**T1 小小数学助手**

**故事背景**

你来到小学帮助小朋友学数学。孩子们每天都会来问你各种数学题：

* 有的小朋友只想知道简单的加减乘除结果；
* 有的小朋友希望你一次性帮他算一整页的练习题；
* 还有的小朋友要求答案统一、整齐，不要多余的话，这样才能方便对照答案本子。

不过，这里的小朋友们写题目时非常"自由"。有的在纸上写正常的 +、-，有的画奇怪的符号，比如 ⊕、－、➕、➖，甚至还有人用中文"加""减"。如果直接用普通计算器，往往认不出这些符号，会算错。

于是，你决定编写一个**小小数学助手**，让它能够理解孩子们各种各样的题目，并用统一的答案格式来回答。这个助手基于 **PyTorch** 和 **Transformers** 库，可以自动回答孩子们提出的数学问题。

**任务目标**

你需要在 `submission.py` 中完成**三个核心函数**，让数学助手具备以下能力：

**1. 定义回答规则：`build\_system\_prompt() -> str`**

编写一个 system prompt，引导模型识别各种数学符号并按照 [Answer]: 数值 的格式输出答案。

**2. 构造对话模板：`apply\_chat\_template\_single(...) -> str`**

使用 tokenizer 将 system prompt 和用户问题组合成模型能理解的输入格式。

**3. 实现推理：`generate\_single(...)` 和 `generate\_batch(...)`**

实现单条推理和批量推理两种模式，返回模型生成的 token 序列。

**注意**：直接返回模型原始输出，不要做后处理。评测程序会统一处理输出格式。

**实现要求**

**文件结构**

|  |
| --- |
| Plain Text T1/ ├── data/ │ ├── test\_data\_1.jsonl # 阶段一测试数据（8题） │ └── test\_data\_2.jsonl # 阶段二测试数据（32题） ├── evaluate.py # 评测主程序（不可修改） ├── submission.py # 考生实现文件（仅此文件可修改） └── README.md # 本文档 |

**修改规则**

* ✅ **仅可修改** `submission.py` 中标注的"考生实现区域"
* ✅ 可以新增少量辅助函数
* ❌ **不得修改** `evaluate.py` 的任何内容
* ❌ **不得在函数中做答案后处理**（如提取数字、格式转换等）

**技术要求**

* 使用 PyTorch 和 Transformers 库
* 正确处理 tokenizer 的 padding 和 truncation
* 批量推理需要考虑内存和效率

**评测说明**

**运行方式**

**演示模式（详细输出）：**

|  |
| --- |
| Bash python evaluate.py # 或 python evaluate.py --mode demo |

* 展示每个题目的完整推理过程
* 显示模型原始输出、处理后文本、提取的答案
* 展示错误样例的详细信息

**评分模式（简洁输出）：**

|  |
| --- |
| Bash python evaluate.py --mode grading |

* 简洁的进度输出
* 最终输出详细评分和性能指标
* 用于正式评测

**评测流程**

**阶段一：逐条推理（8 题）**

**测试内容**：

* 固定的 8 道数学题（基础四则运算 + 特殊符号）
* 逐条调用 generate\_single 进行推理
* 测试模型对各种符号和表达的理解能力

**示例题目**：

|  |
| --- |
| JSON {"problem": "12 + 35", "answer": "47"} {"problem": "6的平方？", "answer": "36"} {"problem": "45 加 89", "answer": "134"} |

**输出内容**（Demo 模式）：

* 题目原文
* 模型原始输出（可能包含 <think> 标签）
* 非思考部分文本
* 提取的答案
* 判定结果（✅/❌）
* 单题耗时

**评分标准**：

- **正确性**：每题 10 分，答对 ≥5 题给满分 50 分

- **时间性能**：单条平均耗时 ≤5s 满分 10 分，得分 = min(10, 50/t)

**阶段二：批量推理（32 题）**

**测试内容**：

* 32 道数学题（包含多种符号和表达）
* 调用 generate\_batch 进行批量推理
* 测试批量处理的效率和准确性

**示例题目**：

|  |
| --- |
| JSON {"problem": "91 + 24", "answer": "115"} {"problem": "34 ⊕ 19", "answer": "53"} {"problem": "47 － 41", "answer": "6"} |

**特殊测试**（Grading 模式）：

* 先用 generate\_single 对相同的 32 题进行单条推理
* 记录单条推理的总耗时作为基准
* 然后进行批量推理，计算加速比

**输出内容**（Demo 模式）：

* 仅展示错误的样例（正确的不显示）
* 全部正确时显示祝贺信息

**评分标准**：

* **正确性**：每题 1 分，答对 ≥30 题给满分 30 分
* **时间性能**：加速比测试
* 加速比 = 单条推理总耗时 / 批量推理总耗时
* 3 倍加速：满分 10 分
* 1 倍加速：0 分
* 1-3 倍之间：线性给分，得分 = (加速比 - 1) × 5

**总分构成（100 分）**

|  |  |  |
| --- | --- | --- |
| 项目 | 分值 | 说明 |
| 阶段一 - 正确性 | 50 分 | 8题，每题10分，≥5题满分 |
| 阶段一 - 时间性能 | 10 分 | 单条平均 ≤5s 满分 |
| 阶段二 - 正确性 | 30 分 | 32题，每题1分，≥30题满分 |
| 阶段二 - 时间性能 | 10 分 | 加速比测试，3倍满分 |
| 总分 | 100 分 |  |