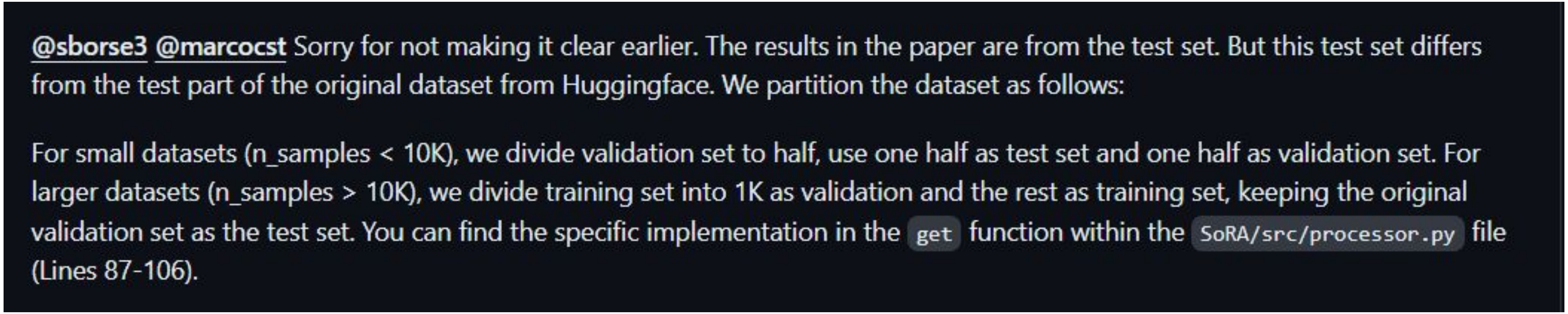
SoRA/Adalora Experiments

Reproducing SoRA Results (2 Runs)

Results	**Epoch**	**Accuracy**	**Loss**	**Runtime**	**Samples**
Train	50/50	-	0.083/ 0.09	2870.99/ 3753	2490/ 2490
Evaluation Train	50/ 50	1/ 1	0.0009/ 0.0009	9.98/ 12.54	1000/ 1000
Evaluation	50/ 50	0.877/ 0.877	0.839/ 0.943	1.38/ 1.73	138/138
Test	15/15	0.55/0.55 **(Paper Result: 0.877)**	0.69/0.69	1.40/1.74	138/138

SoRA Datasets Preparation



@sborse3 @marcocst Sorry for not making it clear earlier. The results in the paper are from the test set. But this test set differs from the test part of the original dataset from Huggingface. We partition the dataset as follows:

For small datasets (n_samples < 10K), we divide validation set to half, use one half as test set and one half as validation set. For larger datasets (n_samples > 10K), we divide training set into 1K as validation and the rest as training set, keeping the original validation set as the test set. You can find the specific implementation in the `get` function within the `SoRA/src/processor.py` file (Lines 87-106).

Problems Identified/ Roadblocks

Issues Faced in Reproducing Results for SoRA

- **Backbone Model Compatibility:**
  - LLaMA-2-7B not supported by OpenDelta's LoRA configuration.
  - OpenDelta supports only models from the Transformers module.

- **Roberta-large Implementation:**
  - Tested on BoolQ task (SuperGLUE) in Alora paper.
  - No pre-implemented SuperGLUE script available.
  - Custom implementation not compatible with original preprocessing script.

Problems Identified/ Roadblocks

Issues with Adalora

- **Complex Structure:**
  - Modified Transformer trainer file for fine-tuning with Adalora.
  - Multiple supporting files also modified, difficult to identify changes.
- **Alternative Method:**
  - Using PEFT/AdaConfig from Hugging Face.
  - SFTTrainer does not support custom compute metrics functionality, limiting accuracy calculation.
- **Callback Functions:**
  - Need to run evaluation step manually.
  - Forward pass, gather logits and labels, then calculate accuracy.
  - Evaluation called twice on each run.

Problems Identified/ Roadblocks

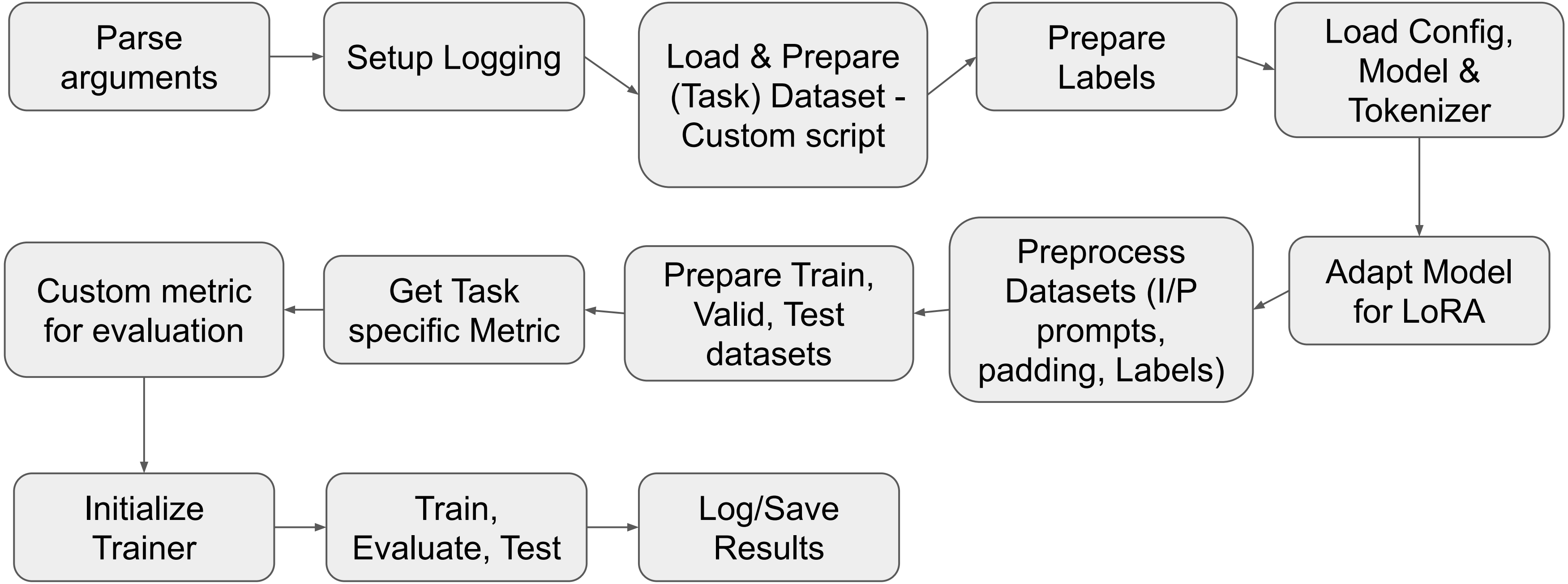
Modularity:

- Different preprocessing steps for each method.
- Resulting in long and complicated code files.

Potential Steps

1. Implement custom Superglue script - Adapt preprocessing and data loading for SuperGLUE tasks. Ensure compatibility with Sora and adalora.
2. Experiment with the original Adalora code

General Idea of Fine tuning using Glue



Parse arguments

Setup Logging

Load & Prepare (Task) Dataset - Custom script

Prepare Labels

Load Config, Model & Tokenizer

Adapt Model for LoRA

Preprocess Datasets (I/P prompts, padding, Labels)

Prepare Train, Valid, Test datasets

Get Task specific Metric

Custom metric for evaluation

Initialize Trainer

Train, Evaluate, Test

Log/Save Results