Meta-Learning for Question Answering on SQuAD 2.0

Chi-Hsuan Chang [1]

[1]Stanford University

Problem

In the context of a Question Answering (QA) system, we train the system to take a question and a paragraph, and learn to extract an answer to such question from the given paragraph. Often time, limited amount of text data is available for the model to learn to optimize a new task. In this study, we aimed to build a robust QA system with meta-learning that is robust to domain shifts using SQuAD 2.0 dataset.

Background

SQuAD 2.0 dataset

Three in-domain (SQuAD, NewsQA, Natural Questions) and three out-of-domain (DuoRC, RACE, RelationExtraction) datasets. The in-domain (IND) and out-of-domain (OOD) datasets contain 50K and 127 question-passage-answer samples each.

Model-Agnostic Meta-Learning (MAML)

MAML was originally proposed by Finn et al 2017 [2] to train the models their own initial parameters so that the parameters allow the algorithm to perform well on a new task ("learn-to-learn") after one or a few gradient steps of updates with few-shot data availability.

Methods

FT Baseline

A fine-tuned (FT) pre-trained transformer model - DistilBERT [3]. The baseline QA model was trained on the overall IND training set, and was validated on the IND validation set.

MAML DistilBERT

We adapted MAML[2] as a framework to train our robust QA system that performs well across different domains.

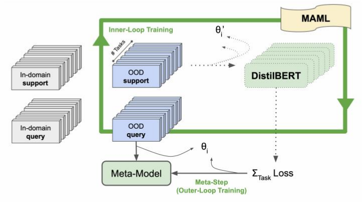

Figure 1. Model architecture of MAML DistilBERT. Training support and query sets can come from In-domain or OOD datasets and are a factor we experimented on

- We defined the baseline DistilBERT [3] as our base learner (f_θ)
- We implemented a task method rather than to pre-define a K-shot task pool ($p(\mathcal{T})$). As K sample support ($\mathcal{D}_i$) and query ($\mathcal{D}_i'$) sets can come from IND and OOD training datasets in different experiments
- We used the same loss function ($\mathcal{L}$, loss $= -\log p_{start}(i) - \log p_{end}(j)$) as the baseline

FT Baseline + MAML DistilBERT

In addition to training MAML model from scratch, we also leveraged the FT DistillBERT (Baseline) model and trained the MAML models from the FT checkpoint.

Experiments

If not 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.

How each of these factors influence model performance after?

Experiment #1: MAML DistilBERT without FT Baseline

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	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 1. Experiment 1: Model configuration

Experiment #2: Training MAML after FT Baseline

If not otherwise specified, the meta-step update used the aggregate query sets from each of the task.

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. M5 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 2. Experiment 2: Model configuration

Analysis

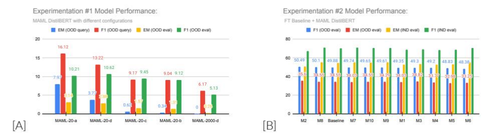

Figure 2. Experiment #1 & #2 model descending sorted by EM (OOD eval)

Key-takeaway #1: MAML DistilBERT without FT Baseline couldn't achieve the same level of model performance as the FT Baseline.

- This can be because of the large IND data available during baseline model pre-training/fine-tuning.
- Larger learning rate helped in faster adaptation with the MAML model given the same sample size as it allowed more aggressive exploration in the gradient at the beginning.
- Larger domain variability in support/query reached similar F1 performance but lower EM performance. This was intuitive as the MAML was learning to learn and exposed to a lot of topics as few-shot learning though benefit understanding synergies across domains, the model also became more "general" and "robust".

Key-takeaway #2: Training MAML after FT Baseline outperformed FT Baseline occasionally. More experimentation configurations in learning rate and domain variability could be explored.

- **M2**, a 10-task 20-shot MAML training on OOD samples post pre-training outperformed the FT Baseline in OOD validation set by 1.22% in F1 and 3.04% in EM. Its performance in IND validation set dropped by **4.57%** in F1 and **6.49%**in EM. This showed the scarification of model performance on the IND datasets in gaining additional robustness on an OOD dataset.
- **M8**, a 10-task 200-shot MAML training on IND samples post pre-training outperformed the FT Baseline in OOD validation set by 0.44% in F1 and 0% in EM, and in IND validation set by **0.78%** in F1 and **1.10%** in EM. This showed that continuously training with the same domain datasets with MAML contributed less improvements than training with few OOD samples.

Conclusions

MAML was a good-to-explore to achieve cross-domain model robustness. MAML might not be the best framework in context of a large amount IND set and small amount OOD set. Training MAML post baseline model pre-training and fine-tuning performed occasionally better than the FT baseline model likely due to additional OOD tasks used to learn by the MAML model.

References

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

[2] 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.

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