FAME-BERT: Stable Meta-learning for Robust Question-Answering

Bharat Khandelwal, Shubham Anand Jain

CS224N Winter 2022


Problem Statement

We work on building a **Robust Question Answering system** that can generalise to out-of-domain datasets with a small number of examples.

Input: Paragraph, Question about the paragraph
Output: Span of text from the paragraph

> **Question:** Why was Tesla returned to Gospic?
> **Context paragraph:** On 24 March 1879, Tesla was returned to Gospic under police guard for not having a residence permit. On 17 April 1879, Milutin Tesla died at the age of 60 after contracting an unspecified illness (although some sources say that he died of a stroke).
> **Answer:** not having a residence permit

Fig 1. Example Prompt

We have 3 in-domain datasets and 3 out-of-domain datasets, where the OOD datasets are used for evaluation.

Dataset	Question Source	Passage Source	Train	dev	Test
in-domain datasets					
SQuAD [5]	Crowdsourced	Wikipedia	50000	10,507	-
NewsQA [7]	Crowdsourced	News articles	50000	4,212	-
Natural Questions [6]	Search logs	Wikipedia	50000	12,836	-
oo-domain datasets					
DuoRC [9]	Crowdsourced	Movie reviews	127	126	1248
RACE [10]	Teachers	Examinations	127	128	419
RelationExtraction [11]	Synthetic	Wikipedia	127	128	2693

Fig 2. Dataset details

Method

We use **Model Agnostic Meta Learning (MAML)** **[1],** which is an algorithm that trains a model to **"learn how to learn"**.

We learn an effective representation of parameters θ that performs well on new tasks given few-shot training.

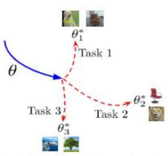

Fig 3. Illustration of Meta-learning

We experiment with various methods to make meta-learning more stable, which leads to the name **FAME-BERT** (Finetune-Augment-Metalearn-Ensemble DistilBERT).

Training Pipeline

Our Training pipeline consists of three parts:

1. **Initial training**: Training on in-domain datasets, done using a pre-trained BERT, Metalearning, or retraining of another model
2. **Fine-tuning**: Training on the OOD train datasets, possibly augmented, tuning LR
3. **Ensembling**: Taking multiple seeds and combining the predictions using voting

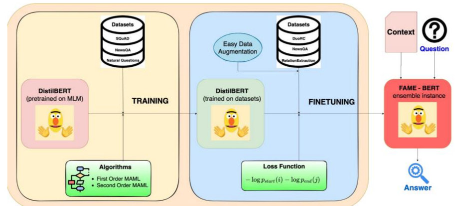

Fig 4. Training Pipeline

Experiments

We evaluate our models based on their EM and F1 scores, which are defined as follows:

- EM Score is a binary measure of whether the answer is correct -> intuition: Is this exactly the actual answer?
- F1 Score is defined as 2 x precision x recall / (precision + recall) -> intuition: How close is the actual answer?

We demonstrate our methods on 4 candidate baseline models, with descriptions as follows:

Model Name	Model Description	Train epochs	Train time (hours)
$\mathcal{M}_1$	DistilBERT baseline, no finetuning	3	4
$\mathcal{M}_2$	DistilBERT baseline	10	13
$\mathcal{M}_3$	First-order MAML	10	17
$\mathcal{M}_4$	Second-order MAML on $\mathcal{M}_2$	1	5

Fig 5. Model Descriptions

We notice that different datasets require different learning rates due to **diverse underlying data characteristics**, as demonstrated in Figure [6].

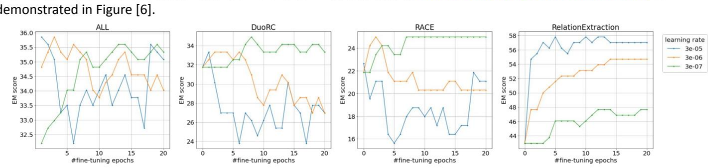

Fig 6. Effect of Learning Rate during fine-tuning across datasets

Choosing the best learning rates for the models, we note that the models compare as given in Figure [7]. Figure [8] illustrates that the model performance **varies wildly depending on the dataset**.

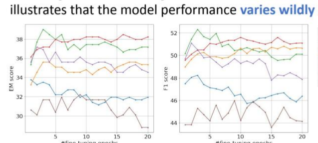

Fig 7. Models v/s number of epochs
a) EM score b) F1 score

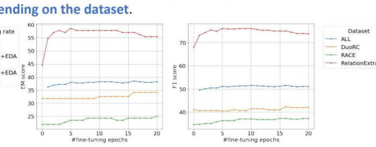

Fig 8. Model Disparity b/w Datasets
a) EM score b) F1 score

Results & Analysis

Based on our preliminary experiments, we had the following findings & followed up on them:

- We noted that our fine-tuning results were unstable, and therefore we **tuned learning rates separately** for each dataset, which helped increase stability significantly, as observed in Figures [6a, 6b]
- We noted **significant variance in predictions** across multiple seeds for the same model (Figure [9]), so we **ensembled** across seeds. This was observed to boost F1 scores, as seen in Figure [10]
- We performed data augmentation [3] to **alleviate the lack of training data** on out of domain sets; the advantage can be seen in Figures [7a, 7b]
- After looking at initial instability of meta learned models, we proposed M4, where we used **2nd order MAML** on top of a good pre-trained model based on ideas from the paper "How to train your MAML" [2]

Times predicted	% occurrences
[5]	88.2
[4, 1]	7.1
[3, 2]	3.9
[3, 1, 1]	0.3
[2, 2, 1]	0.5

Fig 9. Predictions by ensemble models

Model	F1	EM
Average of 5 Seeds	51.11 ± 0.24	38.74 ± 0.29
Ensemble (Majority Vote)	**52.073**	**38.743**

Fig 10. Benefit of Ensembling across seeds

On the dev and validation sets, our final results are as follows:

Model Name	EDA	Dev F1	Dev EM	Test F1	Test EM
$\mathcal{M}_1$	No	46.512	31.675	59.187*	40.28*
$\mathcal{M}_2$	No	51.995	37.173	-	-
$\mathcal{M}_2$	Yes	**53.020**	**39.791**	**59.679**	42.156
$\mathcal{M}_3$	No	52.128	39.529	59.347	**42.431**
$\mathcal{M}_3$	Yes	51.276	37.958	-	-
$\mathcal{M}_4$	No	46.736	33.770	-	-
Ensemble($\mathcal{M}_2, \mathcal{M}_3$)	Yes	**53.065**	**40.314**	**60.042**	**42.959**
Leaderboard rank	-	6	1	11	5

Fig 11. Final dev & test set results

Future Work

Due to high training costs, we were unable to find good hyperparameters for EDA and SO-MAML on training. We also expect that the following methods will help performance:

- Using **OOD train data** in the training step
- Incorporating **importance sampling**, even in just finetuning

References

1. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", Chelsea Finn, Pieter Abbeel, and Sergey Levine
2. "How to train your MAML", Antreas Antoniou, Harrison Edwards and Amos Storkey
3. "EDA: Easy data augmentation techniques for boosting performance on text classification tasks", Jason Wei and Kai Zou