Assignment 5: Sequence-to-Sequence Learning with Transformers for Arithmetic

This project implements an Encoder-Decoder Transformer model from scratch (using PyTorch's `nn.MultiheadAttention` but not `nn.TransformerEncoder/Decoder`) to perform basic arithmetic (addition and subtraction) on character sequences.

Structure

- `config.py`: Contains all hyperparameters, vocabulary definitions, file paths, and configuration settings.
- `data_utils.py`: Handles synthetic data generation, vocabulary management, tokenization, PyTorch Dataset creation, and DataLoader setup.
- `model.py`: Defines the Transformer architecture components: Positional Encoding, EncoderLayer, DecoderLayer, Encoder, Decoder, and the main Seq2SeqTransformer model.
- `train.py`: Orchestrates the entire process: data generation (if needed), model building, training loop, evaluation, inference (greedy decoding), metric calculation (Exact Match, Char Accuracy, Perplexity), model saving, and basic analysis (test set performance, generalization test, sample error printing).
- `report.pdf`: (To be created by the student) Contains the detailed analysis, results, discussion, ablation studies, etc., as required by Part 4 of the assignment.
- `models/`: Directory where the trained model weights (`transformer_arithmetic.pt`) will be saved.
- `data/`: Directory where the generated datasets (`train.json`, `val.json`, `test.json`, `gen_test.json`) will be saved.
- `README.md`: This file.

Setup

PROF

1. **Clone the repository:**

```
git clone <your-repo-url>
cd <repo-directory>
```

2. **Create a virtual environment (recommended):**

```
python -m venv venv
source venv/bin/activate  # On Windows use `venv\Scripts\activate`
```

3. **Install dependencies:**

```
pip install torch tqdm numpy
# Ensure you install the correct PyTorch version for your system
(CPU/GPU)
# See: https://pytorch.org/get-started/locally/
```

Running the Code

1. **Generate Data, Train, and Evaluate:**
   Execute the main training script. It will automatically handle data generation if the `.json` files are not found in the `data/` directory.

   ```
   python train.py
   ```

   - This script performs:
     - Data generation (if necessary).
     - Model initialization.
     - Training loop over specified epochs, saving the best model based on validation loss to `models/transformer_arithmetic.pt`.
     - Evaluation on the test set using the best saved model.
     - Evaluation on the generalization test set (longer inputs).
     - Prints final metrics (Loss, Perplexity, Exact Match Accuracy, Character-Level Accuracy).
     - Prints a few examples of incorrect predictions from the test set for basic error analysis.

2. **Analysis (Part 4):**

   - The quantitative results and generalization performance are printed by `train.py`.
   - Examine the printed errors for initial error analysis.
   - **Crucially, you need to perform further analysis as described in Part 4:**
     - **Deeper Error Analysis:** Categorize errors, correlate with input features (length, carries/borrows). You might want to modify `train.py` or create a separate analysis script/notebook (`analysis.ipynb`) to load the model and data for this.
     - **Ablation/Sensitivity Study:** Modify hyperparameters or architectural choices in `config.py` or `model.py`, re-run `train.py` (potentially with fewer epochs/data for speed), compare results against the baseline, and document findings in your report.
   - **Write the Report:** Compile all findings, justifications, results (tables/figures), and discussion into `report.pdf`.

PROF

```
## File Structure Overview
├── config.py                    # Configuration and hyperparameters
├── data_utils.py                # Data generation and loading
utilities
├── model.py                     # Transformer model definition
```

```
├── train.py                          # Main script for training and
evaluation
├── README.md                         # This file
├── report.pdf                        # (Student needs to create this
report)
├── data/                             # Directory for generated datasets
(created by script)
│   ├── train.json
│   ├── val.json
│   ├── test.json
│   └── gen_test.json
└── models/                           # Directory for saved model weights
(created by script)
    └── transformer_arithmetic.pt
```

The best trained model weights are saved as `models/transformer_arithmetic.pt`.