Math Problem Classification Project

Overview

This project explores various approaches to classify math problems into one of eight categories based on the dataset from the Kaggle competition: Classification of Math Problems by Kasut Academy. It includes data augmentation using AWS Bedrock and fine-tuning of different transformer models, including sequence classifiers, sequence-to-sequence models, and instruction-following LLMs. An ensemble approach is also considered.

Prerequisites

- Python environment with standard data science libraries (pandas, numpy, etc.).
- PyTorch.
- Hugging Face Libraries: transformers, datasets, evaluate, trl.
- unsloth: For efficient Llama model fine-tuning.
- boto3: For data generation scripts using AWS Bedrock. Requires configured AWS credentials with Bedrock access.
- Original competition data files: train.csv, test.csv.

Execution Order and Script Descriptions

The recommended order for running the scripts is as follows:

1. Dataset Acquisition and Download Trained Models (Shell Script)

> [!NOTE] the below command is crucial for downloading the dataset and models. Make sure to run it before proceeding with any other scripts.

```
* Run the `get_assets.sh` script to download the necessary dataset files from Google Drive.
* This script will create a directory named `./code/data/` and download the following files:
    * train.csv
    * test.csv
* This script will also create a directory named `./code/output/models/` and download the model files into it.
* To run the script, use the command:
    ```bash
    chmod +x get_assets.sh
    <!-- NOTE: ./get_assets.sh <MODEL_DRIVE_ID> <DATA_DRIVE_ID> -->
    ./get_assets.sh 1ancje2FsGw9dTMMCXCO2CfuXsqgDJKBE 1OpVGWl8JlRD3G3mB8IqoAE6DY1bd08sv
    ```
```

2. Run Demo (Shell Script)
    - The demo application is implemented using Streamlit and can be run using the command:

      ```
      chmod +x run_demo.sh
      ./run_demo.sh
      ```

    - This will start a local server on port 8888 (hardcoded this due to security group restrictions on AWS), and you can access the demo at `http://localhost:8888`.
    - The demo includes:
        - `app.py` : Frontend for the classification demo.
        - `model_utils.py` : Handles model loading and prediction.
        - `train.py` : Contains cross-validation training logic.
        - `augment.py` : Implements data augmentation techniques.
        - `run_demo.sh` : Shell script to run the demo. It creates and activates a virtual environment, installs dependencies, and runs the demo application on port 8888.
        - `requirements.txt` : Contains the required packages for the demo.
    - The shell script will create a virtual environment, install the required packages, and run the demo application in one command.