Explore Different Transfer Learning Methods for More Robust Q&A Models

Stanford CS224N Default Project
Track: RobustQA

Yu Shen Lu
Department of Computer Science
Stanford University
yushenlu@stanford.edu

Dingyi Pan
Symbolic Systems Program
Stanford University
dpan3@stanford.edu

Abstract

When given a large amount of data, Natural Language Processing (NLP) systems that are fine-tuned on pretrained language models are able to achieve good performance in Question Answering (QA) task. Yet, these systems cannot generalized well to datasets from unseen domains. Kumar et al. [1] introduce the Feature Distortion Theory, which attempts to explain the poor performance of the fully fine-tuned pretrained model on out-of-domain image classification tasks. The theory suggests that complete fine-tuning distorts the pretrained features. In this project, we test the theory in NLP domain for QA task. Using the pretrained DistilBERT model and applying different partial fine-tuning strategies before fine-tuning the full model, we find that partial fine-tuning does not significantly improve the performance. In addition, to enhance the robustness of the QA system, we also use other out-of-domain adaptation and few-shots learning methods, including Data Augmentation and Mixture of Experts. The best model achieves **F1 = 60.49** and **EM = 42.5** on the out-of-domain test set.

1 Key Information to include

- Mentor: Kamil Ali
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

With the introduction of large pretrained language models, many NLP systems are able to improve the performance on many downstream tasks, by using the representations from pretrained models and further fine-tuning on additional data for specific tasks [2]. Transfer learning relaxes the requirement of large-scale well-annotated datasets for complex tasks [3]. In this project we will focus on Question-Answering tasks, where a model is shown a paragraph as the context and an question that queries relevant information. The model is expected to extract the information and predict the span of text within the context that answers the question concisely. Most approaches for the QA tasks assume that the test data are independently and identical distributed as the training data. Thus, current QA models trained on one types of data cannot be well generalized to unseen domains. The main goal of this project is to build a QA system that is robust to the domain shift.

As an attempt to explain the tradeoff between the in-domain and out-of-domain performance in fine-tuning, Kumer et al. [1] propose the feature distortion theory. Specifically, fine-tuning will distort the pretrained features, as it tries to fit the pretained features onto a randomly initialized new head. Hence, fully fine-tuning a pretrained model improves the performance on in-domain tasks but cannot

generalize will to out-of-domain tasks. On the other hand, tuning only the head while freezing the lower layers preserves the pretrained features, which leads to better performance in out-of-domain tasks. Testing on image classification in computer vision, they show that combining fine-tuning with a method that only tunes part of the model parameters leads to better performance in out-of-domain tasks. Hence, the additional goal of this project is to test the Feature Distory Theory in the NLP domain, specifically on the QA task.

Moreover, there are many strategies that improve the robustness of the model. Recent study also shows that the top layers of a pretrained model are more closely related to the pretrained task and will modify significantly during fine-tuning, whereas the intermediate layers are closer to the representation of the linguistic features and thus more transferable. Hence, re-initializing the top layer can lead to a better performance in transfer learning [4]. In addition, Mixture-of-Exerts and data augmentation are two addition methods that are commonly used in fewshot adaptation.

In this project, we first verify whether the hybrid strategy of combining complete and partial fine-tuning. Besides linear probing, which is the method that is used in the original paper that only tunes the random head, we use anothoer partial fine-tuning strategies, which only tunes the bias terms. Additionally, we further improve the model performance by reinitializing the top layer of the pretrained model. In addition to testing the theory, we also use different data augmentation strategies[5] and mixture of experts[6] on the out-of-domain datasets to increase the robustness of the model. We then analyze these effectiveness of these methods and show an improvement in the performance on out-of-domain test data.

3 Related Work

3.1 Transfer Learning

Fine-tuning (FT) and linear probing (LP) are two main approaches to adapt a pretrained model to specific downstream tasks in computer vision. Kumar et al. [1] show that FT models perform better than LP models in in-domain tasks but worse in out-of-domain tasks. On the other hand, linear probing freezes the pretrained features and only tunes the head, which leads to higher accuracy in out-of-domain task. In order to combine the advantages of both approaches, [1] propose the LP-FT model that trains the head with linear probing and then fine-tunes all parameters, and show that it leads to better performance in both in-domain and out-of-domain image classification task.

In addition to LP, in the NLP domain, Zaken et al. [7] show that tuning only the additive bias terms of BERT models (BitFit) can achieve comparable results as training the entire model. In particular, they consider fine-tuning as a way to expose the pretrained language model to a specific task, instead of letting it learn a completely new task. Moreover, this partial tuning strategies works well when the size of the dataset is small.

3.2 Out-of-domain adaptation

Various methods are developed to adapt language models to out-of-domain data with few examples. Zhang et al. [4] empirically shows that few-shot learning benefits from re-initializing pretrained BERT model can help improving the generalization ability of the final model. Since we are interested in generalizing over three different datasets, a natural approach is to have a specialized model that handles different dataset. We took inspiration from another papers from image classification that proposed sparsely gated Mixture of Expert (MoE) models, which a gate function picks the top 1 or 2 models to make final predictions[6]. This reduces the cost of training computation since only part of the expert ensemble is trained at each data point, which makes it possible to create MoE model with few data points.

3.3 Data augmentation

Since there are only few data in the out-of-domain training set, some of the correlations that the model learns may be fragile and it is easy for the model to overfit. A common approach to introduce more data and add noise to the data in low-resource tasks is data augmentation [8]. One popular model-based strategy is to use backtranslation [9], which uses neural machine translation to translate a sequence into a different language and then translate back to the original language. Additionally,

[5] introduce Easy Data Augmentation (EDA) techniques, which consists a set of simple rule-based operations that slightly modify the existing data, by 1) randomly deleting word (RD), 2) inserting a synonym of a word in the sentence at a random location (RI) 3) randomly swapping the location of two word (RS), 4) randomly select words and replace each of them with one of its synonym (SR). The results in text classification tasks show substantial improvement especially when the size of the dataset is small.

4 Approach

Base model Our baseline model is a DistilBERT [10] that is fine-tuned with all the in-domain training datasets and evaluated on the out-of-domain validation sets. Our model input is Our loss function is the sum of the negative log-likelihood (cross-entropy) loss for the start and end location of the answering text.

In partial fine-tuning, we add $l2$ regularization term in the loss function. λ is our regularization factor, and w is the weight of all trainable parameters in our model. Our final loss function is shown in equation 1, where i and j are the true start and end position of the answer, and p_s and p_e are the predicted logits of the start and end position, respectively:

$$loss = -logp_s(i) - logp_e(j) - \lambda \sum |w|^2 \tag{1}$$

We first implement LP [1] which freezes all the pretained DistilBERT layers and only trains on the randomly initialized head for the QA task. To align our approach with the original paper, we test it with additional l2 regularization. Additionally, we adopt BitFit [7], which freezes the most of the encoder parameters and only fine-tunes the additive bias terms of the pretrained BERT model. In order to obtain the optimal learning and weight decay value, we use Bayesian optimization at this hyper-parameter search step. Additionally, since the higher layers are more related to the task of the pre-trained model, we also reinitialize the top layer of the BitFit model to improve its ability to generalize to new tasks. We then compare the performance of these hybrid models with the performance of the baseline on both the in-domain and out-of-domain validation sets. The best model structure is chosen as the expert for the Mixture of Experts step.

Data Augmentation For each sample in the out-of-domain datasets, we implement five different strategies to augment the context paragraph, four at the token level and one at the sentence level. At the token level, we adapt EDA [5] techniques to each sentence in the context paragraph. Then, we discard the generated paragraphs that do not contain the original answers, and each strategy results in four times more data than the original. Furthermore, we also augment the data at the sentence level by inserting a sentence at a random position between sentences. To ensure that the model does not simply memorize patterns from the in-domain sets, each of inserted sentences is randomly selected from a set of target sentences (i.e. sentences contain the answer to the question) from the context paragraph in the in-domain datasets. Each method produces approximately four times more data. We test the effect of each augmentation method and use the combination of them for the final training.

Mixture of Experts Inspired by the sparsely gated mixture of expert models, we have a gate model that assigns the query to best expert model base on the query content. This is a slightly different approach to the original paper since we explicitly train our expert models with one out-of-domain dataset each rather than training all expert models together. Our gating model (Expert Selector) is also trained separately from the expert models. The full pipeline can be seen in Figure 1.

We implement our own Expert Selector model by fine-tuning a DistilBERT model for text classification. We use data from the three datasets as input and the name of their dataset as labels. The expert selector model then predicts which dataset a query comes from and call the expert model that specialize in that dataset. For each expert model, we trained the best model from the partial fine-tuning part with one of the out-of-domain dataset. This way, decoupled the training of expert models and gate model to save computing resources. Each expert model is fine-tuned on one of the three out-of-domain dataset.

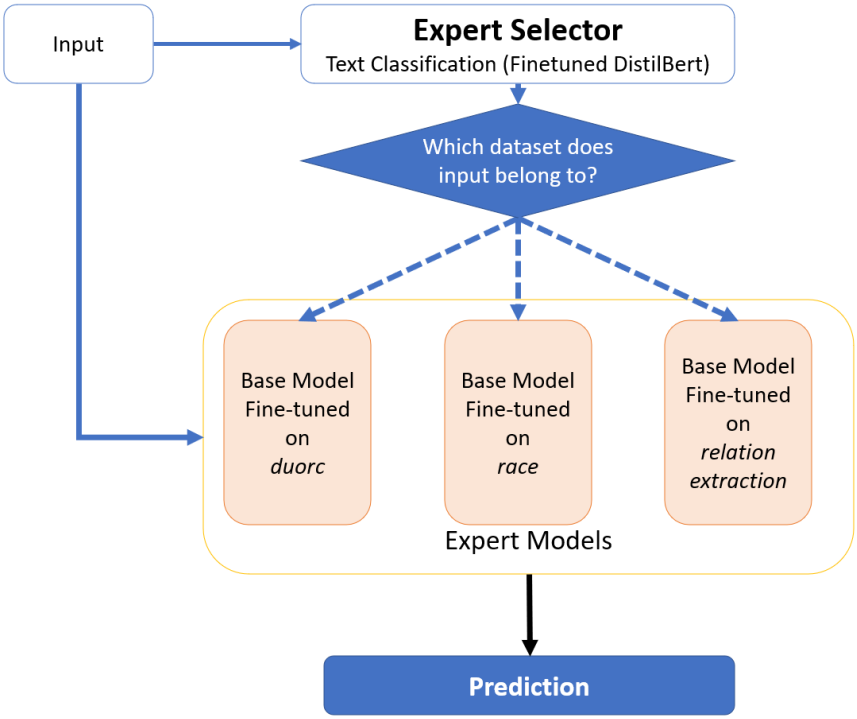

Figure 1: Final model pipeline

5 Experiments

5.1 Data

We use the provided datasets, which include three in-domain reading comprehension datasets (SQuAD [11], NewsQA [12], Natural Questions [13]) and three out-of-domain datasets (DuoRC [14], RACE [15], RelationExtraction [16]).

	In-domain			Out-of-domain		
	SQuAD	NewsQA	Natural Questions	DuoRC	RACE	Relation Extraction
Train	50,000	50,000	50,000	127	127	127
Aug*	-	-	-	2082	1952	2035
Dev	10,507	4,212	12,836	126	128	128
Test	-	-	-	1248	419	2693

Table 1: Statistics for datasets used in this project. The augmented dataset is only used when fine-tuning the model for MoE.

5.2 Evaluation method

We use F1 and Exact Match (EM) as the evaluation metrics to measure the performance of the model. F1 is the harmonic mean of precision and recall with a maximum value of 100. Exact match is a binary measure to indicate the answer produced by the model is exactly the same as the ground truth, maximum value of EM is also 100.

To quantify feature distortion during the training process, we also measured the average change in weights in the pretrained model (excluding the randomly initialized head) in each trained model. The average change in weights is computed as follows, where DB stand for DistilBert model, and DB'

is the model after transfer learning for the question answering task with in-domain data. $W^{(i)}$ is the weight matrix of the i-th layer, and $b^{(i)}$ is the bias term of the i-th layer.

$$\Delta W = \frac{\sum_{i=\{1,2,3,4,5\}} |W_{DB}^{(i)} - W_{DB'}^{(i)}| + |b_{DB}^{(i)} - b_{DB'}^{(i)}|}{\sum \#Parameters_{DB}} \tag{2}$$

5.3 Experimental details

Bayesian Optimization: We modify the training loop and the trainer process to incorporate Bayesian optimization from ax-platform library to find the best learning rate and regularization factor. In general, we see that there are some hyper-parameter settings that will produce extremely poor behavior, but there are not much difference in performance for "good" hyper-parameters. That is, when we graph the fitted function from Bayesian optimization, the local maximums are very similar in height and fairly flat. We show one such plot from the Bayesian optimization experiment when fine-tuning our model on the relation extraction dataset in Figure 2.

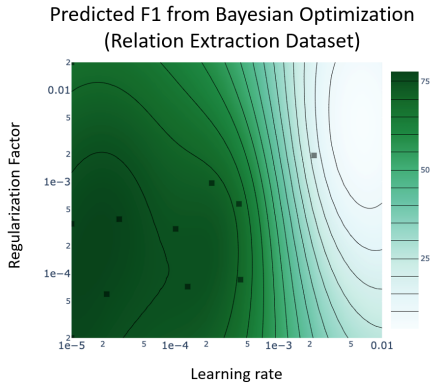

Figure 2: Predicted F1 score from 30 Bayesian optimization trials

Training Configuration and Training Time: We use batch size of 16 for all training processes, all QA models are trained for 3 epochs. For in-domain datasets, the training time of all LP models are around 1.5 hours each, all BitFit models have training time of 2.25 hours, the fine-tuning steps takes on average 3 hours. For out-of-domain datasets, our model is evaluated more frequently due to the limited data, the overall training time for models vary from 1.5 minutes to around 5 minutes depending how much augmented data is used. We run 30 rounds of training in Bayesian optimization, so each experiment takes around 45 minutes to 3 hours. Finally, our expert selector model takes less than 1 minute to train with batch size 8 and learning rate 3e-5, 1 epoch is enough for it to reach 100% accuracy in the training set. The configuration for each expert model is reported in Table 6 in Appendix.

5.4 Results

According to the results in Table 2, using LP before the complete fine-tuning (LP-FT with regularization) has the best performance on the in-domain validation set, but the performance is not better than the baseline in out-of-domain. On the other hand, applying BitFit with reinitialization before fine-tuning (BitFit Reinit FT) has the best performance on the out-of-domain development set. In comparison to the vanilla BitFit model, BitFit Reinit has better performance, which suggests that reinitialization indeed helps with generalizing the pre-trained language model to new tasks.

Yet, although BitFit Reinit is better than LP with regularization as it modifies slightly more parameters, it still does not achieve comparable performance as the complete fine-tuning. The main reason why this hybrid model does not work as expected is because the partial training does not yield good initialization. When partial training performs very poorly, the model performance still largely relies

on the fine-tuning process, such that the head gets changed significantly, eliminating any effect of initialization by the partial training.

Model	In-domain		Out-of-domain	
	F1	EM	F1	EM
FT (Baseline)	70.86	54.62	47.72	30.63
LP with regularization	19.61	10.86	11.45	03.93
LP-FT with regularization	**71.36**	**54.99**	46.26	32.20
BitFit	52.74	36.82	35.28	19.37
BitFit Reinit	63.85	47.40	42.39	25.39
BitFit Reinit FT	70.83	54.32	**47.73**	**33.25**

Table 2: F1 and EM score of the performance on each model in in-domain and out-of-domain evaluation datasets.

Additionally, we test the BitFit Reinit FT model on the original and the augmented datasets, and the result shows that data augmentation slightly improves the performance. Similarly, using MoE with original data also marginally improves the results, and the best model uses MoE to select the best model that is fine-tuned on the augmented dataset (F1 = 50.79, improves the baseline by 4.18%, F = 36.65, improves the baseline by 5.26%). The improvement is as expected, since data augmentation not only increases the number of data but also adds randomness to prevent overfitting. Additionally, since the Expert Selector is fairly accurate in classifying the text, it further boosts the performance.

Model	Validation		Test	
	F1	EM	F1	EM
Baseline	48.75	34.82	–	–
Baseline + Data Augmentation	49.27	35.08	–	–
MoE	49.62	34.82	–	–
MoE + Data Augmentation	**50.79**	**36.65**	60.49	42.50

Table 3: F1 and EM score of the model performance on out-of-domain datasets.

6 Analysis

6.1 Feature Distortion Theory

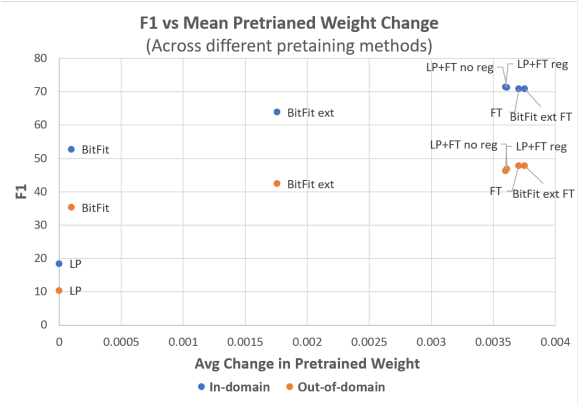

Figure 3: Performance of model vs change in average weight accross different training methods

We evaluate the feature distortion theory using different partial training methods when training the baseline model on the in-domain datasets. We plotted the performance (F1) of a model against its change in pretrained weight, as shown in Figure 3. Feature distortion theory highlights that the fine-tuning process overfits the features to the randomly initialized head, so partial-pretraining would train the head to adapt to the features first in order to minimized the change in pretrained weights in

the final fine-tuning process. It is suggested that this approach might improve a model's performance on out-of-domain data. We notice that partial training does decrease the change in pretrained weights significantly. However, we also observe that some distortion is required to achieve good performance. In fact, all the models with good performance requires Fine-tuning, and they all have similar change in weights. One possible reason is that partial training does not yield a good enough initialization, and the model performance still largely rely on the later fine-tuning process. another possible reason for this is that the features extracted by the pretrained model is overfitted to the text generation tasks that DistilBERT is initially trained for and some distortion is required for the features to be useful in QA tasks. In BERT structure, [17] found that early layers of BERT focuses on different question keywords such as 'how' and 'why' in the questions after fine-tuning for Question-Answering tasks. This means that fine-tuning all the early layers are necessary for BERT to perform well in QA tasks. If DistilBERT has the same property, this might be able to explain why changing pretrained weights is necessary. **We conclude that combining partial training and fine-tuning, which is proposed based on feature distortion theory, does not improve our model in this task**.

6.2 Data Augmentation

As shown in Table 3 suggests, data augmentation improves the result, especially when combined with the MoE. The five types of methods, namely RD, RI, RS, SR at the token level and Sentence RI, all improve the prediction, and among them, SR has the best performance on the validation set. These methods slightly modify the meaning and grammaticality of the sentence or the logical flow of the paragraph. For instance, in one example augmented by SR, 'fiction' is replaced with its synonym 'fabrication' in the compound noun 'science fiction.' Thus, the result seems to suggest that addition noise and randomness to the data can improve the robustness of the model. While this may be counter-intuitive, since the meaning of the sentence is crucial to reading comprehension and QA tasks, it will be interesting to compare these methods with other data meaning preserving augmentation techniques, such as backtranslation, in future experiment.

6.3 Mixture of Experts Effectiveness

The expert selector model has 100% accuracy on the training data to pick the correct expert model to query, and it is 99.01% accurate for the validation set. This means that the three out-of-domain data-sets are easily separable. As a result, we see improvement of the performance of our model with MoE structure, since our model can reliably pick the correct expert model which is trained on the corresponding dataset. However, the overall performance is still bounded by the performance of the expert models (Table 6 in Appendix).

6.4 Types of Mistakes

Error Type	Question	Context
Negation Identification	Where should you go to wash your car when you are in Moscow?	[...] In Moscow, if your car is dirty enough to draw dust art, you will be fined about 2,000 rubles. Worse yet, it's not legal to wash your car by hand *in public places*–forcing you to take it to one of the few **car wash facilities**. [...]
Multiple correct answers	Who was the brother of Peter Miller Cunningham?	[...] Peter Miller Cunningham was the fifth son of John Cunningham, land steward and farmer (1743–1800), and brother of *Thomas Mounsey Cunningham* (1776–1834) and of **Allan Cunningham** (1784–1842).
Wide Answer Span	When is the best time to visit Stonehenge according to the passage?	[...] As the weather can be pretty bleak in winter and the crowds huge in *summer, we suggest **autumn*** should be the best time to visit these monster rocks. [...]

Table 4: Examples of different types of mistakes in model predictions. The ground truth answer is in bold, and the predicted answer is in italics.

We see that our model performs worse on datasets with long and complex contexts such as those in the race dataset. We listed some common mistakes that our model makes in the validation data in table 4. We observe our model has problem understanding uncommon negation and having trouble narrowing down the answer span. However, there are also cases where there can be multiple instances/choices of correct answers and our model simply does not agree with the option given in the validation set.

6.5 Final Performance

The test and validation performance are similar for our model. However, we see that our model have a much higher F1 and EM score on the final test set, which, as of the writing of this report (March 12th, 2022), rank 2 in EM and 2 in F1 among the 27 test submissions. We see that test set is a lot more skewed than the validation and training set. The expert selector is extremely effective on discerning the source dataset of the queries, so we believe that our expert select is fairly reliable and the final test set is in fact a skewed data set which favors the DuoRc and relation extraction models, which both have better performance than our model trained on race dataset. This explains the higher final test performance compare to the balanced training and validation sets.

7 Conclusion

In this project, we first tested feature distortion theory in our question answering task. We found that feature distortion theory does not apply in our case. Partial training doesn't yield a good enough initialization, and the model performance still largely rely on the later fine-tuning process. We observe that some distortion is required to achieve good performance. We wonder if this is characteristic of model structure since there are a lot of weight sharing between tokens, or this may be because the pretrained model is for a different task.

To improve our model's performance on QA tasks, we then tried two main strategies: data augmentation and sparse Mixture-of-Experts.

In data augmentation, we note that since there are very few out-of-domain training data, randomly changing the context not only increases the number of data but also adds noise to it, which leads to a slightly better performance.

Our version of Mixture-of-Experts picks the best model to answer each question, but due to the limitation on the performance of each expert model since we have limited out-of-domain data, it only improves the overall performance slightly. However, in test settings where there are skewed distribution of out-of-domain data, our model can perform better since it can reliably find the source dataset of the queries and pick the best expert model.

References

[1] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning distorts pretrained features and underperforms out-of-distribution. In *International Conference on Learning Representations*, 2022.

[2] D. Jurafsky and J.H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2021.

[3] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham, 2018. Springer International Publishing.

[4] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav. Artzi. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*, 2019.

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

[6] Carlos Riquelme Ruiz, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of

experts. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[7] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models, 2021.

[8] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp, 2021.

[9] 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.

[10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

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

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

[13] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

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

[15] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

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

[17] Sahana Ramnath, Preksha Nema, Deep Sahni, and Mitesh M Khapra. Towards interpreting bert for reading comprehension based qa. *arXiv preprint arXiv:2010.08983*, 2020.

A Appendix

Model	**Training**		**Validation**	
	F1	EM	F1	EM
Baseline	46.60	32.20	48.75	34.82
RD	47.83	33.51	49.89	**35.86**
RI	48.14	32.98	50.26	34.82
RS	**48.48**	**33.77**	50.00	35.08
SR	47.43	33.51	**50.36**	35.60
Sent_RI	48.09	33.51	49.13	33.77

Table 5: F1 and EM score of the performance using different data augmentation method.

Expert	**Learning Rate**	**Weight Decay Rate**	**F1**
DuoRC	1e-05	0.00217	40.86
RACE	1e-05	0.00149	36.13
RelationExtraction	2.887e-05	0.00039	77.13

Table 6: Model configuration for each expert model. F1 is the best F1 score when trained on the specific dataset.