Learning Task Similarity via Meta Learning and Model Splitting

Bora Uyumazturk
Stanford University
buyumazturk@stanford.edu

1 Introduction

Meta-learning has recently emerged as an effective approach to learning from limited amounts of data. In the meta learning context, you are given data for several tasks and the goal is to leverage that experience to learn a new target task with limited amounts of data. One particular flavor of meta-learning which has become more popular is learning initializations. Recent work has shown that good initializations can be learned to do very well on certain few shot tasks.

So far, all of these initialization use all of the previous datasets at once, ignoring any structure between tasks. In this project, we implement a non-parametric method for learning a task hierarchy during meta-learning. We hope that this approach can be used to further understand the challenges of meta learning as well as improve meta-learning algorithms when tasks contain structure.

2 Methods

2.1. Gradient-based Meta-Learning

$$\mathcal{T}_i = i\text{-th training task}$$
$$\phi = \text{model parameters}$$
$$\text{goal: learn } \phi \text{ to quickly learn new task}$$

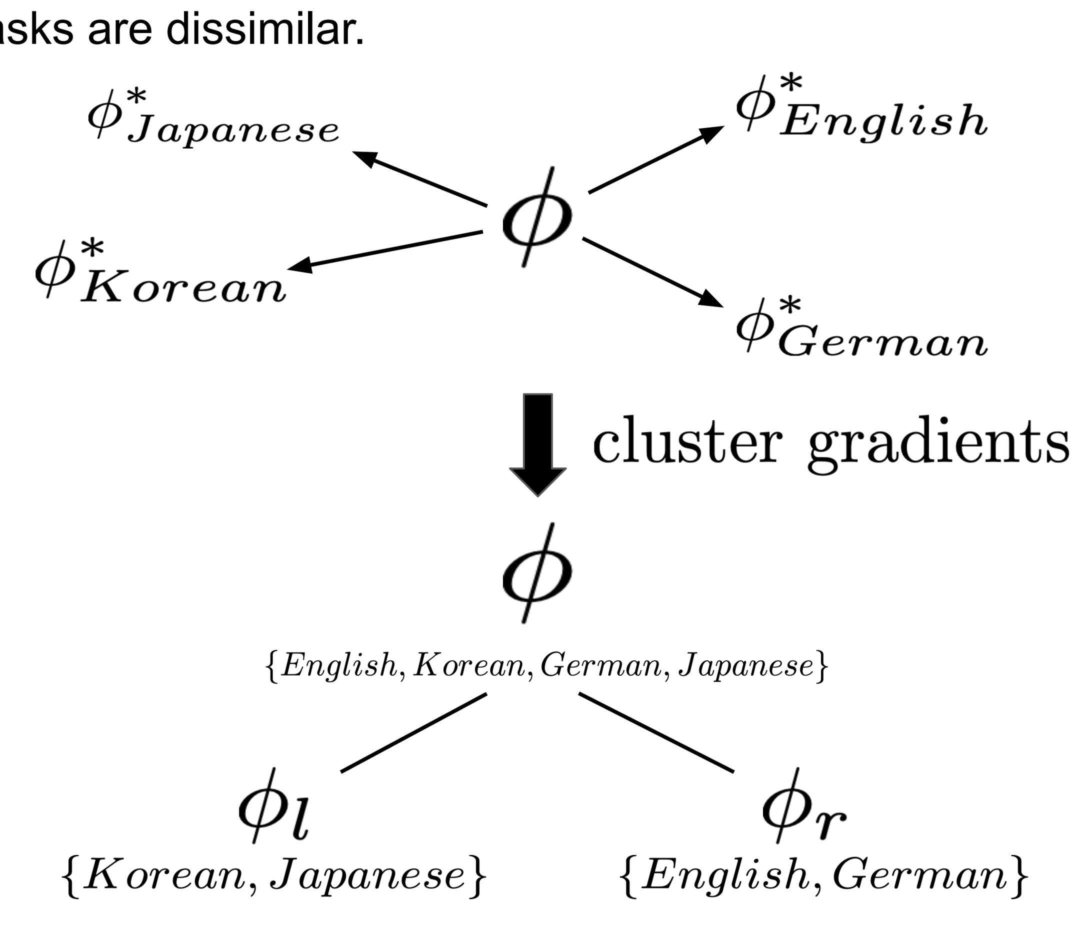
In general, these meta-learning algorithms learn good initializations by backpropagating through gradient updates on on meta-batches of different tasks. Since they were popularized by MAML, there have been several variants and extensions. In this project we focused on Reptile due to its simple implementation and low computational complexity.

2.2. Reptile

$$\phi_i^* = \text{parameters for } \mathcal{T}_i \text{ after gradient descent}$$
$$\phi \leftarrow \phi + \epsilon \frac{1}{n} \sum_{i=1}^{k} (\phi_i^* - \phi)$$

2.3. Model Trees and Parameter Splitting

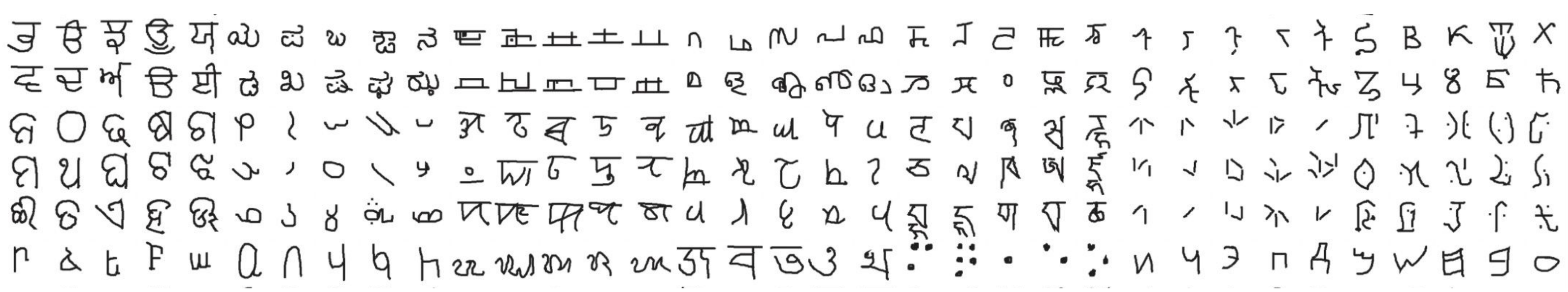
Meta-learning algorithms like Reptile are treat all training tasks identically, and effectively attempt to optimize for all of them at the same time. This can cause problems during training if tasks are dissimilar.



To try and remedy this, we adaptively split our tasks along with the mode parameters by clustering the gradient directions for each task. This leaves us with "children", which we can then continue to meta-train on their subset of tasks. This produces model tree of increasing specificity. In our experiment we examine the efficacy of this both as a training algorithm and a task similarity metric.

3 Dataset

The Omniglot dataset consists of 20 instances of 1623 characters across 50 different alphabets. It is a standard few-shot learning benchmark. Moreover, there is clearly structure between the given tasks (e.g. alphabets), which make it an ideal candidate for model splitting.



4 Experiment

We hand selected a set of 35 training languages and 15 testing languages from the Omniglot dataset and training on the 5-shot 5-way classification task. At test time, for each testing languages, the model is given 5 examples of 5 classes to train on then is asked to classify 5 instances of each of those classes.

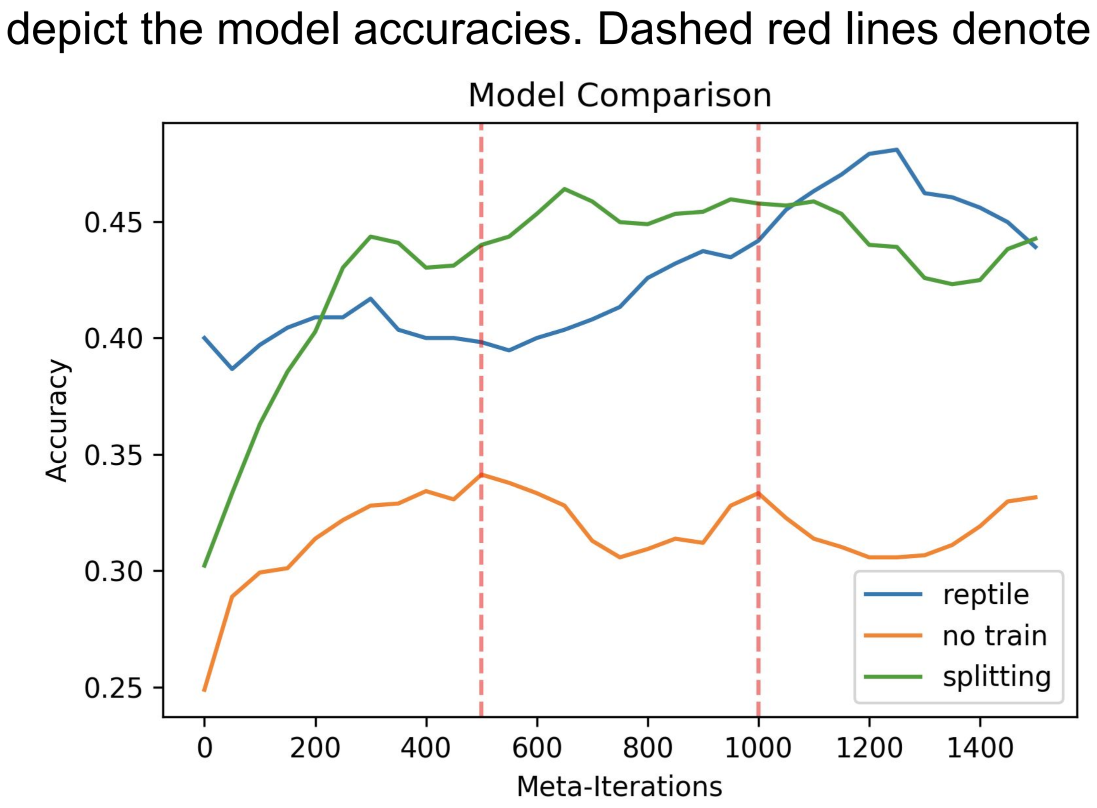
We used a small model to test our our methods. It consisted of 2 convolutional layers each with with 3x3 kernels and 64 channels. The output of these was then put through a dense layer and the output was determined by taking the argmax of the softmax of the output scores.

Our first baseline was a model that did no meta training, but instead tried to learn directly from the 25 examples given. Our next baseline was Reptile with no model splitting. Our final baseline was Reptile with splitting, where splits were made after 500 training iterations.

5 Results

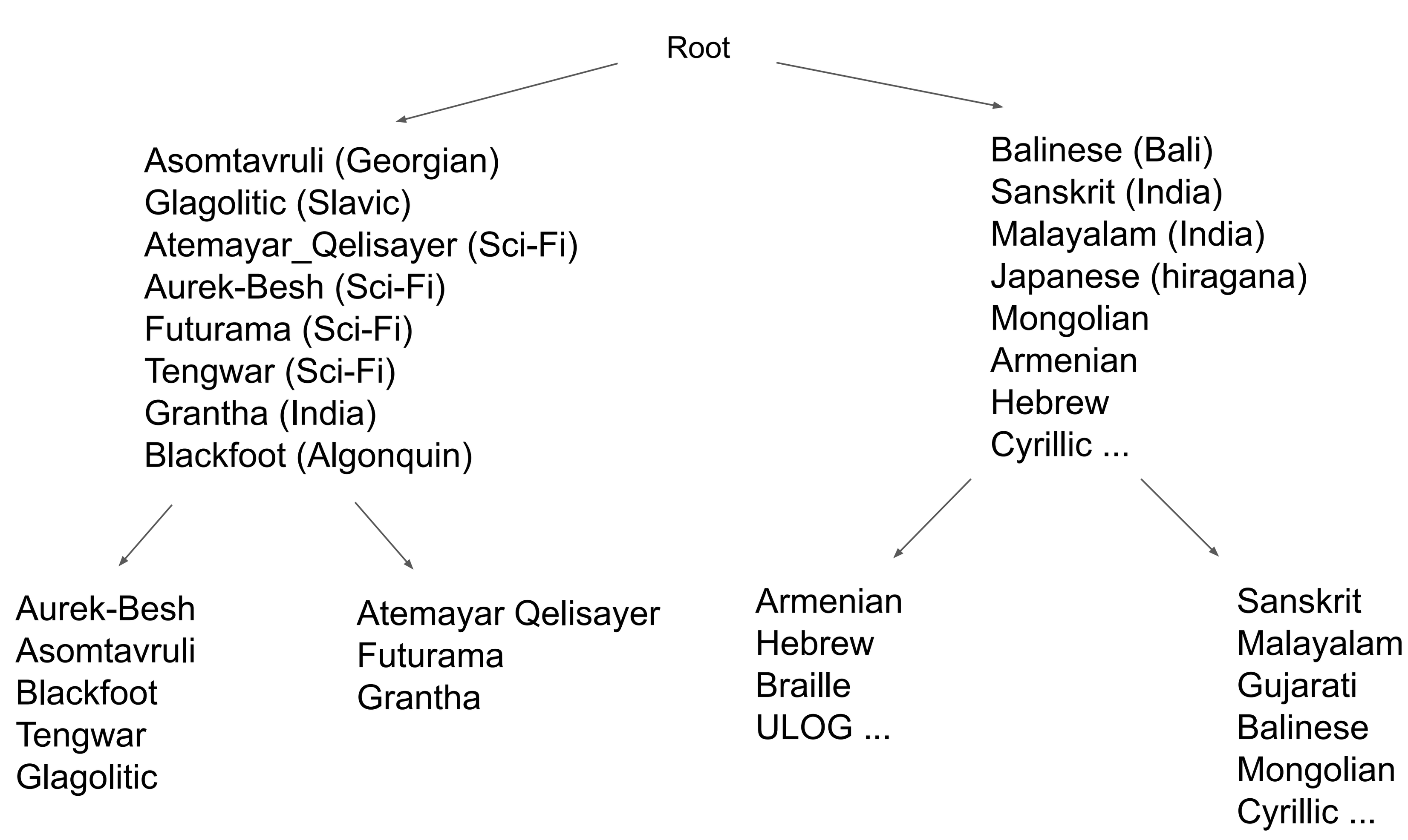
5.1. Model Accuracies

Below we depict the model accuracies. Dashed red lines denote model split points.



5.2. Task Clustering

Below we depict the task hierarchy learned during training. .



6 References

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.* 2017. arXiv: 1703.03400 [cs.LG]
[2] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science* 3540.6266 (2015), pp.1332-1338. ISSN: 0036-8075. DOI: 10.1126/science.aab3050. URL: https://science.sciencemag.org/content/350/6266/1332
[3] Alex Nichol, Joshua Achiam, and John Schulman. *On First-Order Meta-Learning Algorithms.* 2018. arXiv: 1803.02990