

# Pretrained Language Models are Symbolic Mathematics Solvers too!

## Pre-requisites

This code depends on the following packages:

1. Torch
2. NumPy
3. SymPy
4. Transformers
5. Apex

## Structure

The following intruduces what different code files do.

### trainer.py

contains code for fine-tuning the pre-trained language models. Please modify the following parameters for running:

1. ``language``: the pre-trained language.
2. ``Model_Type``: mbart or Marian.
3. ``path1`` and ``path2``: the path of the training and the validation data.
4. ``max_input_length`` and ``max_output_length``: 1024 for the mBART model and 512 for the Marian-MT model.
5. ``model_name``: name of the model you wish to save.

### evaluator.py

contains code for evaluating the fine-tuned language model on the symbolic math data. Please modify the parameter 1-4 same as the `trainer` section and also modify the following parameter:

1. ``path``: the path of the test dataset.
2. ``saved_model``: the path of the saved fine-tuned model.

### src/hf\_utils.py

contains code for reading the datasets and some utilities for evaluation.

The rest of the code is adopted from [Deep learning for symbolic mathematics \(Lample et al.\)](#).

## Datasets

The datasets are available [here](#).

1. `train`, `valid`, and `test` files contain the training, validation and test datasets for the mBART model.
2. `language_data` contains data for the training, validation and test datasets of the Marian-MT model.
3. `distribution_test` contains the test files for the distribution shift section (polynomial, trigonometric and logarithmic).