Robust Question Answering Through Data Augmentation and TAPT

Stanford CS224N Default Project

Dunia Hakim
Department of Computer Science
Stanford University
dunia@stanford.edu

Abstract

In this project, we aim to improve on the given baseline model, which is a DistilBERT pretained transformer, as much as possible in order to make it more robust to out-of-domain data for the task of QA. In order to do this, we experimented with a variety of extensions to the baseline, among which are Task-Adaptive Pretraining [1] and data augmentation. We found that data augmentation was able to improve the results of the baseline the best out of our various attempts. Our best model performed better than the baseline by 0.287 points for the F1 score and 0.941 points for the EM score on the test set. The code for this project can be found on this Github repository: https://github.com/duniahakim/RobustQA. It is a private repository for Honor Code reasons but let me me know if you would like me to invite you as a collaborator to view it.

1 Key Information to include

- Mentor: Elissa
- External Collaborators (if you have any): None
- Sharing project: No

2 Introduction

There have been many advancements in the field of Natural Language Processing in the past few years. Yet, it also seems that in many cases the trained models learn superficial traits in order to perform their tasks. Unfortunately, this often means that the models will not generalize well on data that is not similar to their training data. In this paper, we will explore ways to make one particular system, i.e. a question answering system, more robust so that it is able to generalize better. In other words, we will try to tune a question answering model such that it will perform better on domains that it did not see or rarely saw during training.

Question Answering is a reading comprehension task where given a question a context paragraph, the model returns whether the question can be answered using the context paragraph, and if the question can be answered, then the model returns the chunk of text that answers the question.

In order to make the question answering system more robust, we use data augmentation as well as Task Adaptive Pretraining [1] that are both described in more detail in the Experiments section. We find that some data augmentation was able to improve the performance of the pretrained DistilBERT. However, TAPT [1] caused overfitting for our model as well and actually performed the worst out of all of our experiments described below. Our best model was able to perform better than the baseline by 0.287 points for the F1 score and 0.941 points for the EM score on the test set .

3 Related Work

One of the ways that NLP systems have been made more robust is by continuing to pretrain language models on domain- and task-specific data. This idea is explored in the paper "Don't Stop Pretraining" by Gururangan et. al. [1]. The paper explores the impact of using Domain-Adaptive Pretraining (DAPT) and Task-Adaptive Pretraining (TAPT) on ROBERTA [2]. DAPT is simply the idea of continuing pretraining ROBERTA on a large corpus of unlabeled domain-specific text. TAPT, on the other hand, refers to continuing to pretrain ROBERTA on the unlabeled training set for a given task. The paper found that DAPT always provides better or equal results to simply using ROBERTA. It also found that DAPT is more beneficial when the target domain is more distant from ROBERTA's source domain. As for TAPT, the paper found that TAPT almost always performs better than simply applying ROBERTA and TAPT also performs better than DAPT in the majority of domains.

The second idea that I reference for making an NLP system more robust is data augmentation. The paper "An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering" by Longpre et. al. covers how data augmentation can help a NLP model generalize better to data outside of its training data [3]. The paper discusses the benefits of various data sampling strategies as well as query and context paraphrases generated by back-translation. They find that the negative sampling designed to teach the model when to abstain from predictions are highly effective.

4 Approach

We started with the baseline model that is described in the project handout which finetunes a DistilBERT pretained transformer and uses a negative log-likelihood loss function. We then started to build on top of the baseline model by adding extensions.

First, we noticed that the out-of-domain training and validation sets are by default never used for training or validation. Therefore, our first attempt to try to improve on the baseline's default results is to train on both the in-domain and out-of-domain training set, and validate based on both the in-domain and out-of-domain validation set.

Second, as per the project handout's extension suggestions, we augmented the in-domain training datasets using back-translation. Given a context, a set of questions, and their answers, we used Google Translate API in order to translate the context and question to French and then back to English. We wrote code to do this back translation for all three in-domain training sets and saved the results in a new in-domain training set. We then fine-tuned the trained model from the previous point, Model 1, on any combination of these augmented datasets.

Third, we used Task-Adaptive Pretraining to continue pretraining the DistilBERT pretrained transformer on the out-of-domain training data. Specifically, we first artificially augmented each of the three out-of-domain training datasets by randomly masking different words (using the masking probability of 0.15) across epochs. Then we continued pretraining the DistilBERT pretrained transformer on these three augmented masked datasets. After the second phase of pretraining was complete, we fine-tuned the model on any combination of the three in-domain training datasets.

5 Experiments

5.1 Data

We used the provided datasets for the default project which consist of three in-domain reading comprehension datasets provided to us (i.e. SQuAD [4], NewsQA [5], and Natural Questions [6]) as well as the three out-of-domain dataasets (RelationExtraction [7], DuoRC [8], RACE [9])). The preprocessing needed for these dataset is described in the handout. It consists of chunking and caching, the details of which can be found in the handout.

5.2 Evaluation method

We are using the two evaluation metrics descriped in the project handout: Exact Match (EM) and F1. Exact Match is a binary measure of whether the system output matches the ground truth answer exactly. F1 is the harmonic mean of precision and recall1.

5.3 Experimental details

The hyperparameters of the model were not changed throughout all of the experiments. I instead decided to explore the impact of various data augmentation techniques. The hyperparameters used during training for all seven experiments are as follows:

Hyperparameter	Value
attention dropout	0.1
dim	768
dropout	0.1
hidden dim	3072
initializer range	0.02
max position embeddings	512
model type	distilbert
number of heads	12
number of layers	6
pad tokens ID	0
QA dropoug	0.1
Sequence Classify Dropout	0.2
Sinusoidal Position Embeddings	False
Tie Weights	True
Transformers Version	4.2.2
Vocab Size	30522

Table 1: Hyper-parameter Values

5.3.1 First Experiment

Our first experiment was fine-tuning the DistilBERT pretained transformer with its default hyper-parameters on the in-domain training and validation sets. We used a negative log-likelihood loss function. Fine-tuning the pretrained model took about 15 hours. This step did not require original code. We will refer to this model as the baseline model.

5.3.2 Second Experiment

The second experiment consisted of fine-tuning the baseline using different training and validation sets. Instead of only using the in-domain training and validation sets as in the first experiment, in the second experiment we trained and validated on both the in-domain and out-of-domain training and validation sets (i.e. SQuAD[4], NewsQA [5], and Natural Questions [6], RelationExtraction [7], DuoRC [8], RACE [9]). The fine-tuning in this experiment took around 18 hours and did not require original code.

5.3.3 Third Experiment

In the third experiment, we first augmented the training set and validation set of NewsQA [5]. To do this, we wrote code that went through the dataset's JSON object and translated all the contexts and questions in the JSON object to French and then back to English and created a new JSON object with the back translated versions of the contexts and questions (along with other data that was not back translated). For the translation, we used the Google Translate APIs. Once we had created the new augmented datasets for NewsQA, we fine-tuned the model from the previous experiment, experiment 2, with its default hyperparameters on the NewsQA augmented training and validation dataset. This experiment involved original code written by us.

5.3.4 Fourth Experiment

The fourth experiment is almost identical to the third experiment except that we augmented the Natural Questions dataset [6] instead of NewsQA [5]. As before, after creating the new augmented dataset for Natural Questions dataset, we fine-tuned the model from experiment 2 with its default hyperparameters on the Natural Questions augmented training and validation dataset.

5.3.5 Fifth Experiment

For the fifth experiment, we fine-tuned the model created in experiment three on the Natural Questions augmented dataset from experiment four. There was no data augmentation in this experiment, but rather the use of previous models and augmented datasets.

5.3.6 Sixth Experiment

For the sixth experiment, we first augmented the training set and validation set of SQuAD[4] similarly to experiment three and four. We then used the model created in experiment three and fine-tuned it on the SQuAD augmented dataset we just created.

5.3.7 Seventh Experiment

For the seventh experiment, we used Task-Adaptive Pretraining (TAPT) [1] to make the model more robust to the task at hand. In order to do this, we first augmented each of the three out-of-domain training datasets by randomly masking different words (using the masking probability of 0.15) in the context paragraphs and questions just as in the paper [1]. After the three out-of-domain datasets have been augmented and masked, we started from the DistilBERT pretained transformer and continued to pretrain the DistilBERT on the three augmented and masked out-of-domain training datasets. After pretraining was complete, we then fine-tuned the resulting model on the three in-domain training and validation datasets. This experiment required additional original code.

5.4 Results

Below you can find a table with the EM and F1 scores of each of the experiments. Note that the test scores are only available for four experiments since we are only allowed to submit to the test leaderboard four times.

Experiment	Validation Scores		Test Scores	
	F1	EM	F1	EM
1	48.432	33.246	59.187	40.275
2	48.85	33.77	-	-
3	**49.12**	**33.77**	**59.474**	41.216
4	47.43	30.63	-	-
5	47.59	30.63	58.393	40.298
6	47.43	31.94	59.348	**41.766**
7	39.01	23.04	-	-

6 Analysis

We can see from thee results above that the third experiment achieved the best F1 and EM scores. This was the experiment with the data augmentation for NewsQA. When we tried to fine-tune this best model on more augmented data it actually started performing considerably worse as you can see from the results of experiment five and six. In order to understand why this might be the case, we compare the prediction results on the validation set for the model from experiment three and five. It seems from the predictions that although in most cases the predictions come from the same sentence in the context paragraph for both models, it is clear that the model from experiment five tends to use a bigger slice of the sentence than the model from experiment three. For instance, for one of the questions, the experiment 5 model predicts "age of fourteen" while the experiment 3 model simply predicts "fourteen".

Additionally, we can see from comparing experiment 3 results with results from experiment four that the augmented NewsQA dataset had a better performance than the augmented Natural Questions dataset. This might be due to the fact that the model in experiment four is overfitting. We can see this from the training F1 and EM scores. At the end of training for experiment four, the F1 score is 68.74 and EM is 52.22, whereas for experiment three, the F1 score is 56.81 and EM is 38.87. We are not quite sure why the experiment four model is overffiting, especially since it is fine-tuned on a larger dataset than in experiment three. However, the overfitting in experiment four explains why trying to fine-tune on more data such as in experiment five and six also results in overfitting and worse performance on the validation sets. Interestingly, although the performance of experiment six is worse on the validation test, it actually has comparable results to experiment three on the test set.

Lastly, we can see from the results that experiment seven performed significantly worse than the previous experiments relying on data augmentation. It seems from the training F1 and EM scores, which were F1 score of 67.50 and EM of 51.42 by the end of epoch 2 in training, that the model was highly overfitting.

7 Conclusion

In this paper, we explored the impact of data augmentation and TAPT [1] on the performance of DistilBERT pretained transformer on out-of-domain data. Overall, we found that some data augmentation was able to improve the performance of the pretrained DistilBERT. Interestingly, the impact of data augmentation was only positive in some instances, specifically when augmenting the NewsQA dataset, while in other cases data augmentation actually caused the model to perform worse. This was often due to the model overfitting on training set. We additionally found that TAPT [1] caused overfitting for our model as well and acutally performed the worst out of all of our experiments. Our best model from the seven experiments was model three where we augmented the NewsQA training and validation sets and finetuned the model from experiment two on those augmented dataset. This was able to improve the baseline model by 0.688 points for the F1 score and 0.524 points for the EM score on the validation set, and by 0.287 points for the F1 score and 0.941 points for the EM score on the test set.

As for future work, there are many more areas that could potentially be explored. For instance, we could explore how to stop the models from overfitting when using data augmentation and TAPT, and how to take better advantage of the additional data from data augmentation without decreasing performance. Additionally, we could try using both TAPT and data augmentation. We did not attempt this since we guessed that without solvinng the overfitting issue, trying to use both TAPT and data augmentation will likely also result in overfitting.

References

[1] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

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

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

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.