases in F1 and EM performances of about +2 or +3 on the out-of-domain-validation dataset, with backtranslation, synonym replacement, and MDA generally yielding the largest improvements. On our test set, we did achieve increases in F1 and EM scores, albeit smaller than on the validation set.

One primary limitation of our work is that we randomly generated examples for our augmented datasets (e.g. synonyms are chosen randomly during synonym replacement and insertion, synonyms are inserted at random locations in synonym insertion), so noise could have been introduced from this process, making it more difficult to compare results between different datasets we experimented with. Given more time, we would run more thorough, repeated experiments to ensure the validity of our results. Another limitation is the answer heuristic utilized. 2-gram Jaccard similarity is admittedly a very basic heuristic to find an approximate answer, and in future work, we would consider utilizing a more complex, accurate heuristic for finding approximate answers when necessary. Lastly, we recommend conducting ablation studies to measure the extent to which data augmentation aids in improvement as opposed to simply the extra training allowed through the generation of new examples.

All in all, however, we find that data augmentation is a moderately effective way to generalize existing data to avoid overfitting while providing new examples in the case of low-resource domains. Some data augmentation methods are better than others, and we hypothesize that there exists even more potential with the exploration of more complex augmentation techniques.

References

[1] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *arXiv preprint arXiv:1707.07328*, 2017.

[2] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL)*, 2019.

[3] 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. In *arXiv:1804.09541*, 2018.

[4] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. In *arXiv preprint arXiv:1912.02145*, 2019.

[5] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *arXiv preprint arXiv:1901.11196*, 2019.

[6] Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. Data augmentation for visual question answering. In *Data Augmentation for Visual Question Answering*, 2017.

[7] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *arXiv:1509.01626*, 2015.

[8] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *NAACL-HLT*, 2018.

[9] Toms Bergmanis, Katharina Kann, Hinrich Schütze, and Sharon Goldwater. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, 2017.

[10] 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)*, 2016.

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

[12] Adam Trischler,Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *ACL 2017, page 191*, 2017.

[13] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. In *Association for Computational Linguistics (ACL)*, 2019.

[14] Mitesh M. Khapra, Amrita Saha, Rahul Aralikatte and Karthik Sankaranarayanan. Duorc:towards complex language understanding with paraphrased reading comprehension. In *Association for Computational Linguistics (ACL)*, 2018.

[15] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*, 2017.

[16] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *arXiv preprint arXiv:1706.04115*, 2017.

A Appendix

A.1 Models Utilized

We utilized the following models in our backtranslation:

1. Translation to Spanish, French, Portugese, Italian, Romanian was done with Helsinki-NLP opus-mt-en-roa [6].
2. Translation to German was done with Helsinki-NLP opus-mt-en-de [7].
3. Translation to Indonesian was done with Helsinki-NLP opus-mt-en-id [8].
4. Translation back to English was done with the inverse Helsinki-NLP model of each respective translation model. [9]

A.2 Synonym Insertion Results

n_{avg}	DuoRC	RACE	RelationExtraction	Overall
Baseline	F1: 38.59 EM: 29.37	**F1: 40.04** **EM: 28.12**	F1: 66.51 EM: 42.19	F1: 48.432 EM: 33.246
OOD-FT Baseline	**F1: 40.63** **EM: 30.16**	F1: 36.83 EM: 22.66	F1: 70.08, EM: 48.44	F1:49.23 EM: 33.77
1	F1: 40.13 EM: 30.16	F1: 36.11 EM: 21.88	F1: 69.65 EM: 47.66	F1: 48.67 EM: 33.25
2	F1: 40.20 EM: 29.37	F1: 34.46 EM: 21.09	F1: 73.95 EM: 54.69	F1: 49.59 EM: 35.08
4	F1: 40.55 EM: 28.57	F1: 33.05 EM: 20.31	F1: 74.48 EM: 55.47	F1: 49.41 EM: 34.82
8	F1: 37.26 EM: 25.40	F1: 34.05 EM: 21.09	**F1: 76.96** **EM: 57.81**	F1: 49.49 EM: 34.82
16	F1: 39.36 EM: 28.57	F1: 36.52 EM: 22.66	F1: 74.91 EM: 55.47	**F1: 50.32** **EM: 35.60**

F1 and EM scores on out-of-domain validation set for baseline models and models finetuned on augmented data using synonym insertion. n_{avg} represents the number of augmented examples generated per original data example.

A.3 MDA Results

Probabilities p_{bt}, p_{sr}, p_{si}	DuoRC	RACE	RelationExtraction	Overall
Baseline	F1: 38.59 EM: 29.37	**F1: 40.04** **EM: 28.12**	F1: 66.51 EM: 42.19	F1: 48.432 EM: 33.246
OOD-FT Baseline	F1: 40.63 EM: 30.16	**F1: 36.83** **EM: 22.66**	F1: 70.08, EM: 48.44	F1:49.23 EM: 33.77
$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$	F1: 40.89 EM: 27.78	F1: 36.31 EM: 23.44	F1: 74.89 EM: 53.91	F1: 50.75 EM: 35.08
$\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$	**F1: 43.67** **EM: 32.54**	F1: 34.94 EM: 21.88	F1: 74.09 EM: 54.69	**F1: 50.94** **EM: 36.39**
$\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$	F1: 39.85 EM: 28.57	F1: 34.93 EM: 22.66	F1: 75.18 EM: 54.69	F1: 50.04 EM: 35.34
$\frac{1}{4}, \frac{1}{4}, \frac{1}{2}$	F1: 38.91 EM: 28.57	F1: 35.19 EM: 18.75	**F1: 77.27** **EM: 59.38**	F1: 50.52 EM: 35.60

[6] https://huggingface.co/Helsinki-NLP/opus-mt-en-roa
[7] https://huggingface.co/Helsinki-NLP/opus-mt-en-de
[8] https://huggingface.co/Helsinki-NLP/opus-mt-en-id
[9] https://huggingface.co/Helsinki-NLP/opus-mt-roa-en,https://huggingface.co/Helsinki-NLP/opus-mt-de-en, https://huggingface.co/Helsinki-NLP/opus-mt-id-en

F1 and EM scores on out-of-domain validation set for baseline models and models finetuned on augmented data using mixed data augmentation. In each experiment, we use varying values of p_{bt}, p_{sr}, p_{si}.