Few-shot QA using DNN

Stanford CS224N {Default RobustQA} Project

Adil Sadik
Department of Computer Science
Stanford University
asadik@stanford.edu

Abstract

Large pre-trained models were shown to yield strong results on QA tasks and often outperform human baseline [1]. However these systems don't perform well in a few shot or zero shot settings [2]. In this project we propose a domain adaptation technique for BERT using meta learning based pre-training approach. We introduced a new special [QUESTION] token to learn relationships between question and context passages during training. The model is trained using a modified reptile [3] based meta learning algorithm. The inner loop of original reptile algorithm is modified to sample batches randomly from all tasks instead of sampling from a specific tasks in each step. This improves generalization across multiple domains. Our model achieved F1 and EM score of 52.397 and 39.005 in out of domain validation set and EM:40.023 and F1:57.800 in out of domain test set.

1 Key Information to include

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

2 Introduction

The standard approach to extractive question answering is to pretrain a large language model using MLM objective and fine-tune it with a span selection layer to extract the answer span from context passage [4]. When trained on large amount of task specific data, these system were shown to yield strong results and often outperform human baseline [1]. However the success of these models are based on the assumption that large quantities of annotated data are available. Many training sets contain 100K [5] or more annotated data. This assumption quickly becomes unrealistic because creating annotated data is a manual process, expensive, requires domain knowledge and time consuming. Moreover the model tend to overfit when trained on a large amount of data from a particular domain and fails to perform well on out of domain dataset for the same task. This implies that the QA models fail to learn domain invariant features.

In this work we investigated the effectiveness of various domain adaptation techniques to improve the accuracy and robustness of QA model on out of domain dataset. Specifically, we investigated GAN[6], Task adaptive pretraining (TAPT) [7] and meta learning [3] based domain adaptation approaches. The key idea is to prevent the model from overfitting to the in-domain training dataset so that it can be easily fine-tuned on a new out of domain dataset with only a few labeled samples. We first pre-trained the model using new task specific [QUESTION] token and then fine-tune it with out of domain data sets only. We hypothesize that as the model didn't see [QUESTION] token before, we can learn better contextual representation of the task by adding this new specific vocabulary. We achieved 70.8 F1 and 54.81 EM score in in-domain validation set with this new token. To train the model,

we used a slightly modified reptile based meta learning algorithm. Reptile improves generalization by maximizing the inner dot products of the gradients of different minibatches of the same task or domain [3]. We hypothesize that it's possible to further improve reptile's performance by sampling minibatches from multiple tasks (or domains) in it's inner loop instead of sampling minibatches from a single task or domain. With this modified reptile based meta learning and new task specific pre-training objective we were able to improve QA result upon baseline.

3 Related Work

Even though the deep learning QA models surpass human-level performance, they perform poorly on out-of-domain dataset. To address this, various domain adaptation or generalization approaches are proposed. [8] proposed a gradient reversal layer which is similar to GAN or adversarial training to perform domain adaptation. These techniques attempts to penalize the model for learning domain specific deep features as training progresses. [6] showed promising results on QA tasks using GAN. [9] and [10] proposed QA task specific MLM pre-training objectives in large pre-trained language models to generalize learning of QA task. [11] proposed a recurrent span selection algorithm to automatically generate QA tasks from unlabelled data. The model masks recurrent spans from the context passage and creates question-answer pairs from these masked spans. The model showed promising results but the success of this model depends on the presence of recurring spans in the context which may not generalize well to a new domain. [10] trained BART and T5 models to generate answers in auto-regressive manner from [mask] token. Another popular domain adaptation technique is meta learning [3] which attempts to find a better initialization parameters for the model that can be easily fine tuned for few-shot tasks. In our work, we attempt to combine meta learning with MLM based pre-training. We use pre-training to improve learning of QA task specific features and use meta learning to find an optimal set of parameters and prevent overfitting.

4 Approach

We build upon the standard span selection QA model built using pre-trained large language models. We refer to [7] to learn more about the baseline model. For domain adaptation we combine a novel pre-training objective with custom token with a variant of reptile meta learning algorithm to learn domain invariant features.

4.1 Pre-training

Our first approach is to jointly optimize MLM and QA task. We randomly mask a subset of tokens in the answer span and train the model to predict the masked answer. The intuition behind this approach is that as the baseline DistilBERT model is trained using MLM objective, by masking the answer spans we can improve the contextualization of question-answer relationships. Following modified loss function is used to train the model:

$$loss = \lambda l_s + (1 - \lambda) l_m \tag{1}$$

Here l_s and l_m are losses for span selection and MLM objectives. λ is a tuneable hyperparameter.

We also explore a novel task specific pre-training method by adding a new special token in the vocabulary. We hypothesize that adding a new special token (similar to [mask]) will improve learning of task specific features as the model didn't see this token before. During pre-training the model is trained to minimize span selection loss with a new [QUESTION] token.

4.2 Meta Learning

Meta learning is an approach for learning with small amount of data. A variety of meta learning approaches have been proposed. We refer to [12] to review all domain adaptation techniques including meta learning for a broad survey. In this work we specifically focus on using meta learning to find an ideal initialization of model parameters. A simple but efficient meta learning algorithm was proposed by OpenAI in [3]. The authors proposed an algorithm - reptile - to find an initial set of parameters

ϕ such that for a randomly sampled task τ and loss L_τ the learner will minimize loss in k steps. Formally:

$$\underset{\phi}{\text{minimize}}\, \mathbb{E}_\tau[L_\tau(U_\tau^k(\phi)] \tag{2}$$

Here k denotes the k inner steps in algorithm 1 and U_τ^k denotes the operator that updates ϕ after k inner steps of gradient descent for task τ

Algorithm 1 Original Reptile algorithm

initialize $\theta = \phi$ where ϕ is the meta-model param
for $i \in 0, 1, 2, 3...T$ **do**
  Sample a task τ
  for $k \in 0, 1, 2, 3...K$ **do**
    Sample batch from task τ
    Compute $\theta = U_\tau^k(\theta)$
  end for
  Update $\phi = \phi + \epsilon \frac{1}{k} \sum_{i=1}^{k} (\theta - \phi)$
end for

Algorithm 1 is the original reptile algorithm introduced in [3]. One subtle thing to note here is that $k = 1$ in Algorithm 1 will yield the vanilla SGD algorithm. Fundamentally, Algorithm 1 is taking multiple gradient steps ($k > 1$) before updating the meta parameter and by constraining $k > 1$ it is incorporating information from second or higher order derivatives of loss function L_τ in the meta model parameter. We refer to [3] to review the formal derivation of the loss function. However, we present one critical observation from [3] here and develop the intuition about how reptile algorithm can be modified to improve the initialization point of meta parameters. In [3] authors used Taylor series to approximate the update performed by the algorithm and showed that the gradient can be derived as (equation 40 [3]):

$$gReptile = \sum_{j=1}^{k} \hat{g}_i - \alpha \sum_{i=1}^{k} \sum_{j=1}^{i-1} \hat{H}_i \hat{g}_j + O(\alpha^2) \tag{3}$$

Here g_i is the gradient at step i, $\hat{g}_i$ is the gradient w.r.t loss $L\tau$ for the batch of task τ. α is the step-size and $\hat{H}_i$ is the Hessian. The paper points out that the first term minimizes the expected loss and the second term ($\hat{H}_i \hat{g}_j$) maximizes within-task generalization. It is basically the inner dot product between different batches of same task. Our intuition is that we can modify the second term to improve generalization. The algorithm can be easily modified to sample mini-batches randomly from all domains or tasks instead of sampling batches from the same task in each inner step ($k \in 0, 1, 2, 3...K$) in Algorithm 1. This will maximize the cross-task generalization. In our work we perform experiments with randomly sampling batches from (a) all tasks, (b) one task at a time (Algorithm 1) and (c) N tasks from N domains. We show that (c) provides the best accuracy in the OO domain QA task.

Algorithm 2 Modified Reptile algorithm

initialize $\theta = \phi$ where ϕ is the meta-model param
for $i \in 0, 1, 2, 3...T$ **do**
  for $k \in 0, 1, 2, 3...K$ **do**
    Sample a task τ
    Sample batch from task τ
    Compute $\theta = U_\tau^k(\theta)$
  end for
  Update $\phi = \phi + \epsilon \frac{1}{k} \sum_{i=1}^{k} (\theta - \phi)$
end for

5 Experiments

5.1 Data

We use in domain and out of domain datasets from [7] to train and evaluate the model. For in domain SQuAD, NewsQA and Natural Questions datasets are used. For out of domain, we used DuoRC, RACE and RelationExtraction. We refer to [7] for a detailed review of the dataset descriptions.

5.2 Evaluation method

We used EM and F1 scores as evaluation metrics. Exact Match (EM) is a measure of whether the model output matches with ground truth exactly. F1 is a less strict metric and it accounts for partial match. We refer to [7] for a detailed introduction and description about EM and F1 metrics. EM and F1 scores are averaged across the entire training/evaluation dataset to measure the model performance.

5.3 Experimental details

Table 1 contains a comprehensive list of all the important experiments performed for domain adaptation. First, we implement and analyze the effectiveness of joint training with auxiliary loss. We refer to this model as **QA-MLM**. The loss function in 1 with hyperparameter λ is used for training. p is the masking probability. We randomly mask tokens from answer spans only and compare it with **TAPT** where tokens from both question and answers are masked. This approach improved the accuracy of the model over baseline but it was not significant. [9] showed promising results in various NLP tasks by pretraining LM on QA tasks with bi-encoders where question and contexts are encoded independently. We implemented this architecture but it didn't turn out to be fruitful. To better contextualize QA task representations in the model, we augmented **QA-MLM** model by adding two special token types - [QUESTION] and [ANSWER]. This model is referred as **QA-MLM-NV** (NV for **N**ew **V**ocab). [QUESTION] token is used for question representation and [ANSWER] is used to mask answer tokens in context passage. However, our final model only uses [QUESTION] token as using two new tokens didn't show any significant benefits. [QUESTION] is added after [CLS]. Adding new [QUESTION] token significantly improved accuracy of **QA-MLM** in in-order validation set (F1: 70.80, EM: 54.81, without any auxiliary loss ($\lambda = 1$)). Hence, we take **QA-MLM-NV** as our base model and explore ways to transfer the learning to out of domain dataset.

We utilized GAN and Meta learning approach to train **QA-MLM-NV** model. Adversarial training have been used for QA tasks to learn domain invariant features [6]. However in our experiments, meta learning proven to be most effective. We utilized a slight variation of reptile [3] meta learning algorithm as discussed in previous section to train the model. We refer to our final model as **META+QA-MLM-NV** in Table 1.

We also experimented with training by freezing the word embedding layer, dynamically masking answer tokens in each batch, adding new FC layers at the output span selection logic etc. Results are reported in Table 1.

Hyper-parameters:

- λ for Auxiliary loss and p masking probability
- Inner step K in Algorithm 1. We use $k = 6$ for pre-training and $k = 4$ for fine-tuning.
- Default learning rate is $3e-5$. For meta learning models we tune learning rate for inner and meta optimizer. In the final model inner learning rate is set to $5e-5$ and meta learning rate is set to $3e-5$
- QA-MLM models took 4-5 hours to train. META+QA-MLM models took 8-10 hours to train as meta learning is slower and requires more steps and two models. Final META+QA-MLM-NV[3] model is pre-trained for 600 iterations and fine tuned for 50 iterations.
- Default batch size is 16

5.4 Results

Table 1 contains the results from our experiments. We make following observations from the results:

Model	EM	F1	Description	Configuration
Baseline	32.8	48.8	-	-
QA-MLM[1]	30.1	44.1	Joint training with Aux loss	$\lambda = 0.5$ $p = 80\%$
QA-MLM[2]	35.1	49.9	Joint training with Aux loss	$\lambda = 0.8$ $p = 50\%$
QA-MLM[3]	32.9	48.6	Joint training with Aux loss	$\lambda = 0.9$ $p = 30\%$
QA-MLM[4]	28.53	42.93	Joint training with Aux loss Dynamic masking with weight decay	$\lambda = 0.5$ $p = 30\%$
QA-MLM[5]	31.0	43.3	Joint training with Aux loss Frozen word embedding	$\lambda = 0.8$ $p = 80\%$
TAPT-MLM	33	49	TAPT- mask both question and context	$\lambda = 0.8$ $p = 20\%$
Bi-encoder** (training stopped after 10K iter)	10	12	Independently encode question and context	$\lambda = 0.8$ $p = 20\%$
QA-MLM-NV[1]	28.53	45.90	Pretrain with [QUESTION] and [ANSWER]	$\lambda = 0.8$ $p = 50\%$
QA-MLM-NV[2]	31.15	46.07	Pretrain with [QUESTION]	$\lambda = 1$
QA-MLM-NV[3]	31	46.1	Pretrain with [QUESTION] and 2 FC span layer at the output.	$\lambda = 1$
GAN+QA-MLM[2]	26.70	40.71	Incorporate Adversarial training	$\lambda = 0.8$ $p = 50\%$
META+QA-MLM-NV[1]	32.98	49.03	Meta Learning with QA-MLM-NV variants	$\lambda = 0.7$ $p = 20\%$
META+QA-MLM-NV[2]	37.17	51.91	Meta Learning with QA-MLM-NV variants	$\lambda = 1$ batch=32
META+QA-MLM-NV[3]	**39.01**	**52.45**	Meta Learning with QA-MLM-NV variants	$\lambda = 1$ inner_lr=$5e-5$ batch=32

Table 1: Experiment results in OO val set.
Hyperparameters not shown in last column are set to default

- QA-MLM is able to jointly optimize span selection loss and MLM loss. As the only the answer tokens are masked, this model is able to learn relationships between question spans and the answer spans in the context passage. TAPT didn't perform as well as QA-MLM. One possible explanation is, TAPT may tend to over-fit more compared to QA-MLM as both question and answer tokens are masked. Even though QA-MLM model performed well, the

improvement in EM and F1 scores are not significant. We think this is due to overfitting. The model is not able to generalize to out of domain data as it overfits to in domain training set.

- Bi-encoder model didn't perform as well as QA-MLM. Our intuition is that bi-encoder model requires more training data than cross-encoder models as question and context are encoded independently which is beyond the scope of this project.
- The model outperformed in in-domain validation set when trained with the new [QUESTION] type (baseline F1: 64.56, EM: 48.86, QA-MLM-NV^2 F1: 70.8, EM: 54.81) It indicates that the model is able to do a better job learning task specific signals when trained with new task specific token types. Intuitively it makes sense because the model didn't see the new token before and it is forced to learn new parameters by minimizing the QA span selection loss. However the model again overfits to in domain dataset and fails to generalize well.
- Adversarial training didn't yield strong result as expected and we suspect that it's due to inherent complexity of adversarial training. In GAN the nature of loss landscape changes drastically as two models are trained simultaneously which makes it harder to train.
- The meta learning algorithm discussed in section 4.2 is able to find a good initialization point of META+QA-MLM-NV model parameters. It is expected because the algorithm improves generalization by randomly sampling batches from all tasks in it's inner gradient steps before updating the final meta model parameters. Increasing inner optimizer learning rate improved accuracy because the dot product is multiplied by inner learning rate in 3.

Leaderboard scores:

- Validation EM: 39.005, F1: 52.397.
- Test EM: 40.023, F1: 57.800

6 Analysis

The model didn't perform well in final test set. Following are some of the model reference and actual outputs from out of domain validation set:

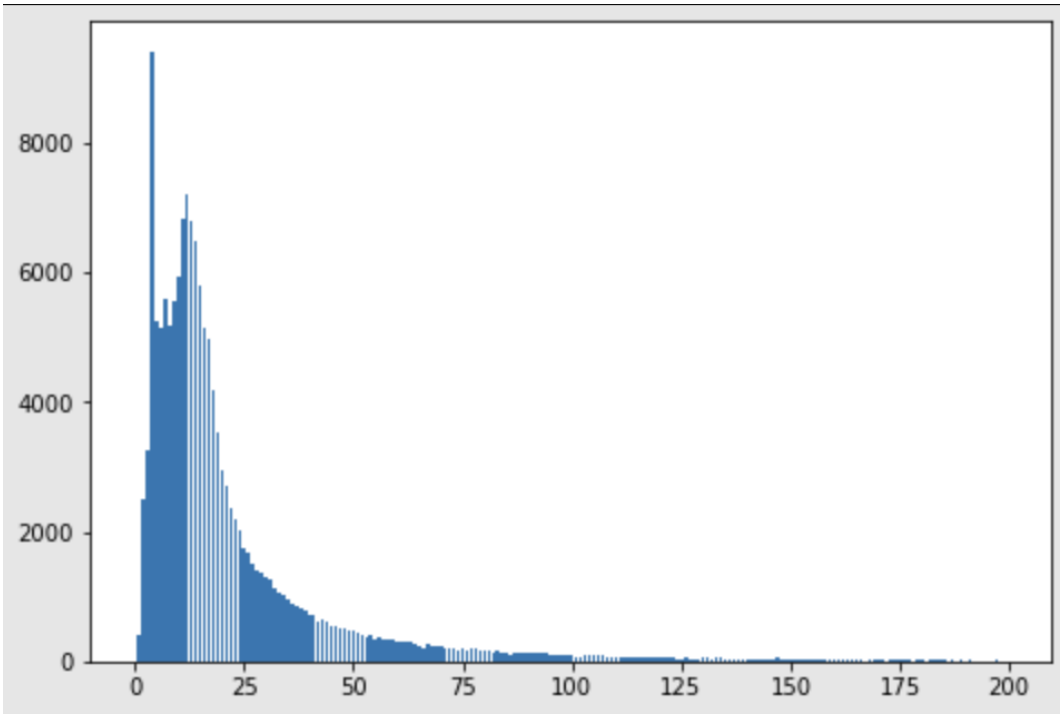
Context: Televisions were among the most talked about items at the 2013 International Consumer Electronics Show last week in Las Vegas, Nevada. Some employed the most advanced technology ever. Some of the TVs used a new technology called Organic Light Emitting Diodes, or OLED. They were thinner, lighter, offered better color1 and were brighter than traditional LEDs. Smart TVs this year were smarter. Many offered technology that let users have a more personalized experience. One such TV from the electronics company TCL uses sensors and voice recognition to determine who is watching. It then offers programming based on the specific user. Another TV from Panasonic offers a similar personalized user experience. In addition to television technology, size also played a major part in CES 2013. Televisions varied in size from big to bigger, with at least two companies – Samsung and HiSense – exhibiting TVs measuring 110 inches. The yearly Consumer Electronics Show is the biggest technology trade show in North America and one of the biggest in the world. Gary Shapiro is president and CEO of the Consumer Electronics Association, the group that organizes CES. He gave one of the keynote speeches on opening day. N̈ow you know that CES is more than a trade show. It's a gathering of the brightest minds and the top leaders from many industries and those seeking a glimpse into the future.T̈hat glimpse into the future included a look at digital health and fitness devices, which were also big at CES 2013. There were devices that track your activity and others that measure blood pressure, heart rate and weight. There was even a fork that tells you when you are eating too fast. Cars, smart-phones, tablet computers and PCs also made news. And a 27-inch table computer drew quite a bit of attention. CEA President Gary Shapiro says there was much to see but not nearly enough time to see it all. Ÿou cannot see the show in the four days that you have. We have over 3200 different industries showing over 20,000 new products. It's ly incredible.

Question: At the 2013 CES, which item drew the most attention?
Model output: a 27-inch table computer
Reference output: Televisions

Context: Last week I visited my friend Pete in the new home where he lives with his wife and daughter. Pete used to spend his holidays travelling the world, visiting the pyramids in Egypt or scuba diving in the Caribbean. Nowadays he prefers to spend his holidays and weekends making his house look more beautiful. Like hundreds of thousands of other British people, he has discovered the joy of DIY (Do It Yourself), which means if there are any things that need fixing around the house, he will try to do the job himself. As he showed me the new kitchen he put together by himself and the newly painted walls, I asked Pete where he got his inspiration from. He told me that his favorite source of ideas was a DIY program on TV. This got me thinking about the great popularity of DIY programs in the UK. Each major channel has at least one home or garden improving show and there's even a satellite channel completely about the subject. I guess it is not really surprising that DIY programs are so popular. Two common sayings in Britain- 'an Englishman's home is his castle' and 'there's no place like home'-show how important our houses are to us. With the present economic downturn, many people can't afford to buy a bigger house so they are looking at how they can make their house better without spending a lot of money. DIY is the perfect choice. But be careful! I read a report that said over 230,000 people were injured while doing home improvements in the UK in just one year, including 41,000 who fell off ladders and 5,800 who were seriously hurt by hammers. So I won't be going down to the hardware store.

Question: Which might NOT be shown in a DIY program on TV?
Model output: how important our houses are to us
Reference output: scuba diving

As seen here, the model performs poorly for large context paragraphs and in relationship extraction tasks. Moreover, approximately 90% of the model output in oodomain validation set contains less than 5 words. It indicates that the model is heavily biased toward extracting short span of factual answers. The EM score (ranked 4 in leaderboard - EM: 39.005) on oodomain validation set corroborates this. We think that this bias is partly influenced by the dataset. Following plot depicts the number of charterers in the answer spans of all indomain datasets in X axis and number of occurrence in Y axis.



The plot shows that most of the answers are 5-25 charterers long. It averages to approx 5 words. The model gets very good at learning how to extract short spans of answers from relatively short context passages due to this high bias in dataset. However we don't think that dataset alone is responsible for the performance degradation in test set. The new [QUESTION] token is aiding the model to overfit to training dataset by learning narrow task specific parameters.

7 Conclusion

Pre-trained language model are good at memorizing and they tend to overfit to training dataset. In this project we investigate various domain adaptation methods in the context of extractive QA task and show that meta learning is effective at finding a good initialization of model parameters.

References

[1] squad explorer. `https://rajpurkar.github.io/SQuAD-explorer/`.

[2] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328, 2017.

[3] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[6] Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training. *CoRR*, abs/1910.09342, 2019.

[7] Robust qa track. `http://web.stanford.edu/class/cs224n/project/default-final-project-handout-robustqa-track.pdf`.

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Advances in Computer Vision and Pattern Recognition*, page 189–209, 2017.

[9] Robin Jia, Mike Lewis, and Luke Zettlemoyer. Question answering infused pre-training of general-purpose contextualized representations. *CoRR*, abs/2106.08190, 2021.

[10] Rakesh Chada and Pradeep Natarajan. Fewshotqa: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models. *CoRR*, abs/2109.01951, 2021.

[11] Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. Few-shot question answering by pretraining span selection, 2021.

[12] Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *CoRR*, abs/1904.05046, 2019.