Meta-Learning for Question Answering on SQuAD 2.0

Stanford CS224N {Default} Project
Track: {RobustQA}

Chi-Hsuan Chang
Stanford University
chc155@stanford.edu

Abstract

In a general Question Answering (QA) system training, we teach the system to answer a question from understanding the associated paragraph. Often time, we'd encounter the challenge in learning new tasks from different domains with limited samples. In this study, we built a QA system that is robust to domain shifts with an implementation of the meta-learning (MAML) framework for few-shot supervised learning [1]. We explored training (1) MAML models from scratch and (2) MAML after baseline model pre-trained and fine-tuned. We found that training MAML after fine-tuning baseline outperformed the baseline occasionally, and the best performing model on the out-of-domain validation set was a 10-task 20-shot MAML that scored **EM: 40.436** and **F1: 50.49** on the out-of-domain test set.

1 Key Information to include

- Mentor: Grace Lam
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

In a Question Answering (QA) system, the system learn to answer a question by properly understanding an associated paragraph. However, developing a QA system that performs robustly well across all domains can be extra challenging as we do not always have abundant amount of data across domains. Therefore, one area of focus in this field has been learning to train a model to learn new task with limited data available (e.g. Few-Shot Learning, FSL).

Meta-learning in supervised learning, in particular, has been known to perform well in FSL, with the concept being teaching the models learn to set up initial parameters well that enable the model to learn a new task after seeing a few samples of the associated data.[2, 1] In this study, we were given a large amount of in-domain (IND) samples with only limited samples of out-of-domain (OOD) set. We were provided with a fine-tuned (FT) DistilBERT model [3] that knew to perform well on the IND set. To improve the robustness of the FT baseline model performance on OOD set, we trained:

- MAML models from scratch
- MAML models after baseline model was pre-trained and fine-tuned

3 Related Work

In a meta-learning setting, two gradient descent update loops are defined. Tasks are defined as a pool of repeated sampled training sets (e.g. support and query). Specifically, a task consists of K support and Q query samples from N classes is defined as K-shot N-way MAML. The inner-loop task learners

optimize their model parameters training on different tasks. The aggregated loss learned across the tasks are passed on to the outer loop meta-learner. The goal of the meta-learner is to optimize its parameters that minimize the error across all inner-loop learners.[2, 4]

Model-Agnostice Meta-Learning (MAML) was originally proposed by Finn et al. in 2017.[1] MAML is designed to be model-agnostic, but it assumes the same model architecture (e.g. the same parametric setup) across task-learners. MAML has been implemented in supervised, unsupervised and reinforcement learning with few improvements made since it published. MAML++ achieves better performance by solving few limitations of MAML by training task learners with per-step learning rates, batch normalization parameters and optimizing on per-step target losses. [5] Implicit MAML (iMAML) overcomes the original need to differentiate during inner-loop training which resolves the computational and memory burdens. It also implemented an inner-loop regularization step to optimize the parameters of the task learners rather than solely leveraging early stop of the gradient descents.[6]

4 Approach

4.1 FT Baseline

Our baseline model was a FT pre-trained transformer model DistilBERT [3] from the Default RobustQA track. The baseline QA model was trained on all IND training sets (as described in the Data section), and was validated on the IND validation set.

4.2 Model-Agnostic Meta-Learning (MAML) DistilBERT

We adapted the **Algorithm 2 - MAML for Few-Shot Supervised Learning** by Finn et al. 2017 [1] as a model improvement. We trained MAML DistilBERT models with IND and OOD training datasets aiming for fast-adapt K-shot learning as an alternative to the baseline model. Illustration of the model architecture was shown in Figure 2.

- We defined the FT DistilBERT [3] model (e.g. model provided by the cs224n teaching group) as our base learner (f_θ).
- We implemented a task method rather than to pre-define a K-shot task pool ($p(\mathcal{T})$). To form each task ($\mathcal{T}_i$), we randomly sampled K samples (e.g. K-shot) as the support set ($\mathcal{D}_i$) and K samples ($\mathcal{D}_i'$) as the query set from IND and OOD training datasets (as discussed in the experiments).
- We used the same loss function specified in the baseline model ($\mathcal{L}$, **loss** $= -\log p_{start}(i) - \log p_{end}(j)$).
- We implemented the inner-loop training step to optimize parameters of the DistilBERT with training support ($\mathcal{D}_i$) from a task $\mathcal{T}_i$. The gradient descent optimization was calculated using the loss function (e.g. $\mathcal{L}_{\mathcal{T}_i}$). Specifically, we implemented the one gradient update step as the following equation: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}(f_\theta)$.
- We implemented the meta-step optimization of the DistilBERT with training query ($\mathcal{D}_i'$) across tasks with the following equation: $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{T_i}(f_{\theta_i}')$. The intention of this step was to learn to minimize the sum of the gradients across all tasks.
- We evaluated the MAML model with the IND and OOD training datasets (as discussed in the experiments).

4.3 FT Baseline + MAML DistillBERT

In addition to training MAML model from scratch, we also leveraged the FT DistillBERT (Baseline) model and trained the MAML model from the checkpoint of the baseline model with IND or OOD training sets.

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
7: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8: Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
9: **end for**
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
11: **end while**

Figure 1: Summary of MAML algorithm for few-shot supervised learning. Figure cited from [1]

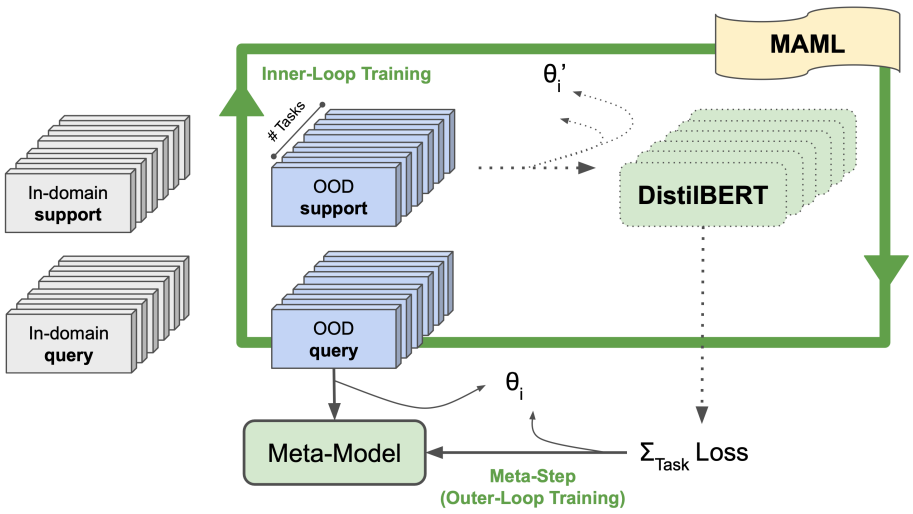

Figure 2: Model architecture of MAML DistilBERT. Training support and query sets can come from IND or OOD datasets and are a factor we experimented on.

5 Experiments

5.1 Data

For the default final project, we were provided with three IND and three OOD datasets. The IND datasets contain 50,000 question-passage-answer each. The OOD datasets are relatively small in size with 127 samples each. The domains covered by each of the datasets were specified in Table 1. The datasets were further split into either Train/Dev for model training or Train/Dev/Test sets for model evaluation. The Train dataset has only one human-provided answer per question, while the Dev and Test sets have three human-provided answers for each question.

Dataset	Question Source	Passage Source	Train	Dev	Test
IND datasets					
SQuAD [7]	Crowdsourced	Wikipedia	50,000	10,507	-
NewsQA [8]	Crowdsourced	News articles	50,000	4,212	-
Natural Questions [9]	Search logs	Wikipedia	50,000	12,836	-
oo-domain datasets (OOD)					
DuoRC [10]	Crowdsourced	Movie reviews	127	126	1248
RACE [11]	Teachers	Examinations	127	126	419
RelationExtraction [12]	Synthetic	Wikipedia	127	126	2,693

Table 1: Summary of datasets used in this project. Table borrowed from default project instruction - Training Datasets.

5.2 Evaluation method

During evaluation, we compared the performance our models against the baseline in the following two metrics (**EM** and **F1**). As there were three human-provided answers for each question, we took the maximum scores of both metrics:

- **Exact Match (EM)**: a binary measure (i.e. true/false) of whether the system output matches the ground truth answer exactly.
- **F1**: the harmonic mean of precision and recall.

5.3 Experimental details

If otherwise specified, batch size for all experiments were 16. To avoid GPU out-of-memory issue, data was loaded in either batch size of 1 or 4 to accumulate the loss. Model is updated at batch size of 16.

5.3.1 Experiment #1: MAML DistilBERT without FT Baseline

We experimented few model configuration as specified in Table 2. Models were trained with OOD set or a mixture of IND/OOD with support and query size of K in the inner-loop, and key experimentation factors and design of comparison were listed as below:

1. **K-shot:** MAML-20-d vs. MAML-2000-d
2. **Learning rate:** MAML-20-a vs. MAML-20-b vs. MAML-20-d
3. **Domain variability in training support:** MAML-20-b vs. MAML-20-c

Model	# Task	K-shot	Learning rate (α / β)	Training support	Training Time
MAML-20-a	10	20	1E-4	OOD	1.8hr
MAML-20-b	10	20	1E-5	OOD	2.4hr
MAML-20-c	10	20	1E-5	50% OOD + 50% IND	2.5hr
MAML-20-d	10	20	5E-5	OOD	2hr
MAML-2000-d	5	2000	5E-5	OOD	2hr

Table 2: **Experimentation #1** setup & model configuration

5.3.2 Experiment #2: Training MAML DistilBERT after FT Baseline

We experimented 10 MAML configuration (i.e. Model M1 - M10). Specifically, all the 10 models were trained from the pre-trained FT baseline model and trained with K-shot MAML from the

checkpoint. Specifically, we'd like to understand how the below factors influence the robustness of model performance across domains:

1. **K-shot:** M1/2/4 vs. M3, M7 vs. M8, M9 vs. M10
2. **IND or OOD for MAML training:** M1 vs. M6 vs. M7 vs. M10, M2 vs. M6 vs. M7 vs. M10
3. **Training time:** M1 vs. M2 vs. M4, M5 vs. M5

Model	K-shot	Learning rate	Inner-Loop /Meta-Step	Training Time
FT Baseline	-	3E-05	IND	3.5hr
M1	20	1E-05	OOD	7hr
M2	20	1E-05	OOD	3.5hr
M3	200	1E-05	OOD	7.5hr
M4	20	1E-05	OOD	9.5hr
M5	20	1E-05	OOD /IND val	6.8hr
M6	20	1E-05	OOD/ IND val	3.8hr
M7	20	1E-05	IND/ IND val	4.8hr
M8	200	1E-05	IND/ IND val	2.3hr
M9	200	1E-05	IND	2.3hr
M10	20	1E-05	IND	2.5hr

Table 3: **Experimentation #2** Model configuration

5.4 Results

5.4.1 Experiment #1: MAML DistilBERT without FT Baseline

We found that training MAML DistilBERT without FT Baseline couldn't achieve the same level of model performance as the FT Baseline. The best performing model configuration after a 2hr run was the **MAML-20-a** model, which was a 10-task 20-shot MAML model trained completely with OOD set with learning rates of 1E-4. This was unexpectedly lower than we thought, but it was an intuitive finding that MAML with small size of support set limited the learning of the task-learners.

Model	**EM** (OOD eval)	**F1** (OOD eval)	**EM** (IND eval)	**F1** (IND eval)
FT Baseline	34.55	49.88	54.54	70.31
MAML-20-a	**7.97**	**16.12**	**3.14**	**10.21**
MAML-20-b	0.34	9.04	1.31	9.12
MAML-20-c	0.62	9.17	1.57	9.45
MAML-20-d	3.77	13.22	2.88	10.62
MAML-2000-d	0	6.17	0.26	5.13

Table 4: **Experimentation #1** model performance

Contribution of K-shot. When comparing the MAML-20-d and MAML-2000-d, we found that the larger K the MAML was trained on did not necessarily lead to improving robustness of model performance on the OOD set.

Contribution of learning rates. When comparing the MAML-20-a, MAML-20-b and MAML-20-d, we found that the relationship between learning rate and MAML performance was not linear given the same K and under similar training time. In our case, the largest learning rate led to the best performing MAML in both EM and F1 in the OOD validation set.

Contribution of domain variability in training support. When comparing the MAML-20-b and MAML-20-c, we found that larger domain variability (i.e. 50% IND + 50%OOD) in support/query reached similar F1 performance but lower EM performance.

5.4.2 Experiment #2: Training MAML DistilBERT after FT Baseline

We found that training MAML after FT Baseline outperformed FT Baseline occasionally. The best performing model configuration was the **M2** model, which was a 10-task 20-shot MAML model trained and evaluated completely with OOD set with learning rates of 1E-5. The final scores among the OOD validation set were **EM: 35.60** and **F1: 50.49** after training time of 3.5hr. These were improvements of 1.22% in F1 and 3.04% in EM compared to the FT Baseline on the OOD validation set. However, the performance in IND validation set dropped by 4.57% in F1 and 6.49% in EM. This could be a sacrifice of model performance on the IND datasets in gaining additional robustness on an OOD dataset. The model performance scores on the RobustQA test Leaderboard were **EM: 40.436** and **F1: 50.49**.

Contribution of K-shot. When comparing M1 and M3; M9 and M10, we found that the larger K the MAML was trained on did not necessarily lead to improving the robustness of model performance on the OOD set. The model performance metrics were pretty close to each other in a 20-shot and a 200-shot setting when running similar training time.

Contribution of IND or OOD for MAML training. When comparing M2, M6, M7, and M10, we observed that the model performances in IND validation set were improved if IND set was involved in the training task. When IND validation set was used in meta-step training (e.g. M6 and M8), the model performance in the IND set got further increased.

Contribution of training time. We found training time, after a certain period, could reduce the model performance in both OOD and IND validation sets. This observation held true in the comparisons of M1, M2, M4 and M5 vs. M6.

Model	EM (OOD eval)	F1 (OOD eval)	EM (IND eval)	F1 (IND eval)
FT Baseline	34.55	49.88	54.54	70.31
M1	34.03	49.35	52.9	68.59
M2	**35.60**	**50.49**	**51**	**67.1**
M3	34.82	49.3	53.58	69.24
M4	33.77	49.2	52.82	68.41
M5	32.98	48.83	53.02	68.65
M6	33.25	48.38	55.14	70.7
M7	34.03	49.74	55.3	70.91
M8	**34.55**	**50.1**	**55.14**	**70.86**
M9	34.29	49.61	55.15	70.74
M10	34.55	49.65	55.08	70.78

Table 5: **Experimentation #2** model performance

6 Analysis

6.0.1 Impact of K-shot.

Even-though we observed that increasing in K in few-shot learning in our MAML DistilBERT didn't contribute much model performance improvements, we did not think the observation always holds true. Instead, we thought the reaon of this observation could be the experimented values in K were set to be too large compared to the size of the OOD set and resulting in a task pool of "bootstrapping" sample of size K larger than the original sample size. To further evaluate this hypothesis, more configurations with smaller K should be explore in an Experiment #1 setting and we would suggest this as a potential research next step.

6.0.2 Impact of learning rate.

It was within our expectation that the relationship of learning rate and model performance was not linear. We thought the reason that larger learning rate led to better model performance with similar training time in this project was associated with the more aggressive step in gradient descent when only a few samples available, which might not always be a valid setup. As a result, tuning learning rate as a hyper-parameter could be an ideal next step to explore if given more time. In addition, we assumed the same learning rate for both inner-loop and meta-step gradient descent updates; which when giving more flexibility in configuration, might work better.

6.0.3 Impact of IND/OOD training sets.

When the training tasks were from a mixture of IND and OOD sets, it was intuitive that the MAML performance reached a similar F1 score while a lower EM score. As the learning goal of the MAML is to learn the initial parameter setup well so that the model adapts faster with small number of new tasks, I'd expect the learning of multiple domains simultaneously benefits the model to understand synergies across domains but leading to its more "general" and "robust" learning. This could mean a slower-down of the gradient of gradient learning during meta-step update. As the aggregation of gradient loss from the inner-loop were from a much more "diverse" pool. However, I'd expect after enough initial training time on all domains, this setup of the task pool would contribute to better performance than the single/less diverse domain task pool.

Slightly different from training the MAML from scratch, when training MAML from a pre-trained FT model checkpoint, we observed larger improvement of model performance robustness on OOD validation set when the training tasks were from OOD set. This could because the pre-training model already tuned the model parameters well enough with all the IND set. To further improve the model parameter, loss from learning the new domains would come with larger gradient for MAML to optimize.

7 Conclusion

MAML was a good-to-explore to achieve cross-domain model robustness. The main contribution of this study was to implement a MAML DistillBERT QA system and conducted lots of experiments around domain variability in task pool, MAML with or without model pre-training and fine-tuning, etc. We found that MAML might not be the best framework in context of a large amount IND set and small amount OOD set as the FT baseline model already learned well in the large set of IND data. Training MAML post baseline model pre-training and fine-tuning performed occasionally better than the FT baseline model likely due to additional random OOD tasks used to learn by the MAML model. However, more investigations in model configuration hyper-parameter tuning are still in need to validate if this always holds true.

References

[1] Pieter Abbeel Finn, Chelsea and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning. PMLR*, 2018.

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

[3] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[4] Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. *CoRR*, abs/1606.04474, 2016.

[5] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml, 2019.

[6] Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. *CoRR*, abs/1909.04630, 2019.

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

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

[9] 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.

[10] Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. *CoRR*, abs/1804.07927, 2018.

[11] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[12] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *CoRR*, abs/1706.04115, 2017.