Improving Language Model Robustness in the Few-Shot Setting

Stanford CS224N Default Project
Track: RobustQA
TA Mentor: Fenglu Hong
No external collaborators/mentor/sharing project **[3 Slip Days Used]**

Rajas Bansal
Stanford University
rajasb@stanford.edu

Drew Kaul
Stanford University
drewkaul@stanford.edu

Dhruva Bansal
Stanford University
bdhruva@stanford.edu

Abstract

Language models have achieved impressive results over the last few years in a wide range of NLP tasks but can struggle to adapt to out-of-distribution data. In this paper, we investigate the problem of building a QA system which can generalize to unseen domains in the few-shot setting and aim to develop models which improve upon the DistilBERT baseline. We explore several approaches, including data augmentation, masked language modeling, mixture of experts, and fine-tuning and find that backtranslation with MLM and fine-tuning performs best. Our best model achieves an EM score of **46.766** (ranked **1st** on test leaderboard) and an F1 score of **62.977** (ranked **4th** on test leaderboard). Overall, we find that augmenting our out-of-domain dataset via backtranslation improves performance the most, and MLM helps our models further adapt to distribution shift.

1 Introduction

Question answering (QA) is an important task in the field of natural language processing (NLP) and an important benchmark for understanding the capabilities of large language models. Over the last few decades, advances in neural network architectures, larger dataset sizes, and increased computing power have led to significant performance improvements in QA, and these gains have been realized in our day-to-day lives with the proliferation of smart assistants and search engines. Despite these impressive gains, QA systems require extremely large datasets to train, and current methods struggle to generalize when presented with out-of-distribution data. Thus, we decide to explore the QA problem in the low-data regime, where we are given a pretrained language model, a large in-domain dataset, and a small out-of-domain dataset and must finetune it to perform well on the QA task with out-of-domain data. More formally, the model must take in a question and span of context as input and output distributions for the start and end indices of the answer in the context.

To build a robust QA system, we use the DistilBERT model and explore several methods, including backtranslation [1], masked language modeling (MLM), mixture of experts [2], and transfer learning [3] (i.e. finetuning). We implement data augmentation by using backtranslation with multiple pivot languages to generate additional out-of-domain data and use the MLM task to further train our DistilBERT. We also explore training several expert models specialized to different in-domain datasets and learning a mixture of experts model. Finally, we explore different fine-tuning strategies using the out-of-domain data in combination with the above methods.

We ultimately find that backtranslation with MLM and fine-tuning results in a **EM** score of **46.766** (ranked **1st** on test leaderboard) and an **F1** score of **62.977** (ranked **4th** on test leaderboard) on the out-of-domain test dataset.

2 Related Work

2.1 Data Augmentation

Data augmentation is a popular approach to generate more training data from existing data by preserving invariances. One type of data augmentation is word substitution based, in which words in a sentence are replaced by their synonyms. Wei et al. [4] introduce 4 text editing operations to perform data augmentation: synonym replacement, random insertion, random swap, and random deletion. Garg et al. [5] use a slightly different approach, masking tokens in the input and replacing them with the output of a BERT model. Another type of data augmentation involves backtranslation, where a sentence is translated into a pivot language (or sometimes a sequence of multiple pivot languages) before being translated back into the original language. This can be seen in [6], which introduces the QANet architecture and combines it with data generated via backtranslation with a neural machine translation model to achieve state-of-the-art performance on SQuAD.

2.2 Mixture of Experts

Mixture of experts was introduced in [2], which is a procedure to combine the outputs of many different neural networks and can be viewed as a form of competitive learning. Mixture of experts consists of training k expert models and learning a weighting over the experts to produce the final output. Mixture of experts has been applied before in NLP contexts, such as in [7], where a sparse combination of thousands of expert models is learned and achieves higher performance than state-of-the-art on large language modeling benchmarks at a lower computation cost.

2.3 Few-shot learning

There is substantial work in the area of few-shot learning in NLP. In [8], the authors present a set of techniques to improve few-shot finetuning of GPT-3 using a small number of annotated examples. They introduce a novel method for automatic prompt generation for prompt-based fine-tuning and a method for incorporating relevant context, leading to an average of 11% improvement over standard fine-tuning in the low-data regime. [9] also explores few-sample fine-tuning but for BERT. The authors find that debiasing optimization, increasing training time, and re-initializing the top layers speeds up learning and improves performance during fine-tuning.

3 Approach

In this section we first formulate the reading comprehension problem in the few shot setting and then describe the proposed model. We add novel data augmentation methods using backtranslation and a masked language model pretraining for the DistilBert on the out of domain dataset.

3.1 Problem Formulation

We first describe the reading comprehension problem. Given a context paragraph C, and a query question q, we need to predict a span of tokens from C, S which will be the answer to the question. S is completely specified by i_{start} and i_{end}, i.e $S = C[i_{start} : i_{end}]$. We consider two kinds of datasets, the indomain datasets (D_i) which have a lot of data available, and the out of domain datasets (D_o) which have a small number of examples assosciated with them. We aim to solve the reading comprehension problem for D_i and then transfer this learning, with minimum data to D_o.

3.2 Approaches

3.2.1 DistilBERT

We use a pre-trained DistilBERT QA model as our baseline for the project. The model consists of a DistilBERT to generate output embeddings followed by a dropout layer and QA head, which is a linear classifier, to produce the start and end index distributions of the answer. We use DistilBERT over BERT due to its efficiency and memory advantages - despite being 40% smaller in size than the original BERT and 60% faster in computation, it retains 97% of its language understanding capabilities [10].

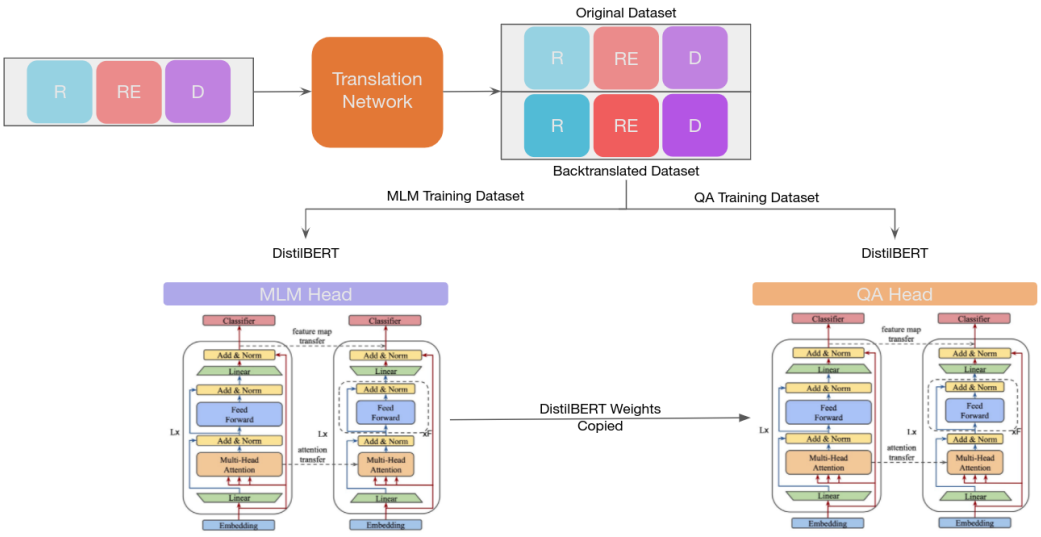

Figure 1: Model Design describing our best performing model

3.2.2 Transfer Learning

The different datasets that a model is exposed to may have different distributions or idiosyncrasies. Current QA models are brittle and perform bad on data which is out of distribution with respect to the data on which the model was trained on. In order to learn the specific distribution of a dataset, a model may need to be finetuned on it.

We follow the following approach for finetuning the DistilBERT on the out of domain datasets D_o. We first take a DistilBERT model pretrained on Masked Language Modeling and finetune it on the in domain datasets D_i. The trained model of the previous step is used to then finetune on D_o. We use early stopping on the validation dataset to pick the best model [11].

The common approach for using the pre-trained BERT model is to replace the original output layer with a new task-specific layer and fine-tune the complete model [9]. Previous work hypothesizes that the higher order layers of BERT work on higher order functions of language while the later layers are more specific to the task [11]. Thus we freeze the initial layers of DistilBERT and unfreeze the last two layers of DistilBERT to learn better representations for the specific dataset on which the model is being trained.

However, finetuning is very instable and also depends on the order of the datasets on which the model is trained [12]. Thus we experiment on taking different datasets from D_i for finetuning followed by finetuning on D_o and take the pick the best inital D_i based on validation set accuracies.

3.2.3 Mixture of Experts

To help adapt to the out-of-domain data, we also try training a mixture of experts model [2] to predict the start and end token distributions.

In the mixture of experts model, we train individual expert models specialized to different subsets of the in-domain data. More specifically, we train a model M_i for each combination of in-domain datasets and fine-tune on out-of-domain datasets. Finally, we have a gating model G which is parameterized by an MLP and determines the weighting of the different experts. During training, we freeze the weights of the experts and only train the mixture model G.

We experiment with two variants of the gating network G. In the first variant, we learn a mixture of the output logits from the expert models and then apply a softmax layer to produce the final distribution over start and end indices. In the second variant, we learn a mixture of the expert embeddings. More specifically, we learn a weight for each expert using the tokenized data d and the embedding e produced by each expert and take a softmax to produce the final weights. We finally compute a

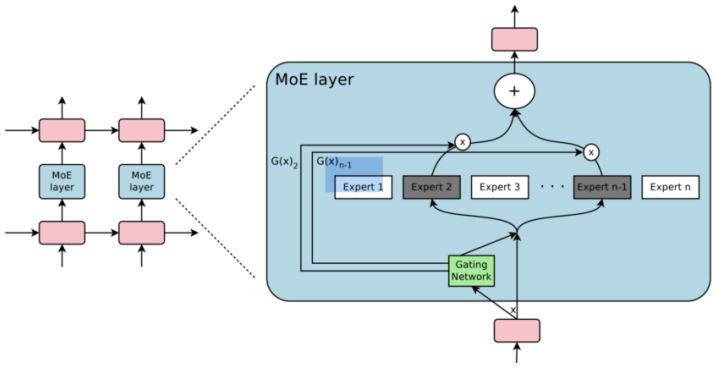

Figure 2: Image describing our approach for building a mixture of experts layer that combines the output of all the experts

weighted sum of the expert embeddings and pass this final embedding through a dropout layer and QA head to get the answer prediction.

3.2.4 Back Translation

The out of domain dataset D_o suffers from the problem of low data. A common approach in this setting is to enrich the training data by introducing new data examples. The idea is to use two translation models, one model which translates from English to a pivot lanuage L_{pivot}, and another which translates from L_{pivot} to English to obtain paraphrases of the text. This approach introduces more training data which contains some noise, even though the answer remains the same. This helps the model understand the real correlations in the data and guides it agianst discovering spurious correlations [6]. With more data we expect to regularize our model better.

Process for backtranslation Current backtranslation models have a limit on the number of input tokens. Hence while backtranslating the context C, we feed in the sentences (separated by a period) present in the context one-by-one. We keep the question unchanged, as noise in the question may change the meaning of the question and thus the answer to the question. We found that the model performed quite worse when the question was backtranslated. An additional consideration is that the answer will not remain the same after backtranslation. To tackle this we followed the approach of not backtranslating the sentence containing the answer span S.

The quality and diversity of the data generated can have large impact on the performance of the model so we consider different variations on the pivot language L_{pivot}. In order to get more diverse data, we also consider the approach of multiple pivot languages, i.e translating the context from English to L^1_{pivot} and then to L^2_{pivot} followed by translating back to English.

3.2.5 Masked Language Modeling

The DistilBERT model has been pretrained on large language corpora. Previous methods show that casting the question answering as a masked language modeling problem can make the language model a better few shot learner [8]. This helps the language retrieve information more easily. Thus, inspired by this approach we pretrain the DistilBERT model on the context data from D_o that we have. This is a relatively large amount of unsupervised data that can be learnt by the language model. GPT [13] has shown that masked language modeling is a powerful task which can capture most NLP tasks.

We mask out random 15% tokens from the context of the out of domain datasets and further pretrain a pretrained language model on this dataset using masked language modeling task. Here the pretrained language model acts as a good initialization. The hope is that adding a question answering head on top of this further trained language model would benefit from the domain specific correlations that have been learnt by the language model.

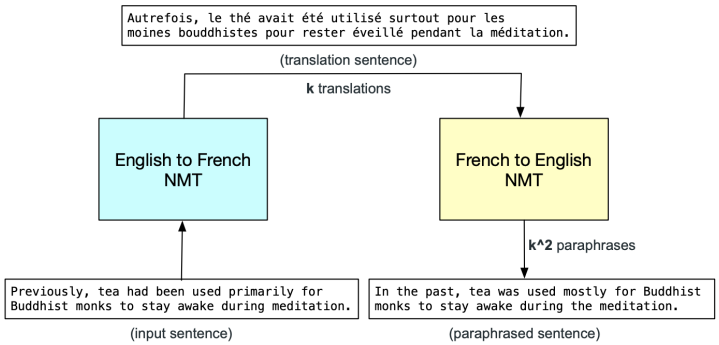

Figure 3: An illustration of the backtranslation model where French is used as the pivot language [6]. We see that we get the sentence meaning the same thing but with a different arrangement of words.

3.2.6 Regularization

As we are training a large language model on a very small dataset, it is completely possible for the model to remember the entire dataset and make predictions. In order to prevent this, it is essential to regularize the parameters of the model. We do this by adding a L2 regualarization to the weights of the model. Freezing the initial layers of the DistilBERT as mentioned before also acts like a regularization for the model.

4 Experiments

This section contains all our experimental details including dataset, evaluation method, experimental details, and results.

4.1 Data

We use a combination of three in-domain and three out-of-domain datasets to train and evaluate our model. Each of the three in-domain datasets contain over 50k training examples while each of the out-of-domain datasets contain only 127 training examples. The in-domain datasets we used were:

1. **SQuAD** [14]: This dataset is composed of contexts taken from Wikipedia articles and question-answer pairs generated by crowdsourced workers.

   The articles used by this dataset were selected using Project Nayuki's Wikipedia's internal PageRanks to ensure they are high quality. Specifically, they sample 536 articles from the list of top 10k articles suggested by the affomentioned pagerank and then pick all the paragraphs longer than 500 characters. Overall, this dataset has a total of 23,215 unique paragraphs covering a wide range of topics, from musical celebrities to abstract concepts.

2. **NewsQA** [15]: This dataset is based on a set of over 10,000 news articles from CNN with answers also consisting of the spans of the text from the corresponding articles. Crowd-sourced workers supply the question-answer pairs based on this repository of articles. The dataset is collected through a five-staged process designed to solicit exploratory questions that require reasoning. This process consists of article curation, question sourcing, answer sourcing, validation, and cleanup.

3. **NatQA** [16]: In contrast to other datasets in this section, NatQA was created via a question centric approach rather than a context centric approach. Questions consist of real anonymized, aggregated queries issued to the Google search engine. Then, an annotator (also crowdsourced) is presented with the wikipedia pages in the top five search results and must annotate a long and a short answer from it.

The out-of-domain datasets we used were:

1. **DuoRC** [17]: The contexts in this dataset are composed of movie plot summaries taken from IMDb and Wikipedia. Similar to the SQuAD dataset, this dataset also contains question-answer pairs collected from crowdsourced workers. What makes DuoRC unique is that for all movies, it contains plot summaries taken from both IMDb and Wikipedia and asks the workers to pick a question from one of the two plots and synthesize the answer from the second plot. this ensures that their is very little lexical overlap between the questions created from one version and the segments containing the answer in the other version.
2. **RACE** [18]: The RACE dataset, which stands for ReAding Comprehension Dataset From Examinations, was collected from english exams of middle and high school chinese students in the age range between 12 and 18. The questions in this dataset were generated by human experts and covers a variety of topics that are carefully designed for evaluating a student's ability in understanding and reasoning.
3. **RelationExtraction** [19]: The relation extraction dataset was developed with a slightly different goal in mind - to prove that relation extraction can be reduced to answering simple reading comprehension questions by associating one or more natural language questions with each relation slot. This dataset was collected by first gathering labeled examples for the task of relation-slot filling. After ccollecting several slot-filling examples via distant supervision, they convert their queries into natural language.

4.2 Evaluation method

We primarily rely on two metrics for evaluation:

1. Exact Match (EM): Refers to a binary score (0 or 1) of whether the output of our model exactly matches the ground truth answer. For example, if the answer to a question is "San Francisco city" and the model outputs "San Francisco", the answer would get an EM score of 0. The answers must match on the character level.
2. F1: Refers to the harmonic mean of precision and recall and is expressed as:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

   In the example presented above, the model's precision would be 100% (since the model answer is a subset of the ground truth answer) but recall would be 66.67% (since the model answer only contains 2 out of the 3 words in the ground truth answer). Hence, the final F1 would be 79.5%.

Throughout the training process, we evaluate the model on the validation split of our out of domain datasets at regular intervals and pick the model which does the best by F1 score. To get a final performance score of our model, we evaluate this trained model on the test split of the out of domain datasets.

4.3 Experimental details

Throughout our experiments, we only used DistilBERT - a distilled version of the BERT model along with a Question-Answering head. For the MLM training task, we replace this QA head with an MLM head. We experiment with a learning rates of 3e-5, 1e-5, 5e-6, and 1e-6. We found that 5e-6 worked better for finetuning RACE and DuoRC while 3e-5 worked better for Relation Extraction. We trained our models on an RTX 3070 GPU and found that it took us 5 seconds per epoch on the Relation Extraction dataset, while it took us 10 seconds per epoch on the RACE and DuoRC datasets. We train our model for 10 epochs, with a batch size of 16, and evaluate it every 20 batches. All experiments were run with a seed of 42 to maintain repeatability and allow us to compare approaches.

4.4 Results

In Table 1, we compare how combining various strategies affects scores on each of the three datasets. Note that in each cell, we display the score of model listed in the left most column after it has been finetuned on the training split of the dataset listed in the top row of the corresponding column. We find that using back translation along with masked language modeling achieves the best scores on

Models	RACE	RelEx	DuoRC	Average
Baseline + FT	33.17 (23.44)	64.13 (45.31)	**47.17 (33.33)**	48.90 (34.02)
Baseline + FT + MOE + BCK	29.38 (14.87)	63.49 (48.11)	33.72 (19.35)	42.19 (27.44)
Baseline + FT + BCK + MLM	**39.88 (24.22)**	75.19 (59.47)	42.95 (34.21)	52.67 (39.3)
Baseline + FT + MOE + MLM	33.21 (15.81)	66.70 (51.38)	34.11 (25.96)	44.01 (31.05)
Baseline + FT [NewsQA] + BCK + MLM	33.53 (21.76)	**76.23 (60.16)**	39.26 (30.09)	49.67 (37.34)

Table 1: Results of combining methods on validation sets for models trained on the OOD dataset [F1 (EM)]

RACE and Relation Extraction datasets while the simple finetuning approach works the best for DuoRC. We also find that using mixture of experts alongside backtranslation or masked language modeling both hurt our results. On the test dataset, our best model achieves an EM score of **46.766** (ranked 1st on test leaderboard) and an F1 score of **62.977** (ranked 4th on test leaderboard).

Models	RACE	RelEx	DuoRC	Average
Baseline + FT + MLM [10%]	37.49 (24.68)	71.91 (53.73)	36.89 (23.43)	48.76 (33.94)
Baseline + FT + MLM [20%]	38.78 (24.70)	72.58 (55.18)	37.27 (24.62)	49.54 (34.83)
Baseline + FT + MLM [30%]	38.14 (23.96)	71.46 (54.79)	37.13 (24.01)	48.91 (34.25)
Baseline + FT + MLM [50%]	36.73 (21.35)	69.82 (51.09)	35.64 (21.17)	47.39 (31.20)

Table 2: Results of training Masked Language Models on validation sets for models trained on the OOD dataset [F1 (EM)]

In Table 2, we compare different masked language modeling hyperparameters. Specifically, we analyze the impact of masking different percentages of training data while training the DistilBERT model. We find that masking 20% of the data works the best and masking any more reduces F1 and EM on all the three datasets.

Models	RACE	RelEx	DuoRC	Average
Baseline + FT + BCK [Hindi]	39.31 (26.56)	73.39 (55.47)	38.81 (25.75)	50.50 (35.92)
Baseline + FT + BCK [German]	36.67 (23.81)	70.27 (52.61)	35.34 (22.08)	47.42 (32.83)
Baseline + FT + BCK [Turkish]	38.34 (25.29)	71.90 (54.37)	38.09 (24.64)	49.44 (34.76)
Baseline + FT + BCK [Hindi] + BCK [Turkish]	37.72 (24.08)	72.87 (54.61)	36.48 (23.17)	49.02 (33.94)

Table 3: Results of training with backtranslation on validation sets for models trained on the OOD dataset [F1 (EM)]

In Table 3, we compare the impact of choosing pivot languages on augmenting the dataset using backtranslation. We find that languages that have models with higher BLEU scores, such as German, aren't as well suited for data augmentation as models with lower BLEU scores, such as those of Hindi and Turkish. We think that this is because having lower BLEU scores allows the model to introduce more variance in the augmented dataset, thus introducing the model to contexts it hasn't seen before. Since these backtranslated samples are much different than the existing dataset, they truly help the model in generalizing to other datasets. We also try appending datasets backtranslated with Hindi and Turkish and find that it reduces our performance for all three datasets. We think that this because having multiple copies of the dataset, even after backtranslation, introduces the model to similar content, thus allowing it to overfit to the training datasets. For the mixture of experts models,

Models	RACE	RelEx	DuoRC	Average
Baseline + FT + 3 MOE [Average]	27.38 (10.87)	61.49 (42.11)	31.72 (14.35)	40.19 (22.43)
Baseline + FT + 9 MOE [Average]	34.41 (19.73)	67.57 (52.43)	36.71 (22.98)	46.23 (31.71)
Baseline + FT + 9 MOE [MLP]	30.98 (16.27)	65.39 (49.61)	34.62 (20.85)	43.66 (28.91)

Table 4: Results of using mixture of experts on validation sets for models trained on the OOD dataset [F1 (EM)]

results are shown in Table 4. We find that averaging the results from the 9 experts performs better than training an MLP that finds weights for each of the experts and then performs weighted averaging. However, the results we see here were inferior to the baseline. We hypothesize that this is because

experts individually perform poorly and they all make similar mistakes, thus not giving us enough improvement in scores. We also think that since our MLP used as inputs the output logits rather than the input sentence, it was not able to correctly figure out the weights for each of the experts.

Models	RACE	RelEx	DuoRC	Average
Baseline + FT [All]	33.17 (23.44)	64.13 (45.31)	47.17 (3333)	48.90 (34.02)
Baseline + FT [SQuAD]	22.10 (15.62)	69.34 (49.38)	37.06 (26.19)	42.83 (30.39)
Baseline + FT [NatQA]	19.90 (14.06)	72.97 (51.56)	35.58 (23.02)	42.81 (29.54)
Baseline + FT [NewsQA]	25.43 (17.97)	75.18 (53.12)	39.31 (27.78)	46.64 (32.94)

Table 5: Results of using various finetuning strategies on validation sets for models trained on the OOD dataset [F1 (EM)]

In Table 5, we show the results of finetuning DistilBERT on different indomain datasets separately. We hypothesize that the model trained on Newsqa dataset and then finetuned on the relationship extraction performs better than the model trained on the complete in domain dataset and then finetuned on the relationship extraction dataset because the newsqa dataset's domain is closer to the relationship extraction dataset than other datasets' domains. Hence, the model has an easier time transferring knowledge from the newsqa dataset to the relationship extraction dataset.

5 Analysis

In this section, we analyze the output of the model and investigate its failure points.

5.1 Long contexts confuse the model

Inputs with a long context make it harder for the model to figure out the answer. There is more chance that the model picks the answer from a random sentence rather than the correct answer. This can be seen from Figure 4. We see that our model performs best on Relation Extraction (20 words/context) as compared to DuoRC (680 words/context) and RACE (290 words/context) which have large contexts. This was also tested by adding sentences to questions the model was already performing well on. For eg consider the following example :-

Context: Ray Eberle died of a **heart attack** in Douglasville, Georgia on August 25, 1979, aged 60.
Question: Why did Ray Eberle die?
Answer: heart attack

Consider adding the sentence :- He was a healthy man who used to run from San Fransisco to San Jose every day. In this case the model gets confused due to the additional context and predicts *used to run from San Fransisco to San Jose*. Thus additional context requires the model to reason about which sentences matter more and thus makes the task more difficult.

5.2 Fails at retrieving addresses, phone numbers etc

We see that the model performs well on inputs which require the model to retrieve English words from the context. However, a number of inputs require the model to retrieve addresses or phone numbers from the context. Consider the following example :-

Context: ... Please call 630-571-5466 http:// www.lionsclub.org Liquor Store For Sale Full equipment, located in Port Saint Lucie, Florida, U.S. Serious inquiries only. Call 302-393-3126 Cafe/Restaurant Business For Sale Busy location. Unbelievable price, $30,000. Call 302-650-4724
Question: Which number should you call to buy a restaurant?

Such questions requires reasoning about numbers and website links which may be out of domain for the BERT model and thus difficult for the model to predict. We find that in this case even though the model retrieves the wrong phone number/website, it still retrieves a phone number/address and not random text from the context.

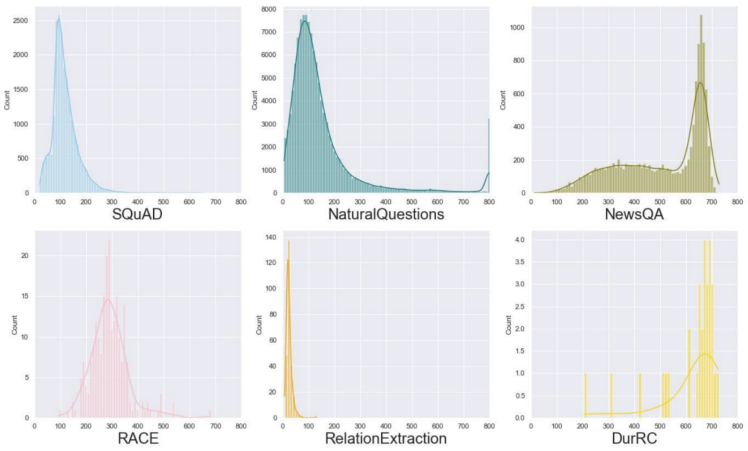

Figure 4: Number of words in the context for different datasets (D_o and D_i

5.3 Model tends to predict long answers

We find that the length of the answer predicted by our model seems to be long (around 15 words on average) while the average answer length in the dataset is 7 words. Thus the model tends to predict longer answers than required which hurts its F1 score.

6 Conclusion

In this paper, we explored several approaches to improve the robustness of question answering models in the few shot setting. We find that data augmentation techniques like backtranslation introduce more training data with some variance which helps the model generalize and remove spurious correlations. We also find that MLM pretraining helps language models in the few shot setting, allowing the language to store some information which can be used by the question answering module. We beat a baseline DistilBERT model and achieved a score which was **1st** on the leaderboard by EM and **4th** by F1. In the future, we would further investigate improving performance on longer context inputs to help the model perform even better.

References

[1] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709, 2015.

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

[3] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *CoRR*, abs/1911.02685, 2019.

[4] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. pages 6382–6388, November 2019.

[5] Siddhant Garg and Goutham Ramakrishnan. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online, November 2020. Association for Computational Linguistics.

[6] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.

[7] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *CoRR*, abs/1701.06538, 2017.

[8] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

[9] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*, 2020.

[10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[11] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

[12] Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. *arXiv preprint arXiv:1909.11299*, 2019.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

[15] 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.

[16] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[17] 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.

[18] 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.

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