MRI Report LLM Extractor (基于 LLM 的 MRI 报告信息提取与评估系统)

项目描述

本项目旨在探索利用大型语言模型 (LLM) 的 API (例如 OpenAI GPT-4 Turbo) 自动从本地存储的医学报告图像（特别是 MRI 报告）中提取结构化信息，以期减轻医学工作者重复且枯燥的数据录入工作。项目还包括将 LLM 提取结果与人工提取的“标准答案” (Ground Truth) 进行对比，以评估 AI 提取的准确性，验证其在实际医疗场景中辅助工作的潜力。

主要功能

- **PDF 转图像:** 将 PDF 格式的 MRI 报告转换为 PNG 图像，便于后续处理。
- **数据准备:** 清理并准备用于对比的 Ground Truth Excel 数据，并生成 JSON 模板以规范 LLM 的输出格式。
- **LLM 信息提取:** 调用 OpenAI API (GPT-4 Turbo)，将报告图像和 JSON 模板发送给模型，提取关键信息并以 JSON 格式返回。
- **结果校验与修正:** 对比 LLM 输出的 JSON 和预定义的模板，自动修正字段名错误（如拼写相似）并补全缺失字段。
- **格式转换:** 将修正后的 JSON 数据转换为更易于查看和使用的 Excel 格式。
- **准确性评估:** 逐项对比 LLM 提取的数据（Excel 格式）与 Ground Truth 数据，计算单元格级别的准确率。
- **差异报告:** 生成详细的文本报告，列出提取错误或不一致的具体字段、原始值和提取值。
- **汇总分析:** 读取所有单个报告的准确率文件，计算平均准确率、中位数、标准差等统计指标，并生成可视化图表。

目录结构

LLM-test/ ├── .env # 环境变量文件 (需手动创建) ├── .gitignore # Git 忽略规则文件 ├── requirements.txt # Python 依赖库列表 ├── README.md # 项目说明文档 ├── data/ # 存放所有原始数据和模板 │ ├── ground_truth/ # (用户需在此放入自己的 Ground Truth Excel 文件) │ ├── raw_reports/ # (用户需在此放入自己的 PDF 报告，可分子目录) │ └── templates/ # 存放 JSON 模板 (通常可由脚本生成) ├── results/ # 存放所有脚本运行生成的输出文件 (被 .gitignore 忽略) │ ├── accuracy_reports/ # 存放每个报告的准确率文本文件 │ ├── extracted_data/ # 存放 LLM 提取的数据 (不同格式) │ │ ├── excel/ │ │ ├── json_checked/ │ │ └── json_raw/ │ ├── overall_analysis/ # 存放汇总分析结果 (图表、统计) │ └── processed_images/ # 存放 PDF 转换后的 PNG 图片 ├── src/ # 存放所有 Python 源代码 │ ├── **init**.py │ ├── config.py # <--- 项目配置文件 (路径等) │ ├── api_interaction.py │ ├── data_conversion.py │ ├── data_extraction.py │ ├── data_validation.py │ ├── evaluation.py │ ├── json_to_excel.py │ ├── main.py │ ├── reporting.py │ └── ... (其他 .py 文件) └── venv/ # Python 虚拟环境目录 (自动生成，被 .gitignore 忽略)

安装与设置

1. **获取项目文件:**
    - 如果使用 Git，请克隆仓库: `git clone <your-repository-url>`
    - 如果是本地文件夹，请确保拥有代码文件。

2. **创建并激活 Python 虚拟环境:** (推荐使用 Python 3.7 或更高版本)

```
# 1. 进入项目根目录 LLM-test/
cd path/to/LLM-test

# 2. 创建虚拟环境 (名为 venv)
python -m venv venv

# 3. 激活环境 (根据你的操作系统选择命令)
#    Windows CMD:               .\venv\Scripts\activate.bat
#    Windows PowerShell:        .\venv\Scripts\Activate.ps1
#    Windows Git Bash / macOS / Linux: source venv/Scripts/activate  (或
source venv/bin/activate)
```

激活成功后，命令行提示符前会出现 `(venv)`

3. **安装依赖库:** 确保虚拟环境已激活。

```
pip install -r requirements.txt
```

4. **[重要 - Windows 用户请确认] 安装 Poppler (用于 PDF 转图片):**
   - `pdf2image` 库在 Windows 上通常需要 Poppler 工具来处理 PDF。
   - **安装步骤 (Windows 示例):**
     1. 访问 Poppler for Windows (Manh)。
     2. 下载最新的 `Release-*.zip` 文件。
     3. 解压到固定位置（例如 `C:\Program Files\poppler-24.02.0`）。
     4. 将解压后文件夹内的 `bin` 目录 (例如 `C:\Program Files\poppler-24.02.0\bin`) **添加到系统的 PATH 环境变量**中。
     5. 重启命令行终端或电脑。
   - **macOS (使用 Homebrew):** `brew install poppler`
   - **Linux (Debian/Ubuntu):** `sudo apt-get update && sudo apt-get install poppler-utils`
   - *(请根据你的实际情况确认并调整此步骤)*

5. **配置环境变量:**
   - 在项目根目录 `LLM-test/` 下手动创建一个名为 `.env` 的文本文件。
   - 在 `.env` 文件中添加你的 OpenAI API 密钥：

     ```
     OPENAI_API_KEY="sk-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
     # 如果需要其他 API 密钥，也在此添加
     ```

     (将 `sk-xxx...` 替换为你自己的真实密钥)
   - **重要:** `.env` 文件已被添加到 `.gitignore`，不会上传到 GitHub。

6. **准备并配置你自己的输入数据:**

- **数据隐私:** 由于涉及医疗数据隐私，本项目**不包含**用于开发的原始 PDF 报告和 Ground Truth Excel 文件。这些数据已被添加到 `.gitignore` 中，不会上传到代码仓库。
- **用户操作:**
  1. 请将你自己的 Ground Truth Excel 文件放入 `data/ground_truth/` 目录。
  2. 请将你自己的 PDF 报告文件放入 `data/raw_reports/` 目录（可以根据需要创建子目录，如 `BENSON DEID RRI REPORTS`）。
  3. **关键步骤:** 打开 `src/config.py` 文件，根据你的实际文件位置和名称，**修改**以下（及其他相关的）路径配置变量：
     - `ORIGINAL_GROUND_TRUTH_XLSX`: 指向你的原始 Ground Truth Excel 文件。
     - `DEFAULT_PDF_SCAN_DIR`: 指向包含你的 PDF 报告的目录。
     - (以及 `config.py` 中其他可能需要根据你的数据调整的路径或文件名设置)。

使用方法 (工作流程)

重要: 建议始终从项目根目录 (`LLM-test/`) 运行所有命令，并确保虚拟环境 (`venv`) 已激活，且你已按步骤 6 准备好自己的数据并**正确配置了 `src/config.py` 中的路径。**

1. **首次运行或数据更新时 - 预处理:**
   - **(a) 清理 Ground Truth 并生成 JSON 模板:**

     ```
     python src/data_extraction.py
     ```

     - *输入:* 你在 `config.py` 中指定的 `ORIGINAL_GROUND_TRUTH_XLSX` 文件。
     - *输出:* 清理后的 Excel 文件 (`CLEANED_GROUND_TRUTH_XLSX`) 和 JSON 模板 (`TEMPLATE_JSON_PATH`)，具体路径也在 `config.py` 中定义。
   - **(b) 转换 PDF 报告为图片:**

     ```
     python src/data_conversion.py
     ```

     - *输入:* 你在 `config.py` 中指定的 `DEFAULT_PDF_SCAN_DIR` (或其他相关配置) 下的 PDF 文件。
     - *输出:* PNG 图片到 `config.PROCESSED_IMAGES_DIR` 定义的目录。
     - *依赖:* Poppler (请确认已按步骤 4 安装配置)。
2. **运行主处理流程 (提取、校验、转换、评估):**
   - **处理所有自动发现的报告 (推荐):**

     ```
     python src/main.py
     ```

     - *扫描目录:* 由 `config.DEFAULT_PDF_SCAN_DIR` 定义，或通过 `--pdf-dir` 参数指定。
   - **处理指定的报告:**

```
# 处理单个报告 RRI002
python src/main.py -i RRI002
# 处理多个报告 RRI003 和 RRI010
python src/main.py -i RRI003 RRI010
```

- **扫描指定 PDF 目录进行自动发现:**

```
# 假设你的 PDF 在 data/raw_reports/MyReports/
python src/main.py --pdf-dir "data/raw_reports/MyReports"
```

- *输出*: 中间及最终结果会存放在 `config.py` 中定义的 `results/` 下的相应子目录中（这些目录会被 `.gitignore` 忽略）。

3. **生成汇总分析报告:**

- 当处理完一批报告，希望查看整体性能时运行：

```
python src/reporting.py
```

- *输入*: `config.ACCURACY_REPORTS_DIR` 目录下的 `_accuracy.txt` 文件。
- *输出*: 汇总统计文本和图表到 `config.OVERALL_ANALYSIS_DIR` 目录。

依赖项

所有 Python 依赖项及其版本均在 `requirements.txt` 文件中列出。请在激活虚拟环境后使用 `pip install -r requirements.txt` 进行安装。