NLP Generalization for QA Tasks

Rahul Kulkarni **Kevin Hanna** **Justin Stanely**

W210 Summer 2020, MIDS, UC Berkeley School of Information

04 August 2020

{rahul.kulkarni, kevinhanna, justinstanely}@berkeley.edu

Abstract

We are motivated to understand the challenges of generalization of natural language models on unseen data specifically for extractive question and answering tasks. We analyze existing studies for understanding the generalization gap, conduct experiments to further characterize the generalization gap and apply techniques used for improving model robustness on language models.

Keywords: NLP, generalization, model robustness

Introduction

Machine learning algorithms have impacted a broad range of industries and the outcomes of such algorithms have an impact on massive scale and repercussions for human safety. Prior work on measuring generalization and overfitting (Roelofs, Schmidt et al. 2019)[1] points out the shortcomings of current classification algorithms and their sensitivity to small shifts in the underlying data distribution. They point out that distribution shifts in data hinders the algorithm to generalize, leading to incorrect decisions and severe consequences, Inability to interpret why a ML algorithm arrived at a certain decision that is driven by empirical progress with little guidance from theory and lastly majority of published papers justify improved performance using key benchmarks with little explanation of how the improvements came about. They propose to empirically examine key theoretical pillars of Machine Learning to build algorithms that are reliable and robust by building accurate measurements for generalization. Building

upon this foundational research of understanding the distribution gap and robustness of machine learning models on unseen data, (Recht, Benjamin, et al. 2019)[2] demonstrated the brittleness of image models trained on ImageNet. Also (Miller, et. al., 2020)[3] demonstrated similar phenomenon on natural language models for extractive QA tasks. We are motivated by the overarching research of understanding and characterizing model robustness of machine learning models and we build upon this analysis in the context of natural language models for QA tasks.

Background

The hypotheses of Recht, Benjamin, et al. challenges the conventional wisdom of machine learning experiments and demonstrates the lack of adaptive overfitting in their experiments on CIFAR-10 and ImageNet, that show no diminishing returns associated with test set reuse. As also demonstrated by Miller, et. al. on Natural Language Models that exhibit similar behavior. (Miller, et. al. 2020)[3] demonstrated that SQuAD models exhibit robustness to some natural distribution shifts, though they still suffer substantial performance degradation on others such as the New York Times, Reddit and Amazon Datasets and the degradation being attributed to changes in the source text.
(Miller, et. al., 2020)[3] raises the following questions:

- What are some metrics to compare Q&A datasets to explain performance differences
- Why do models perform better on certain Q&A models vs others
- What is the interplay of training models with additional data and model robustness

Methods

Our research methods as shown in Figure 1 are based on exploring two key areas for natural language models, characterizing the lack of generalization on out-of-domain data

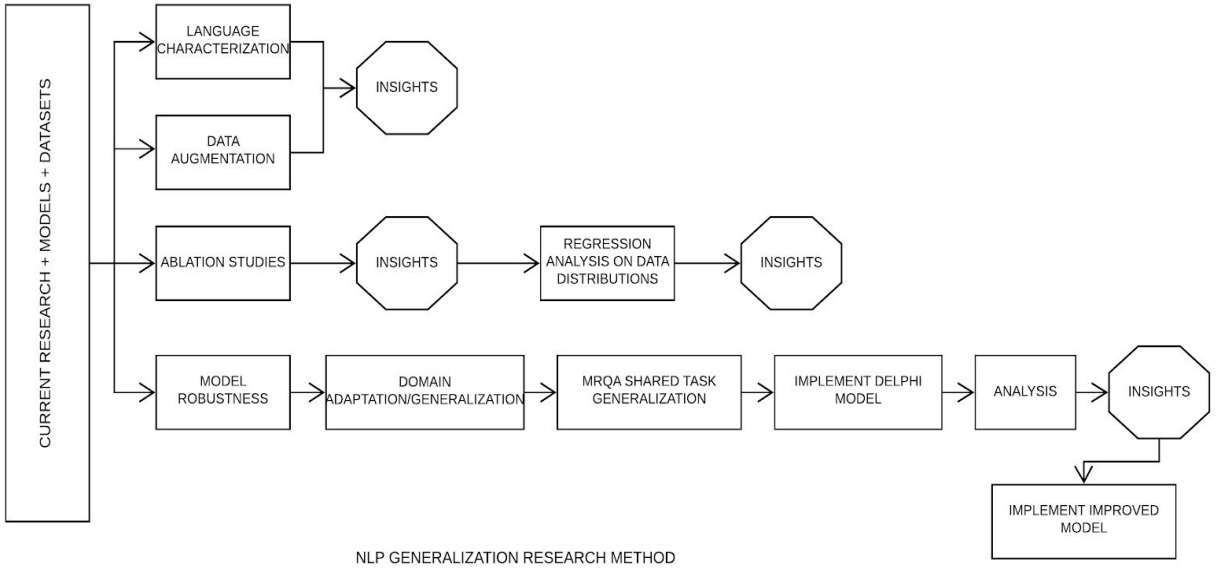

Figure 1

and applying techniques to analyze and improve model robustness. We categorize our research into key building blocks of language characterization and data augmentation, ablation studies on MRQA datasets and model robustness for shared task generalization.

Analyzing Lack of Generalization

Answer Characterization based on answer diversity, syntactic divergence and reasoning required do not explain model performance drops, and many models trained with more data do not exhibit

improved robustness but some do better. Metrics such as maximum mean Discrepancy (MMD) (Gretton, Arthur, et al.)[5] explain mean function values between two distributions, can we attribute larger MMD to a larger generalization gap? We explore metrics for language semantics and structure and data augmentation techniques to attribute the generalization gap between distributions.

Analyzing Model Robustness

Our initial analysis for model robustness on out-of-domain data is motivated broadly by Domain Adaptation and Domain Generalization. We explore techniques mentioned for Domain Adaptation in (Ma, Xiaofei, et al. 2019)[8] and Domain Generalization in MetaReg (Balaji et. al 2018)[9]

We analyze models from Machine Reading for Question Answering (MRQA) shared task (MRQA 2019 Shared Task)[6] that focuses on generalization. Some models from the MRQA shared task were particularly robust on unseen data as shown by (Miller, et. al. 2020)[3]. We further explore those models. We explore techniques such as negative sampling (Longpre, Shayne, et al.)[7] for language models.

Exploratory Data Analysis

The primary objective of our exploratory data analysis was to find a quantitative method for measuring differences between the text used to train the SQuAD question answering models and the out-of-domain test sets developed by (Miller, et. al. 2020)[3]. Such methods could be used to save users the time and cost needed to experiment with different pretrained models when selecting the best model for their specific task.

As previously mentioned, (Miller, et. al. 2020)[3] created four new test sets for use against SQuAD models derived, one from the original SQuAD source (Wikipedia) and three from different corpora: New York Times articles, Reddit posts, and Amazon product reviews. Their research found that, while higher F1 scores obtained by a given model on the original training data were predictive of higher F1 scores on the new test sets, the performance of those models decreased by an average of 3.8, 14.0, and 17.4 points, respectively, on the New York Times, Reddit, and Amazon sets.

We intuited that the type of language used in the professionally-written and edited New York Times articles, which performed so much better on average, would differ greatly from that used in amateur Reddit and Amazon review posts. We began exploring various lexical features of the text in the test sets to determine if any might be predictive of the F1 scores for the various SQuAD models; if so, those same metrics might lead to a straightforward method for selecting the best model for a given type of text.

We focused on two different types of metrics: simple descriptive measures of metrics like word counts and sentence lengths, and commonly-used complexity and readability scores used by educators. In most cases, those latter measures attempt to yield a score that roughly translates to the US grade-level the average human would need to reach in order to comprehend the given text. The formulas for these scores can be found in Table 1.

Flesch-Kincaid Grade Level	(.39 * mean word count per sentence) + (11.8 * mean syllable count per word)
Coleman-Liau Index	(0.0588 * mean character count per 100 words) - (0.296 * mean sentence count per 100 words)
Gunning Fog Index	0.4 * ((word count / sentence count) + ((polysyllable word count / word count) * 100))
Automated Readability Index	4.71 * (alphanumeric characters count / word count) + 0.5 * (word count / sentence count) - 21.43
Lexical Diversity	unique words / word count

Table 1

Sentence counts and word counts were derived using the Stanford CoreNLP sentence splitting and tokenization annotators with all the default settings.

Syllable counts were obtained by one of two methods: the Words API [25] or the PyHyphen [26] library. We first attempted to use the latter exclusively, but samples of the results indicated that the library was often incorrect, typically underreporting the number of actual syllables in a given word. To compensate, we first passed each unique word from the tokenization steps through the Words API instead and only used the PyHyphen library for those words for which the Words API failed to return syllable data. We also found that the Words API tended to underreport the number of syllables in hyphenated words by one, so we manually incremented the counts for those words. Future work might include using human-derived syllable data, or exploring more reliable sources.

Each of these metrics was calculated for the context text used in the original SQuAD training set, as well as each of the four new SQuAD test sets created by Miller, et al. Those metrics were then compared to the two evaluation metrics (exact match and F1 score) provided for each of the SQuAD model/test set combinations in an attempt to measure the correlation between each metric and the evaluation statistics.

Our results suggest that none of the text metrics analyzed have more than a weak correlation with the two evaluation metrics, either when comparing a specific model/test set combination or when looking at the test sets in aggregate. In fact, the greatest Pearson's correlation coefficient found among all combinations (polysyllable_count vs. the exact match score for the AllenNLP BiDAF single model against the new

Wikipedia test set) was only r = 0.083767, and most were much lower.

Our observations do show that the aggregated lexical diversity of the test set text corresponded to the average F1 scores obtained by the various SQuAD models. That is, the test sets that achieved the highest average F1 scores also had the highest lexical diversity value. We also note the same ranking when comparing the percentage of words appearing in the test set that also appeared in the training set as shown in Table 2.

Test Set	Lexical Diversity	% Shared Words
New Wikipedia	0.107	76.5%
New York Times	0.097	69.4%
Reddit	0.083	60.2%
Amazon	0.053	56.0%

Table 2

Data Augmentation

With the hypothesis that the distribution gap between corpora is a function of vocabulary, we performed two experiments to determine if the gap between the training corpus used to train question answering models, and the source of test set query context pairs could be narrowed through text augmentation. We set up two experiments which masked words and predicted replacements.

We opted to augment the test sets rather than training sets as our goal wasn't to increase the training set size, rather to attempt to narrow the distribution gap and develop a better understanding of the features of the gap. Augmenting the test sets allowed us to perform multiple iterations without performing any fine tuning. Augmenting the training sets would have required augmenting the training sets and fine tuning for each test set, becoming cost prohibitive.

We used two augmentation methods on the four test sets (Amazon, New Wiki, NYT & Reddit) developed by (Miller, et. al., 2020)[3]. Both methods consisted of masking and predicting replacement words in the context, question and answers using parameters to vary the word masking. Masked word predictions for each augmentation were performed using both Bert Large Cased and RoBERTa and the answer predictions were run using Bert Large both Cased & Uncased as well as DistilBERT both Cased & Uncased.

Experiment 1: Masking by part of speech.

In this first experiment, we used Natural Language Toolkit (NLTK)[22] to tokenize and identify the parts of speech for each token and replaced tokens from one or more parts of speech with the prediction model's mask token. The results were fairly positive as detailed in Table 3, especially when predicting masked words using Bert Large Cased and predicting answers with Bert Large Uncased.

	Bert Large Cased		**DistilBert Uncased**	
Parts of Speech	**F1**	**F1 Delta**	**F1**	**F1 Delta**
Original (Baseline)	80.0		72.3	
adverb	**82.3**	**2.4**	**74.5**	**2.2**
verb	**82.3**	**2.3**	**74.5**	**2.2**

verb, adverb	**81.1**	**1.1**	**73.5**	**1.2**
adjective	**81.7**	**1.7**	**73.8**	**1.6**
adjective, verb	80.5	**0.5**	73.0	**0.7**

Table 3: Results from Parts of Speech Augmentation.

Experiment 2: Masking by Wikipedia Word Frequency

For the next augmentation experiment, we used the same experiments setup, though, rather than masking words based on their part of speech, we masked words by the frequency they appeared in Wikipedia. With a single-word frequency distribution, we rated each token in the context by the frequency at which it was found in Wikipedia and masked tokens that fell below a specific threshold relative to tokens in the context. Words which appeared in wikipedia fewer that 3 times were left unchanged. This method produced very minor improvement, and only with the 10th percentile frequency as shown in Table 4.

	Bert Large Cased		**DistilBert Uncased**	
Percentile	**F1**	**F1 Delta**	**F1**	**F1 Delta**
Original (Baseline)	80.0		72.3	
10th Percentile	80.0	0.0	72.7	0.4
20th Percentile	75.9	-4.1	69.0	-3.3
30th Percentile	71.3	-8.7	64.9	-7.3
50th Percentile	59.9	-20.1	55.3	-17.0

Table 4: Results from Wikipedia Frequency Augmentation.

The results of Experiment 1 do suggest that the distribution gap can be reduced by replacing words which are predicted from a model trained on a similar corpus as the predicting model. However, there is little practical use for this method as it depends on having a trained model to perform the augmentations. Furthermore, analyzing the results at the question level shows that questions which performed poorly prior to augmentation performed more poorly on the augmented question context pairs, and vice versa. The change in mean F1 scores were not localized to changes in a small number of predicted answers, rather many predictions changed, and the aggregate results implied a small change leaving a lot of noise in the resulting data. This was amplified when looking at the specific test sets, the New Wiki and NYT test sets, which scored better F1 scores prior to being augmented improved slightly after augmentation, and the Amazon and Reddit test sets had a lower mean F1 score after being augmented as shown in Figure 2.

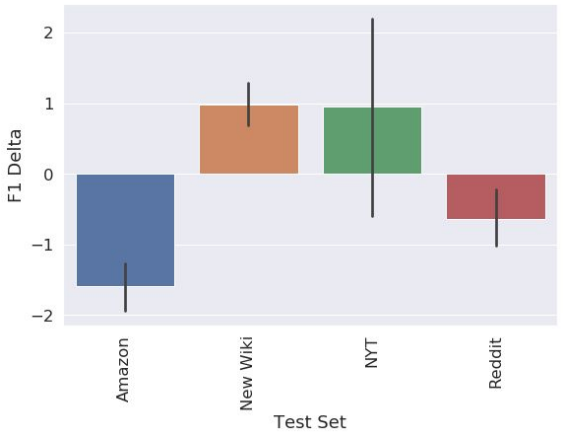

Figure 2: The change in mean F1 score caused by augmenting the test sets.

Ablation Studies

With several training sets available through MRQA[16] were able to perform an ablation study to understand how training sets contributed to models and their ability to generalize. We fine-tuned 43 permutations of those 6 training sets on Bert Base Cased and our new models to predict answers for 16 different test sets (6 MRQA in-domain, 6 MRQA out-of-domain, and 4 from Miller et. al. [3] creating 688 observations to analyze. A heatmap of 13 models (Figure 3)

demonstrates the relative influence a training set has on a model when it is used exclusively for fine tuning and when excluded from the training set. The diagonal starting at *Only SQuAD* demonstrates the influence of in-domain models, and the inverse starting at *No SQuAD* shows how out-of-domain models are penalized.

Using a model fine-tuned using all 6 training training sets as our baseline, we measured the change in F1 scores for each of the resulting models. SQuAD had significant influence on the resulting models. Of the models fine-tuned exclusively on each MRQA training set, the one trained on SQuAD had a mean F1 score closest to the baseline model, the inverse also proved true. Six models each trained excluding only one of the MRQA training sets, the one which excluded SQuAD had the lowest mean F1 score. The least influential training sets were TriviaQA and SearchQA, the former having the least impact when excluded from the training set, and the later having the lowest mean delta in F1 score (Table 5).

	Trained Only With		Trained Without	
Model	Mean F1	F1 Delta	Mean F1	F1 Delta
Baseline	68.77		68.77	
SQuAD	54.71	**-14.05**	61.30	**-7.47**
NewsQA	53.40	-15.36	66.81	-1.95
TriviaQA	40.40	-28.36	68.25	**-0.51**
SearchQA	31.30	**-37.47**	67.08	-1.68
HotpotQA	46.73	-22.03	67.30	-1.46
NaturalQuestions	49.87	-18.90	65.54	-3.22

Table 5

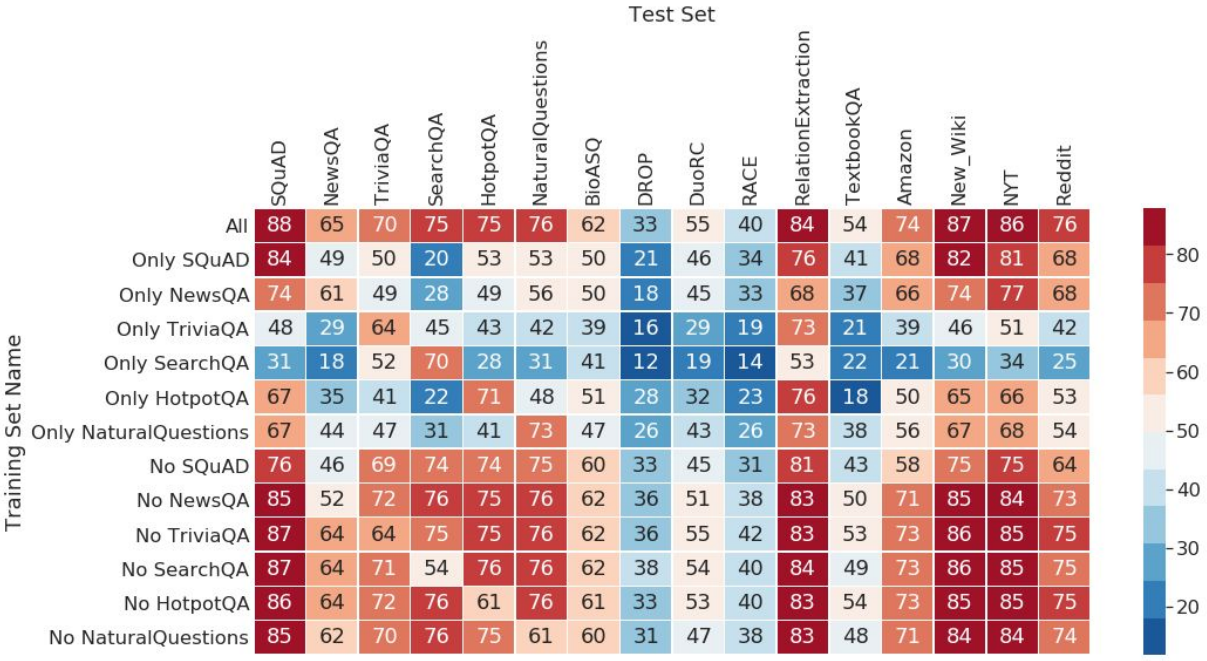

Figure 3: Heatmap of F1 Scores for fine-tuned BERT Base Cased models, the top model is fine-tuned on all six MRQA training sets, the next six are fine-tuned directly on each training set, and the bottom six are fine-tuned on five of the six training sets.

The generalizable observations from the ablation study were; the number training sets used in training correlated with the mean F1 scores, the test sets scored significantly higher when the model was trained using its related training set, as

those two data sets have a narrow distribution gap. The specific observations were that SQuAD was the most influential training set, and training sets sourced from *web snippets* were the least influential.

The N-Gram Frequency Distribution Analysis

To further explore the F1 score observations identified above, we created n-gram frequency distributions of each training set (1-gram, 2-gram, 3-gram, 4-gram and 5-gram), and several measures of our training and test sets, allowing us to perform regression analysis. It is important to note that, although there are 688 observations, these all come from 6 training sets, thus any biases in those training sets will be prevalent throughout the entire observations dataset and there is a high likelihood for multicollinear relationships between parameters. A summary of the parameters can be found in Appendix A and examples of the measurements and metrics in Appendix B.

In our first regression analysis, we looked at several metrics to build intuition about our data, this model results in an R-Squared of 67.5%, results in Table 6.

	coef	std err
Intercept	51.17	2.27
In-Domain	10.07	2.96
Mean Test Context Tokens	-13.08	1.68
Total Training Examples	22.62	3.33
5 Gram Distribution Intersection	49.65	42.38
In Domain Interaction Term for 5-Gram Intersection	-30.31	42.79
Unique N-Grams	-7.21	2.29

Table 6: Regression analysis coefficients.

In-Domain: The In-Domain parameter indicates whether the model was trained on a corpus which included the training set related to the test set being scored. This positive correlation to the F1 score is indicative of the distribution gaps, without any distribution gap, this parameter would show little correlation.

Mean Test Context Tokens: We observed that the Mean Test Context Tokens (the mean number of tokens in the contexts of the test set) correlates negatively with the mean F1 score, this is intuitive as the goal of the model is to identify the answer within the context, a smaller context decreases the potential incorrect predictions.

Total Training Examples: This positive correlation indicates that the more training examples provided the better the model should perform.

5-Gram Distribution Intersection Ratio: This parameter indicates how much of the test set overlaps with the training set. There is a positive correlation, however the large standard error makes this unreliable, the relationship could be spurious, though seems less likely when explored further below. With the In-Domain interaction term, we can see the coefficient is biased when in-domain training sets are used in the model. Without the interaction term, the standard error drops significantly, however, the relationship between overlap in distribution n-gram frequency distributions is viewed as a confounding relationship. This parameter is more relevant using exclusively out-of-domain models, which we explore below.

Unique N-Grams: This parameter represents the breadth of the vocabulary

used in training sets, the negative correlation is suggestive that smaller vocabularies lead to better model performance, however the standard error is relatively high, making inferences about the coefficient questionable.

N-Gram Frequency Distribution Intersection Ratio

Five n-gram frequency distributions were created and the intersection ratios calculated for each observation. The different ratios are multicollinear, thus only one can be included in each regression. The 1-gram contributes the most to resulting models' r-squared when in-domain models are excluded, we will limit regressions to the 1-gram, however, as they are multicollinear, each frequency distribution intersection ratio has a similar effect to our models.

As described above, when included with other parameters, the n-gram frequency distribution intersection ratio has a high standard error. However, those parameters can be used to describe 75% of the variance in n-gram frequency distribution ratio, much of that is a result of the In-Domain interaction term. The ratio for in-domain models will be 1.0. A better evaluation of the ratio then needs to exclude in-domain models. When those models are excluded, we find the 1-Gram Distribution Intersection Ratio explains 9.3% of the variance in F1 scores.

	coef	std err
Intercept	44.80	2.25
1 Gram Distribution Intersection	13.10	3.68

Table 7: Intersection and coefficient from 1-gram frequency distribution intersection ratio to F1 Scores.

When other parameters are included in the regression, the coefficient of the intersection ratio decreases to 8.19, and the standard error remains similar at 3.55. N-gram frequency distributions do provide some insight into the NLP Distribution Gap. Where the goal is to create a model that generalizes well, an n-gram frequency distribution for the general distribution, which would be unfeasible.

Test Set Mean Context Tokens

There is a negative correlation for the Mean Context Tokens in test sets, this seems intuitive as the probability of selecting the correct answer in the context is better when the context is smaller. We find that up to 14% of the variance in F1 scores can be explained by the mean context tokens of the MRQA test sets.
However, there are some potential confounding factors. The mean context tokens for data sets sourced from expertly written text happen to be among the smallest, and the largest mean context tokens are sourced from web snippets. If it is true that models perform better on expertly written text than more casually written text, this would bias our results.

Model Robustness and Generalization

We were motivated to analyze models trained on diverse datasets and to validate if such models are more robust and generalizable. Unlike SQuAD training and evaluation methods, MRQA Shared Task primarily focuses on Generalization and MRQA models are trained on six different datasets and evaluated on six out-of -domain (OOD) datasets. MRQA evaluations are done based on Exact Match and F1 scores similar to SQuAD albeit on OOD datasets.

As observed in (Miller, et. al. 2020)[3] the MRQA Delphi Model (Longpre, Shayne, et al.)[7] achieves a higher F1 score than any SQuAD models and benefits substantially by training with additional data.

The Delphi model benefits are attributed to the negative sampling techniques used by the model during training. Though SQuAD 2.0 have extended extractive question answering to include a No Answer option, in typical QA datasets there is no notion of a negative class. MRQA guarantees that there is an answer span found in each example context. For long contexts Delphi splits the contexts into multiple segments of size S which results in a set of (question, sub-context) pairs from the original lengthy context where each sub-context is a smaller segment of the original context. This split naturally introduces No answer contexts. These sub-contexts then are trained as an "abstention option" for the model and the segment indices for the start and end spans point to the [CLS] token. Inference is done based on computing the highest probable answer span from the set of all the (question, sub-context) pairs by excluding the No answer option.

Delphi Replication for Evaluation:
We replicated the MRQA Delphi prediction and evaluations on codalab [10]. The results of our evaluations and details of the experiment are documented in *Appendix C*.

Delphi Replication for Training:
The Delphi model[1] is based on the HuggingFace Transformers code base [11] Their experiments involved using BERT (Devlin, Jacob, et al. 2018)[14] for expensive sampling explorations and XLNet (Yang, Zhilin, et al. 2019)[24] for their final submission.

We implemented the negative sampling techniques for No answer questions in the MRQA training set based on the Delphi paper. We used the huggingface version 3.0.0 codebase to build the model. The huggingface code for pre-trained transformers is very modular and requires us to implement three key aspects of the model: Configuration, Tokenization, Modeling and additionally a dataset processor for pre-processing the data and a metrics processor for evaluations.

Our replicated Delphi model can be trained on MRQA training data for further analysis. The details of the model code are listed in *Appendix C*

We intend to continue the analysis of this model, first by measuring the evaluation benchmarks to be comparable with the original Delphi model benchmarks on the MRQA test-sets and validating our model code from the Delphi team. We further would like to validate the hypothesis from the Delphi authors on why including the NA segments yields significant improvements. What the authors (Longpre, Shayne, et al.)[7] hypothesize as quoted from their paper "We hypothesize this is primarily because a vaguely relevant span of tokens amid a completely irrelevant NA segment would monopolize the predicted probabilities. Meanwhile the actual answer span likely appears in a segment that may contain many competing spans of relevant text, each attracting some probability mass". To measure this quantitatively, we would need to capture start and end span probabilities of all segments including the NA segments and validate if the

[1] Due to legal restrictions, Delphi Team from Apple has not published their training code.

hypothesis holds true about the not-relevant span of tokens amassing predicted probabilities.

Future Work

Based on the research done during our Capstone, We strongly feel that we have established a strong foundation to continue making significant strides in some key areas of interest namely the MRQA model robustness and analysis for generalization.

Some key questions we would like to pursue

- The hypothesis of non-relevant span tokens amassing predicted probabilities leading to lower scores
- Why NA segments lead to improved model robustness for large context sizes
- Does the mean context tokens in the training sets have any significant impact on the resulting models.

We intend to contribute our MRQA model code back to the huggingface transformers library[11]. We also intend to collaborate with the authors from (Miller, et. al. 2020)[3] on characterizing the MRQA models further.

Conclusion

Our experiments to characterize the generalization gap using language characteristics using multiple descriptive measures of context complexity and corresponding F1 score showed no evidence of correlation. We see some evidence between lexical complexity of a corpora and a model's ability to make accurate predictions, but given the small sample size of 4 OOD test sets does not allow us to make a conclusive argument.
Our Data Augmentation experiments reduced the distribution gap by a small fraction and there is some room for improvement in predicting replacement tokens. In our Ablation studies our in-domain models out performed the out-of-domain models and SQuAD was the most influential training set and the most successful test set. We can only surmise based on this result that models trained on expertly written text such as news sources and Wikipedia perform better than models trained on sources such as Reddit and Amazon Reviews. Our Regression analysis shows some evidence that smaller contexts in the training set result in better F1 scores but we would need to run additional experiments reducing large contexts to fewer tokens so that each training set has similar mean context tokens and would require further analysis.

The strongest evidence of model robustness on out of domain datasets is shown by the MRQA Delphi model which uses No Answer questions for fine tuning pre-trained transformers, our research continues to focus on using these techniques to characterize and improve its behaviour further.

Acknowledgements

We would like to thank Alberto Todeschini and Puya Vahabi for introducing us to the original problem statement of Model Robustness and Generalization. We would like to thank John Miller and Ludwig Schmidt for their immense support for our research by providing us feedback and sharing their source code. We would also like to thank the Apple Delphi Team for

answering all our questions on the Delphi model implementation

References

1. Roelofs, Rebecca. *Measuring Generalization and Overfitting in Machine Learning*. Diss. UC Berkeley, 2019.
2. Recht, Benjamin, et al. "Do imagenet classifiers generalize to imagenet?." *arXiv preprint arXiv:1902.10811* (2019).
3. Miller, John, et al. "The Effect of Natural Distribution Shift on Question Answering Models." *arXiv preprint arXiv:2004.14444* (2020).
4. Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." *arXiv preprint arXiv:1606.05250* (2016).
5. Gretton, Arthur, et al. "A kernel two-sample test." *The Journal of Machine Learning Research* 13.1 (2012): 723-773.
6. https://mrqa.github.io/shared
7. Longpre, Shayne, et al. "An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering." *arXiv preprint arXiv:1912.02145* (2019).
8. Ma, Xiaofei, et al. "Domain Adaptation with BERT-based Domain Classification and Data Selection." *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. 2019.
9. Balaji, Yogesh, Swami Sankaranarayanan, and Rama Chellappa. "Metareg: Towards domain generalization using meta-regularization." *Advances in Neural Information Processing Systems*. 2018.
10. https://codalab.org/
11. https://github.com/huggingface/transformers
12. Bowman, Samuel R., et al. "A large annotated corpus for learning natural language inference." *arXiv preprint arXiv:1508.05326* (2015).
13. Cer, Daniel, et al. "Universal sentence encoder." *arXiv preprint arXiv:1803.11175* (2018).
14. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
15. Henderson, Matthew, et al. "Efficient natural language response suggestion for smart reply." *arXiv preprint arXiv:1705.00652* (2017).
16. Yang, Yinfei, et al. "Learning semantic textual similarity from conversations." *arXiv preprint arXiv:1804.07754* (2018).
17. Adiwardana, Daniel, et al. "Towards a human-like open-domain chatbot." *arXiv preprint arXiv:2001.09977* (2020).
18. Fast, Ethan, et al. "Iris: A conversational agent for complex tasks." *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018.
19. Ma, Xiaofei, et al. "Domain Adaptation with BERT-based Domain Classification and Data Selection." *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. 2019.
20. Dehghani, Mostafa, et al. "Learning to transform, combine, and reason in open-domain question answering." *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 2019.
21. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.
22. Fisch, Adam, et al. "MRQA 2019 shared task: Evaluating generalization in reading comprehension." arXiv preprint arXiv:1910.09753 (2019).
23. Loper, Edward, and Steven Bird. "NLTK: the natural language toolkit." arXiv preprint cs/0205028 (2002).
24. Yang, Zhilin, et al. "Xlnet: Generalized autoregressive pre training for language understanding." *Advances in neural information processing systems*. 2019.
25. https://www.wordsapi.com/
26. https://pypi.org/project/PyHyphen

Appendix A: Distribution & Data Set Measurements and Metrics

Available for download as a gzipped Parquet file:
https://nlp-distribution.s3.ca-central-1.amazonaws.com/ablation/results/model_scores_meta.parquet.gz

- **model_id**: ID assigned before training
- **test_set_name**: The Question Answering Test Set used to predict and score answers on.
- **f1**: The F1 score per SQuAD eval v1.1.
- **exact_match**: The Exact Match score per SQuAD eval v1.1.
- **model_name**: A name given to the model, which is the training set names joined with "_".
- **SQuAD, NewsQA, SearchQA, Hotpot, NaturalQuestions**: Boolean whether model included training set.
- **training_set_count**: Number of training sets used during fine tuning.
- **model_trained_sets**: List of training sets used in fine tuning.
- **model_nickname**: Shorter name generated for plotting.
- **mean_training_context_length**: The mean *character* length of the context in the training set.
- **mean_training_context_tokens**: The mean number of tokens in the training set.
- **mean_test_context_length**: The mean *character* length of the context in the test set.
- **mean_test_context_tokens**: The mean number of tokens in the test set.
- **total_training_contexts**: The number of contexts in training sets.
- **total_training_examples**: The number of questions in the training sets (some training sets have 1 question per context, others have more).
- **total_test_contexts**: The number of contexts in the test set.
- **total_test_examples**: The number of questions in the test set (some test sets have 1 question per context, others have more).
- **unique_training_ngrams**: The number of contexts in the test set.
- **total_training_keys**: The number of n-grams in training distribution.
- **intersection_score_#gram**: The sum of n-grams in intersection/the number of n-grams in training related to the test set.
- **in_domain**: Whether the training set related to the test set was used in fine tuning.

Appendix B: Test Set Stats

Test Sets

Test Set Name	F1 Score	Exact Match	Mean Context Tokens
RelationExtraction	78.81	65.79	24.80
NYT	76.34	66.32	129.21
SQuAD	76.22	66.35	122.76
New_Wiki	75.33	63.32	120.13
Reddit	65.16	52.18	141.36
NaturalQuestions	64.17	51.53	158.84
Amazon	63.04	49.13	145.41
HotpotQA	62.86	46.66	200.10
TriviaQA	62.50	55.24	701.02
SearchQA	57.22	49.38	644.22
BioASQ	55.77	40.49	209.15
NewsQA	51.55	37.59	489.54
DuoRC	45.42	36.50	601.81
TextbookQA	40.93	33.16	585.49
RACE	32.70	22.77	296.99
DROP	28.60	19.87	201.33

Training Sets

Model Fine Tuned on Single Training Set	F1 Score	Exact Match	mean_training_context_tokens	total_training_contexts	total_training_keys
SQuAD	54.71	43.79	116.66	18,885	123,212
NewsQA	53.40	39.76	492.19	11,428	116,980
NaturalQuestions	49.87	37.07	152.06	104,071	248,199
HotpotQA	46.73	35.79	159.15	72,928	239,492
TriviaQA	40.40	31.50	699.75	61,688	723,288
SearchQA	31.30	24.09	648.10	117,384	1,079,954

Appendix C:

MRQA Delphi Evaluations

	Exact Match	F1
Amazon	76.55	88.60
New Wiki	84.15	92.51
NYT	86.98	93.57
Reddit	77.84	88.36

MRQA Delphi Experiment Details

MRQA Delphi Predictions Results	https://worksheets.codalab.org/bundles/0x63119a4cbc984f6fb6b233a5b5975716
MRQA Delphi Evaluation Results	https://worksheets.codalab.org/worksheets/0x98a6d3c9a5cc488e87f7f0a749f6e370
Command to Generate Predictions for Delphi Model	cl mimic old_testset_uuid prediction_bundle_for_old_testset new_testset_uuid cl mimic 0x51ce3f58fce84197ad42d4414e3ec086 0x9a53e9c50f1244699c4a24aee483bd4c 0x7ecf931adac34c7e9cc52b5ecc058af8
Bert Baseline MRQA Command	(clab) (base) ubuntu@ip-172-31-19-45:~/mids/clab/models/MRQA-Shared-Task-2019/baseline$ pwd /home/ubuntu/mids/clab/models/MRQA-Shared-Task-2019/baseline python -m allennlp.run train MRQA_BERTbase.jsonnet -s Models/SQuAD -o "{'train_data_path': '/home/ubuntu/mids/clab/models/data/mrqa/train/SQuAD.jsonl.gz','validation_data_path': '/home/ubuntu/mids/clab/models/data/mrqa/dev-in/SQuAD.jsonl.gz','trainer': {'cuda_device': -1, 'num_epochs': 1, 'optimizer': {'type': 'bert_adam', 'lr': 3e-05, 'warmup': 0.1, 't_total': 29000}}}" --include-package mrqa_allennlp

	Change cuda_device param for gpu
	python -m allennlp.run train s3://multiqa/config/MRQA_BERTbase.json -s ../Models/MultiTrain -o "{'dataset_reader': {'sample_size': 75000}, 'validation_dataset_reader': {'sample_s ize': 1000}, 'train_data_path': 'https://mrqa.s3.us-east-2.amazonaws.com/data/train/SQuAD.jsonl.gz,https://mrqa.s3.us-east-2.amazonaws.com/data/train/NewsQA.jsonl.gz,https://mrqa .s3.us-east-2.amazonaws.com/data/train/HotpotQA.jsonl.gz,https://mrqa.s3.us-east-2.amazonaws.com/data/train/SearchQA.jsonl.gz,https://mrqa.s3.us-east-2.amazonaws.com/data/train/T riviaQA-web.jsonl.gz,https://mrqa.s3.us-east-2.amazonaws.com/data/train/NaturalQuestionsShort.jsonl.gz', 'validation_data_path': 'https://mrqa.s3.us-east-2.amazonaws.com/data/dev/SQuAD.jsonl.gz,https://mrqa.s3.us-east-2.amazonaws.com/data/dev/NewsQA.jsonl.gz,https://mrqa.s3.us-east-2.amazonaws.com/data/dev/HotpotQA.jsonl.gz,https://mrqa.s3.us-east-2.amaz onaws.com/data/dev/SearchQA.jsonl.gz,https://mrqa.s3.us-east-2.amazonaws.com/data/dev/TriviaQA-web.jsonl.gz,https://mrqa.s3.us-east-2.amazonaws.com/data/dev/NaturalQuestionsShort .jsonl.gz', 'trainer': {'cuda_device': [0,1], 'num_epochs': '2', 'optimizer': {'type': 'bert_adam', 'lr': 3e-05, 'warmup': 0.1, 't_total': '120000'}}}" --include-package mrqa_all ennlp

MRQA Training Code

Replicated Delphi Model Training code based on Huggingface Version 3.0.0	https://github.com/rahul-kulkarni/W210-Capstone/tree/master/src/mrqa/transformers

Appendix D

CodaLab Worksheets for Predictions and Evaluations

Predictions Official

Amazon	
NYT	https://worksheets.codalab.org/worksheets/0xba3eb032b1724048a6ddb11a36580216.
Reddit	https://worksheets.codalab.org/worksheets/0xe7c36d3ef3e9470aafdd4ee8c6bdb381
New Wiki	https://worksheets.codalab.org/worksheets/0xba3eb032b1724048a6ddb11a36580216.
Delphi	https://worksheets.codalab.org/bundles/0x9a53e9c50f1244699c4a24aee483bd4c

Evaluations Official

Amazon	https://worksheets.codalab.org/worksheets/0x412235fe516945fa81d04e6938109f0b
NYT	https://worksheets.codalab.org/worksheets/0xbd5582770bcf45558f038575a79ace63
Reddit	https://worksheets.codalab.org/worksheets/0xe7c36d3ef3e9470aafdd4ee8c6bdb381
New Wiki	https://worksheets.codalab.org/worksheets/0x85b5fba9828d40e29cb38162481b9067
Adversarial	https://worksheets.codalab.org/worksheets/0xb55d6c8c77ca482fab513b878b37a9c3

Delphi	https://worksheets.codalab.org/worksheets/0x6faa1a13051a46cf97ee92edefeea17f
All results	https://squad-model-evals.s3-us-west-2.amazonaws.com/model_db.json
MRQA Evaluation for Distribution shift	https://worksheets.codalab.org/worksheets/0xfe9a7590c75145498c0895b24c1e5fee