Exploring Mixture of Experts, In-context Learning and Data Augmentation for Building Robust QA

Stanford CS224N Default Project (Robust QA track)

Chen Chen
Stanford University
inquisit@stanford.edu

Hanzhao Lin
Stanford University
hanzhao@stanford.edu

Shan Lu
Stanford University
shanlu33@stanford.edu

Abstract

The task of learning from a limited number of examples (few-shot learning) is valuable because it uses only small amounts of data for prediction, making the models applicable in practical and realistic situations. In this work, we aim to train and fine-tune a robust QA model that can generalize well on out-of-domain datasets in few-shot settings. Our team explored mixture-of-experts (MoE), in-context learning, data augmentation, complemented by hyperparameter tuning to improve the domain adaptation robustness of DistilBERT-based[1] question answering systems. We significantly improved the baseline with a fine-tuned MoE model, while ending up with no metrics improvement with other approaches. The performance of in-context learning was subpar compared to the baseline, and data augmentation techniques were not able to generalize well in our experiments. Quantitatively, the final submission of our best approach, a fine-tuned MoE network without other techniques, **achieved 43.876 EM and 61.933 F1 score and was ranked 4th out of 56 submissions** in the test set.

1 Key Information to include

- Mentor: Kendrick
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

In the tasks of question answering (QA), language models are given input pairs each consisting of a context and a question related to the paragraph. Models are trained to extract a text span that answers the question from the context. Recently, the advances in large-scale pre-trained language models including BERT[2] has resulted in a huge performance boost on all kinds of natural language processing (NLP) tasks. With fine-tuning, the insights from pre-trained language models can be transferred to QA tasks. In certain datasets such as SQuAD, recent work has even produced systems that surpass humand-level performance in certain metrics[3]. However, while humans can easily generalize learnings in related tasks, language models often struggle to transfer the knowledge between datasets. Models trained on in-domain datasets typically need to be fine-tuned on thousands of out-of-domain examples to perform well. In few-shot settings, when only limited numbers of examples are available, the performance degrades significantly for such systems.

In this paper, we explore several techniques to make our DistilBERT-based QA models more robust on out-of-domain datasets in few-shot settings. Specifically, we implemented a simplified mixture-of-experts (MoE) model, where a gating network classifies the input sequence according to its data source, and forwards the input to the specialized domain expert. We also explored prompt-based in-context learning techniques, adding a querying prompt and mask tokens in between a question and a context to align the training objective (MLM) with the pre-trained framework. Finally, we evaluated these approaches with and without data augmentation techniques and hyperparameters tuning.

3 Related Work

3.1 Mixture-of-Experts (MoE)

Mixture-of-experts was first introduced by Jacobs et al.[4] where separate networks learn to handle a subset of the complete set of training cases, which can be viewed as either a modular version of a multi-layer supervised network, or as an associative version of competitive learning. Since then, various applications and improvements of MoE have been proposed. For example, Shazeer et al.[5] introduced a sparsely-gated mixture-of-experts layer, a trainable gating network determining a sparse combination of the experts to use for each example. Another example is conditional computation where parts of the network are active on a per-example basis. We also got inspiration from Eigen et al.[6], where multiple sets of gating and experts were used to tackle sub-problems.

3.2 In-context Learning

Large pre-trained models such as GTP-3 are known to display "in-context" learning characteristics[7][8][9], in which NLP tasks can be formulated into masked language modeling tasks with a querying prompt (prompting: Figure 4). Additional demonstrative text sequences with the same structure can be appended to the querying sentence to "show" how related problems should be solved[10] (demonstration). Both prompting and demonstration can be used at training or inference time, giving in-context language models more contextual evidence to generate the correct answer. While classification problems have been shown to work "in-context" on smaller pre-trained models such as BERT[11], our paper explores in-context QA models which are more challenging.

3.3 Data Augmentation

Data augmentation techniques are commonly used in NLP tasks to generate additional and synthetic data in response to a lack of annotated training examples[12][13]. The effectiveness of context paraphrases on improving the performance and robustness of QA systems has been investigated[14][15]. The studies claim that back-translation significantly improved the F1/EM scores.

4 Approach

Our QA system is built with 3 main building blocks as shown in the Figure 1, namely, a DistilBERT-based mixture-of-experts gating network, in-context learning components, and data augmentation. The system is set up to conduct experiments on flexible combinations of individual components. We implemented the mixture-of-experts classifier based on `DistilBertForSequenceClassification`[1], heavily modified `FewShotDataset`[2] and `DistilBertForMaskedLM`[3] for in-context learning on QA datasets, and integrated `nlpaug`[4] into the system for data augmentation capabilities.

4.1 Baseline

The DistilBERT baseline model, as shown in Figure 2, `DistilBertForQuestionAnswering`[5], trains a classification head responsible for classifying which indices from the context have the best chance of being the start or end of the answer to the question. The model takes a question-context pair as sequence input, whose generated hidden states are then consumed by a classification head. The loss function is the summation of cross entropy loss between golden indices and predicted ones.

[1] `https://huggingface.co/docs/transformers/model_doc/distilbert#transformers`

[2] `https://github.com/princeton-nlp/LM-BFF/blob/main/src/dataset.py`

[3] `https://huggingface.co/docs/transformers/model_doc/distilbert#transformers`

[4] `https://github.com/makcedward/nlpaug`

[5] `https://huggingface.co/docs/transformers/model_doc/distilbert#transformers`

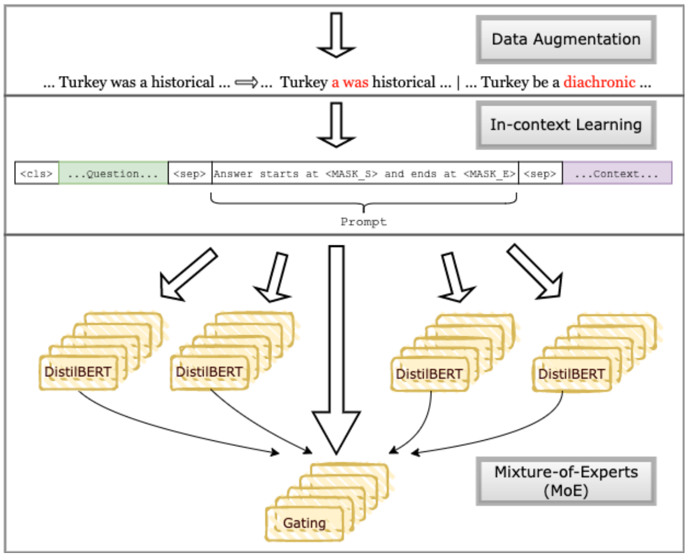


Figure 1: System Architecture

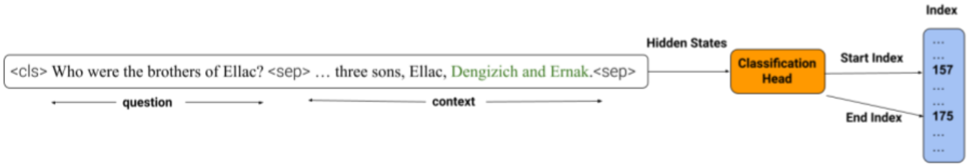

Index
Hidden States
<cls> Who were the brothers of Ellac? <sep> ... three sons, Ellac, Dengizich and Ernak.<sep>
question
context
Classification Head
Start Index
End Index
157
175

Figure 2: DistilBERT with classification head for QA (baseline)

4.2 Mixture of Experts

In the mixture-of-experts network[16], multiple DistilBERT QA model instances are fine-tuned corresponding to every individual out-of-domain dataset. Additionally, a top-level gating network is trained for classifying the input source and forwarding the input to the potential domain experts. The final model output y is taken as the weighted average of outputs of domain experts, which could be formulated as

$$y = \sum_i g_i f_i$$

where f_i is the output of input x evaluated on expert i's DistilBERT QA model and g_i is the mixture weight for this expert i produced by the gating function applied to input x.

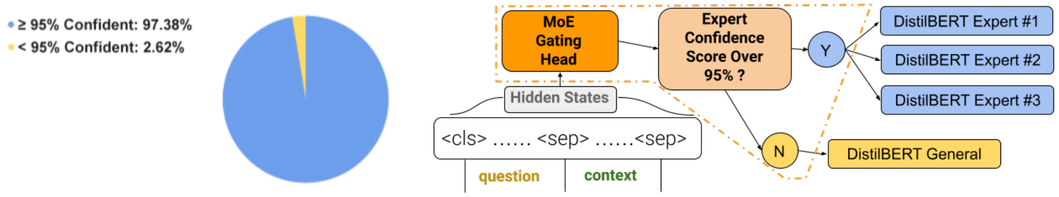
We chose `DistilBertForSequenceClassification` as the core part of our gating network. And after observing that it was quite easy for a DistilBERT-based classifier to achieve a very high classification precision in these 3 datasets, we simplified this approach by removing the averaging layer. Complementarily, we introduced another "generalist" model, which was fine-tuned on all out-of-domain datasets, to make prediction when the classifier got confused. In other words, the final output of our mixture-of-experts network is purely based on one domain expert model or the generalist model. The predicted domain expert model will determine the final output when it earned a high-enough confidence score from the classifier, otherwise the generalist model will take it over.

We also performed analysis on the confidence score distribution to figure the right confidence threshold here, as shown in 3a. In validation set, the gating network produced a confidence score less than 95% for only 10 out of 382 test samples, and all errors fell into this bucket.

With all the modifications mentioned above, the output y of our final version model could be formulated as

$$y = \begin{cases} f_i & \text{if } g_i \geq 0.95 \\ f_{generalist} & \text{otherwise} \end{cases}$$

where $f_{generalist}$ is the output from the generalist model. And the whole network architecture could be visualized as 3b.



(a) Confidence score distribution

(b) Mixture-of-experts model architecture

Figure 3: Mixture-of-experts

4.3 In-context Learning

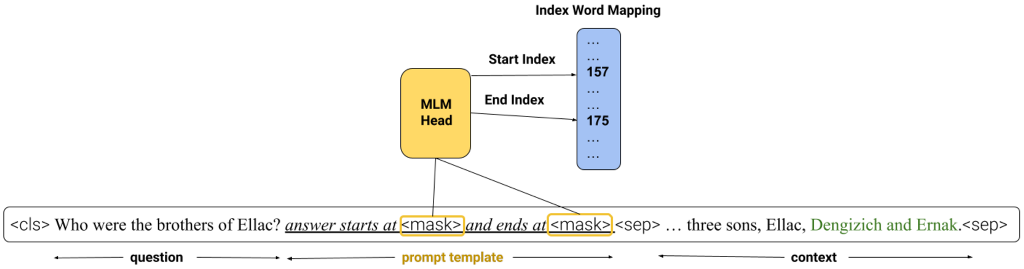

Index Word Mapping
Start Index
End Index
MLM Head
...
...
157
...
...
175
...
...
<cls> Who were the brothers of Ellac? answer starts at <mask> and ends at <mask> <sep> ... three sons, Ellac, Dengizich and Ernak.<sep>
question
prompt template
context

Figure 4: In-context learning for QA

Our in-context implementation for QA system (in-context QA) is inspired by LM-BFF[11] and `DistilBertForQuestionAnswering`. Nonetheless, many adaptations are required when applying in-context learning to QA datasets. For instance, instead of a classification head in the default model, our approach uses a DistilBERT MLM head to generate start and end indices at two `<mask>` locations of a querying prompt in an "in-context" manner, as shown in Figure 4. Moreover, in order for certain generated words to convey unambiguous meanings of labels, a Label Word Mapping between MLM generated words and the true label is required. Unlike LM-BFF's classification mapping which contains only a few catagories, our in-context QA model is capable of generating indices in output language. This is achieved by using a much larger category mapping that covers all 512 available locations from input sequences.

Engineering-wise, in-context QA uses an innovative two-pass tokenization process to construct in-context `input_ids` before feeding examples to DistilBERT. The first pass comprises of the standard sentence pair tonkenization specifically supported by `DistilBertTokenizer` for QA tasks. It is in this pass that long contexts can be broken down into suitable sequence length that is in compliance with DistilBERT's maximum sequence length. In the second pass, the prompt template crucial for in-context learning is tokenized and injected.

The advantage of this design is that demonstration, a technique that combines multiple examples into 1 input sequence, can be easily supported. Although we did not get to experiment with the demonstration setup due to the nature of QA tasks and the sequence length limitation of DistilBERT. QA tasks generally requires more context for answers to be identified. We are finding it challenging to meaningfully fit more than 1 QA pair in the 512 maximum supported sequence token length[6].

In terms of loss function, the choice of framing QA as a language modeling task allows us to use the standard encoder-decoder objective that maximizes the log likelihood of the text in the ground truth target from the output of the model, specifically at the masked indices. Formally,

$$L(\theta) = -log\mathbb{P}(y_{m_s}|x;\theta) - log\mathbb{P}(y_{m_e}|x;\theta)$$

[6] `https://github.com/huggingface/transformers/issues/1791`

4.4 Data Augmentation

We employed a few selected data augmentation techniques provided in `nlpaug`[17] in our system, based on their past success and adaptability to QA datasets. In order for these techniques to not disrupt the question answering coherence while still providing similar but different contextual reinforcement from original datasets, the augmentation will only be performed in the context text with the sentence including answer phrases fully preserved. Examples could be found in Appendix A.1.

The data augmentation techniques we selected include

- **Back Translation**: Translate the context text from English to an intermediate language, e.g., German, and then back to English using Facebook WMT19 models[7].
- **Random Swap (RS)**: Randomly swap a word with its siblings in the same sentence.
- **Synonyms Replacement (SR)**: Replace some words with their synonyms.

4.5 Hyperparameter Tuning

Besides the major components mentioned above, we also identified two hyperparameter-tuning tricks from our analysis on data and model, which led significant performance boost.

4.5.1 Reduced Output Length

After inspecting model outputs and error pattern, we found the model is likely to produce over-detailed predictions like the example in Appendix A.2. The results make sense to the question, but they hurt both EM and F1 scores. Given the length of answers in out-of-domain datasets are relatively shorter and no answers are longer than 8 words, we reduced the max length of model outputs from 30 words (default setup) to 9 words to reduce such redundancy.

4.5.2 Number of Frozen DistilBERT Layers when Fine-tuning

We observed that the first a few layers of DistilBERT carry important lower-level language features, which could be transferred really well between datasets. Motivated by this finding, we conducted experiments to figure out the best amount of layers to freeze. It turned out freezing embedding layers and first 4 layers of transformers block when fine-tuning the model worked best.

5 Experiments

5.1 Data

The datasets we use are provided by the CS224N teaching staff. Each input consists of a context-question pair and answer to each question is guaranteed to be located within the context. The datasets are divided into two categories: in-domain datasets out-of-domain datasets. The amount of data samples in each dataset is shown in Table 1. The in-domain datasets all contain 50,000 training examples, while out-of-domain datasets only have 127 training examples individually, which made the knowledge transfer between datasets very critical.

5.2 Evaluation method

To analyze the performance of our model, we used the evaluation method provided by the CS224N staff. The best performing models are determined by their exact match (EM) and F1 scores of QA span predictions. The EM score is a binary measure of whether the system output matches the ground truth answer exactly, and the F1 score is the harmonic mean of token-level precision and recall of predictions. The models were first trained on training sets, and then evaluated on the out-of-domain dev sets. The model with best performance in dev sets is is selected for benchmarking on test sets.

[7] `https://huggingface.co/facebook/wmt19-en-de`

Dataset	**Question**	**Context**	**Train**	**Dev**	**Test**
In-domain Datasets					
SQuAD[18]	Crowdsourced	Wikipedia	50,000	10,507	-
NewsQA[19]	Crowdsourced	News articles	50,000	4,212	-
Natural Questions[20]	Search logs	Wikipedia	50,000	12,836	-
Out-of-domain Datasets					
DuoRC[21]	Crowdsourced	Movie reviews	127	126	1248
RACE[22]	Teachers	Examinations	127	128	419
RelationExtraction[23]	Synthetic	Wikipedia	127	128	2693

Table 1: Data sources and splits

Model name	**EM/F1 (Dev)**	**EM/F1 (Test)**
Baseline	34.55/50.28	-
Back Translation (BT)	32.46/47.45	-
Random Swap (RS)	35.08/50.11	-
Synonyms Replacement (SR)	34.03/49.21	-
Reduced Output Length (ROL)	34.82/50.92	-
Mixture of Experts	37.17/52.44	42.94/59.82
Mixture of Experts (ROL)	38.73/54.19	43.85/61.90
Mixture of Experts (RS, ROL)	38.22/54.27	-
Mixture of Experts (SR, ROL)	**39.27**/53.75	-
Mixture of Experts (with Generalist, ROL)	39.01/**55.13**	**43.88/61.93**
In-context Learning	17.28/39.94	-
In-context Learning (SR)	18.48/38.79	-

Table 2: Experimental results on validation and test sets

5.3 Experimental details

The baseline model was trained on all in-domain datasets and then fine-tuned on all out-of-domain datasets. Random-swap augmentation was configured to randomly swap nearby words in the same sentence at max 10 times. Synonyms replacement augmentation was configured to randomly replace words at max 10 times. Reduced output length (ROL) reconfigured model to output predictions not longer than 9 words. Mixture-of-experts classifier was trained in in-domain datasets individually with precision as its north-star metric during evaluation. Generalist models were directly taken from the baseline models, and domain expert models were fine-tuned individually on its corresponding out-of-domain dataset. In-context QA models were configured to run with or without Synonyms Replacement to evaluate if any gain can be achieved with data augmentation.

All models were trained with learning rate 3×10^{-5} and up to 10 epochs. After finishing an entire epoch of training, the checkpoint is evaluated in the dev set to obtain its metrics. The checkpoint with highest F1 score on dev set will be saved as the final model when training the QA model, and the checkpoint with highest precision on dev set will be saved when training the classifier in mixture-of-experts gating network.

5.4 Results

By assembling multiple approaches and optimization techniques, we obtained performance metrics of 12 variants of our system, as shown in Table 2.

5.4.1 Mixture-of-experts Models

The performance metrics of all expert models and generalist models on individual out-of-domain dataset show in Table 3. As we expected, all expert models outperformed the generalist model on their specialized domain. By being trained individually, DuoRC expert models showed biggest performance

Model name	EM/F1 (Dev)	EM/F1 (DuoRC)	EM/F1 (RACE)	EM/F1 (RE)
Generalist	34.55/50.28	30.16/40.84	18.75/35.26	54.69/74.60
DuoRC Expert	-	35.71/46.42	-	-
RACE Expert	-	-	21.09/36.06	-
RE Expert	-	-	-	54.69/74.76
Models with Random Swap (RS)				
Generalist (RS)	35.08/50.11	28.57/38.34	19.53/35.54	57.03/76.27
DuoRC Expert (RS)	-	35.71/45.07	-	-
RACE Expert (RS)	-	-	22.66/37.82	-
RE Expert (RS)	-	-	-	59.38/78.24
Models with Synonyms Replacement (SR)				
Generalist (SR)	34.03/49.21	27.78/37.66	17.19/33.18	57.03/76.62
DuoRC Expert (SR)	-	37.30/46.64	-	-
RACE Expert (SR)	-	-	18.75/37.55	-
RE Expert (SR)	-	-	-	58.59/78.53
Models with Reduced Output Length (ROL)				
Generalist (ROL)	34.82/50.92	33.33/44.34	21.09/36.46	50.00/71.85
DuoRC Expert (ROL)	-	37.30/47.16	-	-
RACE Expert (ROL)	-	-	22.66/38.34	-
RE Expert (ROL)	-	-	-	60.94/78.64

Table 3: Performance of expert models and generalist models on dev sets

Model name	Precision (Dev)	Precision (Test)
Mixture-of-Experts Classifier	98.43%	99.45%

Table 4: Performance of the mixture-of-expert data domain classifier on dev and test sets

boost with EM+6.55/F1+6.03 on average. And RACE expert models gained EM+2.15/F1+2.33 boost on average. RelationExtraction expert models gained subtle improvements without reduced output length trick (ROL) applied, but with ROL, the expert model got significant improvement of EM+10.94/F1+6.8 by being trained individually, which is very unexpected.

The classifier in mixture-of-expert network, which is used to assign input data to individual expert, has reached very high precision in both dev and test set, as shown in Table 4. In the dev set, it reached 98.43% precision, meaning 376 out of 382 samples were classified correctly. Out of our expectation, the classifier even achieved higher precision of 99.45% in the test set, which indicates that only 24 out of 4360 data samples resulted in wrong expert assignment.

Based on the strong performance of MoE classifier and significant performance gain by individual expert models, the MoE approach gained EM+2.62/F1+2.16 by nature, which was ranked top-10 of submissions ordered by EM score. Identified that most errors of expert assignment were caused by low confidence score, we introduced generalist model activated by low confidence score as the fix, which turned out to work really well in dev set with EM+0.28/F1+1.06. But unfortunately, this fix only earned +0.03 EM/F1 in test set.

5.4.2 Data Augmentation

Surprisingly, data augmentation didn't bring much improvements to our model, instead, its performance effects seemed unpredictable - it could improve EM or F1 score by a small points, but it could also hurt the performance. The randomness introduced by data augmentation greatly hindered us from improving the model performance steadily, and we decided not to try out possible combinations of multiple augmentation techniques.

5.4.3 In-context Learning

For in-context QA model, as we anticipated, it did not perform as well as other configurations possibly due to a downsized model and the lack of training examples that are usually required for better in-context performance. What we did not anticipate, however, was that data augmentation did not help to improve upon the vanilla in-context configuration. We speculate that the amount of augmented data we deployed was simply not enough to make a difference in in-context QA models.

5.4.4 Hyperparameter Tuning

Reducing the length of model output has showed us significant performance improvement by EM+1.56/F1+1.75 from very beginning, which is really impressive. This could be caused by the nature that the ground-truths in all 3 out-of-domain datasets are really short. Finally, by combining the mixture of experts model with this trick, we reached the best scores in the validation set with 39.01 EM score and 55.13 F1 score. And the final submission of this approach was ranked 4th place out of 56 submissions in the test set, with 43.88 EM score and 61.93 F1 score.

6 Analysis

The success of mixture-of-experts was largely due to the skewed distribution of tasks and data in the three out-of-domain datasets. It shows that our models might be fine-tuned to one specific dataset, but not all three of them at once. The high classification precision achieved by MoE classifier also suggested that out-of-domain datasets have very different characteristics in latent space. Also, the test set contains about 2,700 questions from RelationExtraction dataset, while only 1,248 questions from DuoRC and 419 questions from RACE. This data distribution is not aligned with it in training and dev sets, resulting best metrics in dev set cannot be directly translated to the test set performance. Actually from Table 3, it's very intuitive that the RelationExtraction is the easiest one among all three out-of-domain datasets. Given most test samples are from this easiest dataset, optimizing towards the RelationExtraction dataset is the key to obtain a highly-ranked result. This also explains why most teams were able to obtain much better EM and F1 metrics in the test set than in the dev set.

In-context learning did not perform as well as other methods due to various factors. For a start, the lack of demonstration means less contextual correlations were made during training. A good demonstration strategy is required to fit potentially long context into the limited sequence length supported by DistilBERT. Demonstration will increase both training and inference efficiency, which might improve model performance for In-context QA models. Moreover, the current training objective and loss function that solely depend on the correctness of start and end indices can be improved upon. Loss function that capture answer text span and the language distributions of target domains (MLM-based) can be engineered together.

Data augmentation techniques achieved mixed results with unreasonable randomness. We managed to identified a few failure modes in which structural information of answers or/and context has been broken by careless replacement or swapping of key information. Although we tried to mitigate its effect by only augmenting the context text and not the answer text in our datasets, there is still risk in changing the contextual text or dataset characteristics too much. This could be why augmentation techniques that significantly alter the text, e.g., back-translation, generally yield worse results.

7 Conclusion

After trying out many combinations of techniques, we concluded that the simple idea of mixture-of-experts worked very well, since drastically different distribution and quality were observed in target datasets. We also emphasized the importance of tuning hyperparameters based on statistical understanding of the datasets. Even without significant fruits from in-context learning and data augmentation, we conducted very valuable exercises and identified possible difficulties on applying those techniques in reality.

Under the constraint of both time and resources, we were not able to try out more possible combination of the techniques we implemented. As future work, we could conduct more experiments leveraging the composability of our system, and also research towards identifying the root cause of the poor performance of our in-context learning implementation and exploring effectiveness of demonstration.

References

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

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

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

[4] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[5] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. 01 2017.

[6] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. 12 2013.

[7] Nick Ryder Melanie Subbiah Jared Kaplan Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry Amanda Askell et al. Tom B Brown, Benjamin Mann. Language models are few-shot learners, 2020.

[8] Alec Radford, Jong Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 02 2021.

[9] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. pages 255–269, 01 2021.

[10] Tianyu Gao. Prompt-based fine-tuning with demonstrations.

[11] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners, 2021.

[12] Yan xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. Improved relation classification by deep recurrent neural networks with data augmentation. 01 2016.

[13] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. 05 2017.

[14] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. An exploration of data augmentation and sampling techniques for domain-agnostic question answering, 12 2019.

[15] Jasdeep Singh, Bryan McCann, Nitish Keskar, Caiming Xiong, and Richard Socher. Xlda: Cross-lingual data augmentation for natural language inference and question answering, 05 2019.

[16] Robert Jacobs, Michael Jordan, Steven Nowlan, and Geoffrey Hinton. Adaptive mixture of local expert. *Neural Computation*, 3:78–88, 02 1991.

[17] Edward Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019.

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

[19] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset, 2017.

[20] 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. *Transactions of the Association of Computational Linguistics*, 2019.

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

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

[23] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension, 2017.

A Appendix

A.1 Example of Augmented Text

Context: In olden times, England is in turmoil. With the death of the King, noone can decide who is the rightful heir to the throne. With war threatening to tear the country asunder, a stone and anvil appear `from the heavens in London town`, with a sword planted firmly in the anvil. On the hilt of the sword, read the words, "Whoso pulleth out this sword of this stone and anvil is rightwise king, born of England."
Augmented by Back Translation: In **ancient** times, England **was** in turmoil. With the death of the king, no one could decide who **was** the rightful heir to the throne. With war threatening to tear the country asunder, a stone and anvil appear `from the heavens in London town`, with a sword planted firmly in the anvil. **The handle of the sword reads**: "Whoever **draws** this sword **out** of this stone and anvil is a righteous king, born **in** England.
Augmented by Random Swap: In olden times, England is **the in** turmoil. With death of King, the noone can decide who is **rightful the** to heir the throne. With war threatening to tear the country asunder, a stone and anvil appear `from the heavens in London town`, with a sword planted firmly in the anvil. On the **of hilt the read sword, words, the** "Whoso pulleth out **sword this** this of stone and anvil is rightwise **born king,** of England.
Augmented by Synonyms Replacement: In olden times, England is in turmoil. With the death of the King, noone can **settle** who **be** the rightful heir to the throne. With war threatening to tear the country asunder, a stone and anvil appear `from the heavens in London town`, with a sword planted firmly in the anvil. On the hilt of the sword, read the words, "Whoso pulleth out this **blade** of this stone and **incus** be rightwise king, born of England.

* Note that the sentence containing answer `from the heavens in London town` remained untouched during augmentation.

A.2 Error Examples

Question: How big were his artificial bolts?
Context: He produced artificial lightning, with discharges consisting of millions of volts and up to 135 feet long. Thunder from the released energy was heard 15 miles away in Cripple Creek, Colorado. People walking along the street observed sparks jumping between their feet and the ground. Sparks sprang from water line taps when touched. Light bulbs within 100 feet of the lab glowed even when turned off. Horses in a livery stable bolted from their stalls after receiving shocks through their metal shoes. Butterflies were electrified, swirling in circles with blue halos of St. Elmo's fire around their wings.
Answer: `135 feet`
Prediction: `up to 135 feet long.`