Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks

Zi-Yi Dou, Keyi Yu, Antonios Anastasopoulos
Language Technologies Institute, Carnegie Mellon University
{zdou, keyiy, aanastas}@cs.cmu.edu

Abstract

Learning general representations of text is a fundamental problem for many natural language understanding (NLU) tasks. Previously, researchers have proposed to use language model pre-training and multi-task learning to learn robust representations. However, these methods can achieve sub-optimal performance in low-resource scenarios. Inspired by the recent success of optimization-based meta-learning algorithms, in this paper, we explore the model-agnostic meta-learning algorithm (MAML) and its variants for low-resource NLU tasks. We validate our methods on the GLUE benchmark and show that our proposed models can outperform several strong baselines. We further empirically demonstrate that the learned representations can be adapted to new tasks efficiently and effectively.

1 Introduction

With the ability to learn rich distributed representations of data in an end-to-end fashion, deep neural networks have achieved the state of the arts in a variety of fields (He et al., 2017; Vaswani et al., 2017; Povey et al., 2018; Yu et al., 2018). For natural language understanding (NLU) tasks, robust and flexible language representations can be adapted to new tasks or domains efficiently. Aiming at learning representations that are not exclusively tailored to any specific tasks or domains, researchers have proposed several ways to learn general language representations.

Recently, there is a trend of learning universal language representations via language model pre-training (Dai and Le, 2015; Peters et al., 2018; Radford et al., 2018). In particular, Devlin et al. (2019) present the BERT model which is based on a bidirectional Transformer (Vaswani et al., 2017). BERT is pre-trained with both masked language model and next sentence prediction objectives and exhibits strong performance on several benchmarks, attracting huge attention from researchers. Another line of research tries to apply multi-task learning to representation learning (Liu et al., 2015; Luong et al., 2015). Multi-task learning allows the model to leverage supervision signals from related tasks and prevents the model from overfitting to a single task. By combining the strengths of both language model pre-training and multi-task learning, Liu et al. (2019) improve the BERT model with multi-task learning and their proposed MT-DNN model successfully achieves state-of-the-art results on several NLU tasks.

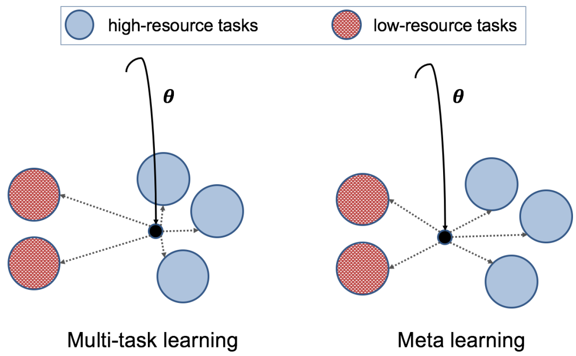

Figure 1: Differences between multi-task learning and meta learning. Multi-task learning may favor high-resource tasks over low-resource ones while meta-learning aims at learning a good initialization that can be adapted to any task with minimal training samples. The figure is adapted from Gu et al. (2018).

Although multi-task learning can achieve promising performance, there still exist some potential problems. As shown in Figure 1, multi-task learning may favor tasks with significantly larger amounts of data than others. Liu et al. (2019) alleviate this problem by adding an additional fine-tuning stage after multi-task learning. In this paper, we propose to apply meta-learning algorithms in general language represen-

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1192–1197,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

tations learning. Meta-learning algorithms aim at learning good initializations that can be useful for fine-tuning on various tasks with minimal training data, which makes them appealing alternatives to multi-task learning. Specifically, we investigate the recently proposed model-agnostic meta-learning algorithm (MAML) (Finn et al., 2017) and its variants, namely first-order MAML and Reptile (Nichol et al., 2018), for NLU tasks.

We evaluate the effectiveness and generalization ability of the proposed approaches on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019). Experimental results demonstrate that our approaches successfully outperform strong baseline models on the four low-resource tasks. In addition, we test generalization capacity of the models by fine-tuning them on a new task, and the results reveal that the representations learned by our models can be adapted to new tasks more effectively compared with baseline models.

2 Proposed Approaches

In this section, we first briefly introduce some key ideas of meta learning, and then illustrate how we apply meta-learning algorithms in language representations learning.

2.1 Background: Meta Learning

Meta-learning, or learning-to-learn, has recently attracted researchers' interests in the machine learning community (Lake et al., 2015). The goal of meta-learning algorithms is to allow fast adaptation on new training data. In this paper, we mainly focus on optimization-based meta-learning algorithms, which achieve the goal by adjusting the optimization algorithm. Specifically, we investigate MAML, one of the most representative algorithms in this category, and its variants for NLU tasks.

MAML and its variants offer a way to learn from a distribution of tasks and adapt to target tasks using few samples. Formally, given a set of tasks $\{T_1, \cdots, T_k\}$, the process of learning model parameters θ can be understood as (Gu et al., 2018):

$$\theta_t^* = \text{Learn}(T_t; \text{MetaLearn}(T_1, \cdots, T_k)),$$

where T_t is the target task.

Hopefully, by exposing models to a variety of tasks, the models can learn new tasks with few steps and minimal amounts of data.

2.2 General Framework

In this part, we introduce the general framework of the MAML approach and its variants, including first-order MAML and Reptile.

Algorithm 1 Training procedure.

Pre-train model parameters θ with unlabeled datasets.
while not done **do**
  Sample batch of tasks $\{T_i\} \sim p(T)$
  for all T_i **do**
    Compute $\theta_i^{(k)}$ with Eqn. 1.
  end for
  Update θ with Eqn. 2.
end while
Fine-tune θ on the target task.

We first describe the meta-learning stage. Suppose we are given a model f_θ with parameters θ and a task distribution $p(T)$ over a set of tasks $\{T_1, T_2, \cdots, T_k\}$, at each step during the meta-learning stage, we first sample a batch of tasks $\{T_i\} \sim p(T)$, and then update the model parameters by k ($k \geq 1$) gradient descent steps for each task T_i according to the equation:

$$\theta_i^{(k)} = \theta_i^{(k-1)} - \alpha \nabla_{\theta_i^{(k-1)}} L_i(f_{\theta_i^{(k-1)}}), \quad (1)$$

where L_i is the loss function for T_i and α is a hyper-parameter.

The model parameters θ are then updated by:

$$\theta = \text{MetaUpdate}(\theta; \{\theta_i^{(k)}\}). \quad (2)$$

We would illustrate the *MetaUpdate* step in the following part. It should be noted that the data used for the MetaUpdate step (Eqn. 2) is different from that used for the first k gradient descent steps (Eqn. 1).

The overall training procedure is shown in Algorithm 1. Basically, the algorithm consists of three stages: the pre-training stage as in BERT, the meta-learning stage and the fine-tuning stage.

2.3 The MetaUpdate Step

As demonstrated in the previous paragraph, MetaUpdate is an important step in the meta-learning stage. In this paper, we investigate three ways to perform MetaUpdate as described in the following parts.

MAML The vanilla MAML algorithm (Finn et al., 2017) updates the model with the meta-objective function:

$$\min_{\theta} \sum_{T_i \sim p(T)} L_i(f_{\theta_i^{(k)}})$$

Therefore, MAML would implement the MetaUpdate step by updating θ according to:

$$\theta = \theta - \beta \sum_{T_i \sim p(T)} \nabla_{\theta} L_i(f_{\theta_i^{(k)}}),$$

where β is a hyper-parameter.

First-Order MAML Suppose $\theta^{(k)}$ is obtained by performing k inner gradient steps starting from the initial parameter $\theta^{(0)}$, we can deduce that:

$$\nabla_{\theta^{(0)}} L(f_{\theta^{(k)}}) = \nabla_{\theta^{(k)}} L(f_{\theta^{(k)}}) \prod_{i=1}^{k} \nabla_{\theta^{(i-1)}} \theta^{(i)}$$
$$= \nabla_{\theta^{(k)}} L(f_{\theta^{(k)}}) \prod_{i=1}^{k} (I - \alpha \nabla^2_{\theta^{(i-1)}} L(f_{\theta^{(i-1)}})).$$

Therefore, MAML requires calculating second derivatives, which can be both computationally and memory intensive. First-Order MAML (FOMAML) ignores the second derivative part and implement the MetaUpdate as:

$$\theta = \theta - \beta \sum_{T_i \sim p(T)} \nabla_{\theta_i^{(k)}} L_i(\theta_i^{(k)}).$$

Reptile Reptile (Nichol et al., 2018) is another first-order gradient-based meta-learning algorithm that is similar to joint training, as it implements the MetaUpdate step as:

$$\theta = \theta + \beta \frac{1}{|\{T_i\}|} \sum_{T_i \sim p(T)} (\theta_i^{(k)} - \theta).$$

Basically, Reptile moves the model weights towards new parameters obtained by multiple gradient descent steps. Despite the simplicity of Reptile, it has been demonstrated to achieve competitive or superior performance compared to MAML.

2.4 Choosing the Task Distributions

We experiment with three different choices of the task distribution $p(T)$. Specifically, we propose the following options:

- **Uniform**: sample tasks uniformly.

Model	Test Dataset			
	CoLA	MRPC	STS-B	RTE
BERT	52.1	88.9/84.8	87.1/85.8	66.4
MT-DNN	51.7	89.9/86.3	87.6/86.8	75.4
MAML	**53.4**	89.5/85.8	88.0/87.3	76.4
FOMAML	51.6	89.9/86.4	88.6/88.0	74.1
Reptile	53.2	**90.2/86.7**	**88.7/88.1**	**77.0**

Table 1: Results on GLUE test sets. Metrics differ per task (explained in Appendix A) but the best result is **highlighted**.

- **Probability Proportional to Size (PPS)**: the probability of selecting a task is proportional to the size of its dataset.

- **Mixed**: at each epoch, we first sample tasks uniformly and then exclusively select the target task.

3 Experiments

We conduct experiments on the GLUE dataset (Wang et al., 2019) and only on English. Following previous work (Devlin et al., 2019; Liu et al., 2019) we do not train or test models on the WNLI dataset (Levesque et al., 2012). We treat the four high-resource tasks, namely SST-2 (Socher et al., 2013), QQP,[1] MNLI (Williams et al., 2018), and QNLI (Rajpurkar et al., 2016), as auxiliary tasks. The other four tasks, namely CoLA (Warstadt et al., 2018), MRPC (Dolan and Brockett, 2005), STS-B (Cera et al., 2017), and RTE (Dagan et al., 2005) are our target tasks. We also evaluate the generalization ability of our approaches on the SciTail dataset (Khot et al., 2018). The details of all datasets are illustrated in Appendix A.

We compare our models with two strong baselines: the BERT model (Devlin et al., 2019) and the MT-DNN model (Liu et al., 2019). While the former pre-trains the Transformer model on large amounts of unlabeled dataset, the latter further improves it with multi-task learning.

For BERT and MT-DNN, we use their publicly available code to obtain the final results. The setting of MT-DNN is slightly different from the setting of BERT in terms of optimizer choices. We implement our algorithms upon the $\mathbf{BERT_{BASE}}$

[1] data.quora.com/First-Quora-DatasetRelease-Question-Pairs

Model	CoLA	MRPC	STS-B	RTE
Reptile-PPS	**61.6**	**90.0**	**90.3**	**83.0**
Reptile-Uniform	61.5	84.0	**90.3**	75.7
Reptile-Mixed 2:1	60.3	87.8	**90.3**	71.0
Reptile-Mixed 5:1	**61.6**	85.8	90.1	74.7

Table 2: Effect of task distributions. We report the accuracy or Matthews correlation on development sets.

model.[2] We use the Adam optimizer (Kingma and Ba, 2015) with a batch size of 32 and learning rates of 5e-5 to train the models for 5 epochs in the meta-learning stage. We set the update step k to 5, the number of sampled tasks in each step to 8 and α to 1e-3.

3.1 Results

We first use the three meta-learning algorithms with PPS sampling and present in Table 1 the experimental results on the GLUE test set. Generally, the meta-learning algorithms achieve better performance than the strong baseline models, with Reptile performing the best.

Since the MT-DNN also uses PPS sampling, the improvements suggest meta-learning algorithms can indeed learn better representations compared with multi-task learning. Reptile outperforming MAML indicates that reptile is a more effective and efficient algorithm compared with MAML in our setting.

3.2 Ablation Studies

Effect of Task Distributions As we have mentioned above, we propose three different choices of the task distribution $p(T)$ in this paper. Here we train Reptile with these task distributions and test models' performance on the development set as shown in Table 2.

For uniform sampling, we set the number of training steps equal to that of the PPS method. For mixed sampling, we try mix ratios of both 2:1 and 5:1. The results demonstrate that Reptile with PPS sampling achieves the best performance, which suggests that larger amounts of auxiliary task data can generally lead to better performance.

Effect of Hyperparameters for Meta-Gradients In this part, we test the effect of the number of update steps k and the learning rate in the inner learning loop. The experimental results on the development sets are shown in Table 3. We find that setting k to 5 is the optimal strategy and more or fewer update steps may lead to worse performance.

Model	#Upt	α	CoLA	MRPC	STS-B	RTE
Reptile	3	1e-3	60.7	89.7	90.2	77.9
	5	1e-4	**62.0**	88.0	90.1	81.2
		1e-3	61.6	**90.0**	**90.3**	**83.0**
		1e-2	60.1	87.8	89.5	73.9
	7	1e-3	57.8	88.7	90.0	81.4

Table 3: Effect of the number of update steps and the inner learning rate α.

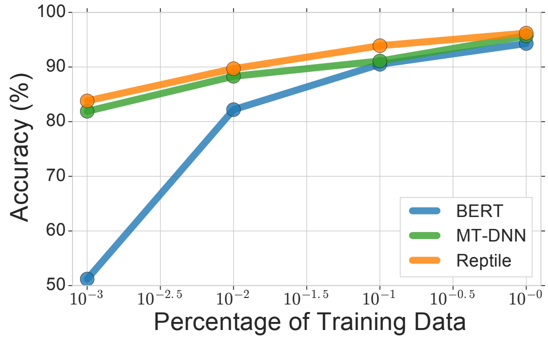

Figure 2: Results on transfer learning. The target task is SciTail which the model does not come across during the meta-learning stage.

Smaller k would make the algorithms similar to joint training as joint training is an extreme case of Reptile where $k = 1$, and thus cause the model to lose the advantage of using meta-learning algorithms. Similarly, Larger k can make the resulting gradients deviate from the normal ones and become uninformative.

We also vary the inner learning rate α and investigate its impact. The results are listed in Table 3. We can see that larger α may degrade the performance because the resulting gradients deviate a lot from normal ones. The above two ablations studies demonstrate the importance of making the meta-gradient informative.

3.3 Transferring to New Tasks

In this part, we test whether our learned representations can be adapted to new tasks efficiently. To this end, we perform transfer learning experiments on a new natural language inference dataset, namely SciTail.

We randomly sample 0.1%, 1%, 10% and 100% of the training data and test models' performance on these datasets. Figure 2 reveals that our model consistently outperforms the strong MT-

[2] $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ differ at the number of hidden layers (12 vs. 24), hidden size (768 vs. 1024) and the number of attention heads (12 vs. 16).

DNN baseline across different settings, indicating the learned representations are more effective for transfer learning. In particular, the algorithm is more effective when less data are available, especially compared to BERT, suggesting the meta-learning algorithms can indeed be helpful for low-resource tasks.

4 Related Work

There is a long history of learning general language representations. Previous work on learning general language representations focus on learning word (Mikolov et al., 2013; Pennington et al., 2014) or sentence representations (Le and Mikolov, 2014; Kiros et al., 2015) that are helpful for downstream tasks. Recently, there is a trend of learning contextualized word embeddings (Dai and Le, 2015; McCann et al., 2017; Peters et al., 2018; Howard and Ruder, 2018). One representative approach is the BERT model (Devlin et al., 2019) which learns contextualized word embeddings via bidirectional Transformer models.

Another line of research on learning representations focus on multi-task learning (Collobert et al., 2011; Liu et al., 2015). In particular, Liu et al. (2019) propose to combine multi-task learning with language model pre-training and demonstrate the two methods are complementary to each other.

Meta-learning algorithms have received lots of attention recently due to their effectiveness (Finn et al., 2017; Fan et al., 2018). However, the potential of applying meta-learning algorithms in NLU tasks have not been fully investigated yet. Gu et al. (2018) have tried to apply first-order MAML in machine translation and Qian and Yu (2019) propose to address the domain adaptation problem in dialogue generation by using MAML. To the best of our knowledge, the Reptile algorithm, which is simpler than MAML and potentially more useful, has been given less attention.

5 Conclusion

In this paper, we investigate three optimization-based meta-learning algorithms for low-resource NLU tasks. We demonstrate the effectiveness of these algorithms and perform a fair amount of ablation studies. We also show the learned representations can be adapted to new tasks effectively. Our study suggests promising applications of meta-learning algorithms in the field of NLU. Future directions include integrating more sophisticated training strategies of meta-learning algorithms as well as validating our algorithms on other datasets.

Acknowledgement

The authors are grateful to the anonymous reviewers for their constructive comments, and to Graham Neubig and Junxian He for helpful discussions. This material is based upon work generously supported partly by the National Science Foundation under grant 1761548.

References

Daniel Cera, Mona Diabb, Eneko Agirrec, Inigo Lopez-Gazpioc, Lucia Speciad, and Basque Country Donostia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. In *11th International Workshop on Semantic Evaluations*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yang Fan, Fei Tian, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. 2018. Learning to teach. In *ICLR*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *EMNLP*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *CVPR*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NeurIPS*.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *NAACL*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *NeurIPS*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohamadi, and Sanjeev Khudanpur. 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *InterSpeech*.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *ACL*.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. 2018. Deep layer aggregation. In *CVPR*.