Leveraging Meta-Learning Algorithms for Robust Question Answering

Stanford CS224N Default Robust Project

Zhiyin Lin
Department of Computer Science
Stanford University
zhiyinl@stanford.edu

Beining (Cathy) Zhou
Department of Computer Science
Stanford University
cathyzbn@stanford.edu

1 Key Information to include

- External collaborators (if you have any): N/A.
- Mentor (custom project only): N/A.
- Sharing project: N/A.

2 Research paper summary (max 2 pages)

Title	Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks
Author	Zi-Yi Dou, Keyi Yu, Antonios Anastasopoulos
Venue	Conference on Empirical Methods in Natural Language Processing (EMNLP)
Year	2019
URL	https://arxiv.org/abs/1908.10423

Table 1: Table for bibliographical information [1].

Background. Through the development of Natural Language Understanding, many models such as BERT are able learning the representation of the language and produces models that performs well in the in-distribution tasks. However, when these models are applied to low-resource tasks where less data are available, their performance decreases significantly. The paper above attempts to address this problem by increasing the robustness of the model and making the language representations more flexible. They aim to learn representations that are shared among different tasks, so that the model could adapt to a new task in a different domain quickly with little training.

Previously, researchers have attempted to solve this problem by multi-task learning. This approach leverages the domain-specific information across related tasks by using parameter sharing. However, multi-task learning requires large amount of data and fine-tuning. This paper proposes to use the alternative of meta-learning algorithms, which finds good initialization for the model, allowing fine-tuning on the model to require less data.

Summary of contributions. The contributions of this paper are three-fold:

- It provides an experimental comparison of model-agnostic meta-learning algoritm and its variants including MAML, FOMAML, and Reptile.
- It evaluates on design choices on tasks distribution selection and hyperparameters for meta-gradients.
- Most relevantly, it shows empirical evidence that the model could be adapted to out-of-distribution NLU tasks and outperform strong baseline models such as BERT and MT-DNN.

Meta-learning algorithms have performed well on various tasks such as computer vision and have received significant research attention because of the improved performance. However, the exploration for applications of meta-learning algorithms on natural language understanding tasks have been limited. This paper provides a first probe to using MAML-based algorithms, especially Reptile, on natural language understanding. This paper is the precedent to test out such experimentation.

The three models that the paper used are based on the model-agnostic meta learning algorithm (MAML). Given a set of tasks $\{T_1, \ldots, T_k\}$ and a target task $\{T_t\}$, the model meta-learns a (hopefully close to optimal) initialization of parameters from $T_1, \ldots T_k$ and fine-tune the its parameters on the (often low-resourced) target task T_t. The key idea in MAML is the optimization-based search for a good initialization using two gradient descents: one in the inner loop to evaluate how much the model descent given a provided initialization, based on which (using a gradient descent in the outer loop) to update this provided initialization to a better one. With a good initialization, the model can learn new tasks with few data.

In mathematical terms, in the step of meta-learn, the model updates the parameters of the auxiliary tasks with $k \leq 1$ number of gradient descent steps:

$$\theta_i^{(k)} = \theta_i^{(k-1)} - \alpha \Delta_{\theta_i^{(k-1)}} L_i(f_{\theta_i^{(k-1)}})$$

where α is the learning rate, and L_i is the loss function for task T_i. While performing the gradient descent, the algorithms samples the tasks with a distribution $p(T)$. AFter updating the parameters for the individual auxiliary tasks, the model then performs Meta-Update on θ, the parameter of the target task T_t. The algorithms repeatedly performs the updates until the optimization completes.

FOMAML and Reptile are first-order variants of MAML. They simplified the computations by discarding second derivatives, with Reptile making additional simplification by taking multiple gradient descents steps to capture a general direction before updating its weights. Despite simplified, their empirical performances are of similar level as MAML.

The paper has made several interesting design choices. One is on the effects of hyperparameters, specifically the number of update steps and the inner learning rate in Reptile. The paper has found the most successful number of update steps k is around 5 across tasks, and comments that in cases where k = 1, it makes the algorithm close to joint training; but when k is too large, the resulting gradient can go too far and deviate, becoming uninformative.

As all three algorithms require training across tasks, the authors propose three options for task sampling at train time: uniform, probability proportional to size (PPS), and mixed. It is worthnoticing of the PPS idea, which has the probability of selecting a task proportional to the size of its dataset. Conceptually, this prevents overfitting tasks of smaller sizes while underfitting the bigger-sized tasks, allowing fair contribution by each data. The authors empirically proves that this approach gives best performances out of the three.

This paper evaluates the model on the GLUE dataset. It uses four high-resource tasks, namely SST-2, QQP, MNLI, and QNLI, as auxiliary task and four low-resource tasks, CoLA, MRPC, STS-B, and RTE as target tasks for evaluation. It also tests the generalizability of the model on the SciTail dataset, which is not a part of GLUE. It compared the trained models to two strong baselines, the BERT model, which pre-trains the Transformer model on large datasets, and the MT-DNN model with PPS sampling which was the state-of-the-art multitasking model.

At test time performance, Reptile has a tiny edge over FOMAML and MAML, though all of which outperform strong baselines BERT and MT-DNN. One interesting plot the authors include is on the OOD target task SciTail, which the model does not come across during meta-learning. When percentage of fine-tune data remains as low as 10^{-3}, Reptile performance remains above 80% accuracy while BERT only has 50%, though BERT's performance catches up when the percentage of training data goes to 100%. This shows exactly that at the NLU scenarios when few training samples are available, meta-learning algorithms are good options to go with.

Limitations and discussion. This paper poses limitations in the fact that it only evaluates the model on the GLUE and SciTail datasets. These datasets contains only English language and do not account for the performance for other languages which have different structure and forms of representation. In GLUE, although there is a distinction between high-resource and low-resource tasks, the low-resource tasks for testing out-of-distribution still contained a significant amount of data

of around 5,000. The researchers did not evaluate their model on tasks with even less data, say of size 100. It would be interesting to see the comparison of meta-learning methods and the baseline for few-shot learning instances. Furthermore, the distribution shift among the tasks in GLUE are different than our project's distribution shift — GLUE examines the shift among different natural language understanding tasks, whereas our project will examine that among different context of the question-answering task. This might pose new challenges to the model.

We also hoped that this model elaborated more on the fine-tuning techniques that it used.

In addition, meta learning has intrinsic limitations. Although the results showed improvements, the increase in accuracy was not significant as one would hope. We hypothesize this is due to the fact that meta learning only proposes an initialization for the model, which would be updated in future training and fine-tuning. We wished that the models similar to the LSTM could help the model to partially update the initialization and retain more information from the meta learning step. We also wonder if the initialization would cause a more significant improvement if the target task dataset was smaller and if the model undergoes less updates.

Another limitation of meta learning is that although it might have learned from different auxiliary task how to adapt to a new task, it did not fully explain the centerpiece of this puzzle — which are the shared representations among these similar tasks of natural language understanding, and which are representations specific to each individual tasks? These are some further questions to dig into.

Why this paper? Despite these limitations, this paper presents exciting empirical results that we believe could be applied to wider contexts. Applying meta learning algorithms for out-of-distribution or few shot predictions has gain popularity over the past years. The idea of learning to learn, and particularly the idea of MAML where we optimize an initialization weights, are neat and promising for general robustness. It is naturally promising to consider it for OOD and robust NLP. After reading it in-depth, we gained partial empirical insights in applying meta-learning for robust NLP, though stronger and more variations of the current empirical results would be appreciated.

Wider research context. Meta learning is an approach that could foster development in many NLP tasks. As foundational models become more prevalent, fine-tuning them on specific low-resource tasks emerges as a significant focus. Meta learning provides a good initialization for fine-tuning, which is an invaluable asset given the small dataset. Specifically, since language spans across various domains and contexts, large-scaled and well-annotated data is hard to obtain for each context. Many niche languages have few translation sources and data available. Meta learning is applicable to these contexts.

This paper also probes the general problem of robustness and distribution shift. In all cases of machine learning and deep learning, the real world data is different from the training and test data. It explores the problem of discovering the shared word representation in NLP and the shared features in any ML/DL tasks. It helps use to understand the nature of these tasks and the commonalities and differences among them. Therefore, we find research in meta learning and few-shot learning intriguing.

3 Project description (1-2 pages)

Goal. We wish to develop a language model that is robust to out-of-distribution data, meaning that it could generalize beyond the distribution of the training dataset into other scenarios. Specifically, we will look at the question-answering task with in the Stanford Question Answering Dataset (SQuAD). We wish to explore how meta-learning algorithms, including MAML, FOMAML, and Reptile. We will assess how these models perform on the Robust QA task and find the best hyperparameters for the model on this task. Then, we will explore data augmentation and fine-tuning techniques to improve our model. We quantify the improvements by the EM and F1 score.

To improve on what the chosen paper has done, we will fine-tune the model to allow the initialization of the meta learning approach to retain more information through fine-tuning. This idea originates from the observation that the initialization is quickly lost by future training. This improvement will be done by a mix-out approach proposed by Lee et al. where at each training iteration, individual model parameters randomly reverse back to the pre-train initialization.[2] We think this is a promising approach especially when combined with meta learning.

Task. The task of question answering is when a model inputs a paragraph and a question answered in the paragraph, and is asked to answer the question correctly. How well the model answers provides insights into how well model understand the text. In particular, we aim to work on a question answering system robust to domain shifts, so it adapts to out-of distribution domains with little training. Note that the context paragraph could be derived from many sources, including Wikipedia, news articles, and movie reviews, and the model aims to perform well across all those distributions.

Data. We use three in-domain reading comprehension dataset: Natural Questions [3] (Train size 50000, dev size 12836), NewsQA [4] (Train size 50000, dev size 4212), and SQuAD [5] (Train size 50000, dev size 10507). We evaluate our model on three out-of-domain datasets: Relation Extraction [6] (Train size 127, dev size 128, test 2693), DuoRC [7] (Train size 127, dev size 126, test 1248), and RACE [8] (Train size 127, dev size 128, test 419). More specific information are covered in the handout.

All datasets are processed into the same format as SQuAD. Specifically, given a question paragraph pair, we implement chunking to convert each paragraph into multiple chunks of size 384 with a stride of 128 (see more specifics in the handout). We follow it by caching to avoid repeated time consuming computations.

Methods. We plan to apply meta learning algorithms including MAML [9], FOMAML [9], and Reptile [10] on the Robust QA task, and we will compare their performance. We will perform experiments on the above datasets, and we will find the best hyperparameters in meta learning, including the probability distribution applied to auxiliary tasks, the number of updates performed, and the learning rate.

In addition, we plan to use fine-tuning and data augmentation methods to further improve the performance of our model.

We will use several data augmentation methods.

- First, we will paraphrase the context and query by back-translation. We will train a model using the opensource TensorFlow NMT with German as the pivot language. We will translate the query directly and pass in the context source in segment sentences. Finally, we retrieve the target answer by string matching and heuristics.[11]
- Second, we will use Easy Data Augmentation (EDA) techniques, including replacing context words with synonyms, inserting a random synonym of a random word into a random position in the sentence, randomly swapping words in a sentence, and randomly deleting words in a sentence. [12]
- We will also place masks in the sentence and allow BERT to fill in the mask with a generated word. [13] These augmentation models would ideally make our model more robust.

We plan to experiment with a few BERT fine-tuning techniques summarized by Zhang et al. [14].

- Initialization for fine tuning: (1) initialize all layers except one specialized output layers with pretrained weights as baseline; (2) re-initialize the pooler layers and the top L = 5 layers (L can vary, start with L = 5) layers closer to the output, motivated by that layers closer to the outputs specialize more on specific tasks and layers farther from the outputs focus on more generalized features [14].
- Mixout: similar to dropout which randomly knocks out neurons to 0 at train time, this replaces model parameters with its pre-trained value with probability p [2].
- Layer-wise Learning Rate Decay: one applies a higher learning rate to layers closer to the output layer and lower learning rate to layers farther from the output layer. This is designed under the intuition that, again, layers closer to the output layer encode more specific information so it shall be quicker in adapting, corresponding to a higher learning rate [15]

Baselines. As indicated in the handout, we will implement the baseline off from the starter code provided. The baseline is a fine-tuned version of the transformer model DistilBERT [16], a smaller, distilled version of the original BERT. Loss is calculated as the sum of the cross-entropy loss for the

start and end locations (for example, for a single example loss = -log$p_{start}(i)$− -log$p_{end}(j)$), and it is averaged across the batch during training. The model uses the AdamW optimizer [17]. There is an hyperparameter at train time to bound the maximum length of a predicted answer (set to 15 as default).

Evaluation. We will use two evaluation metrics: the Exact Match (EM) score and the F1-score. At evaluation time, we take the maximum across all three human-provided answers to the question. Finally, the final score takes the average of the EM and F1 scores across test samples.

Since we are implementing our own baseline, we plan to compare our score with the baseline score. We aim to fairly outperform the baseline.

References

[1] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

[2] Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. *arXiv preprint arXiv:1909.11299*, 2019.

[3] 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.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[10] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[11] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. *arXiv preprint arXiv:1912.02145*, 2019.

[12] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[13] Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*, 2020.

[14] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*, 2020.

[15] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[16] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.