AI Agent - DataSet URL Finder 项目报告

王思宇　薛佳音　王镜凯

2025 年 6 月 5 日

目录

1 DataSet URL Finder

1.1 项目概述

- **目标**：从学术论文 PDF 中自动识别和提取数据集相关的 URL。
- **核心技术**：
  - PDF 处理与文本提取
  - 正则表达式 + 大语言模型提取 URL
  - 三维评分机制，确定一个 URL 是否确为数据集 URL

1.2 系统架构

下图为我们设计的整个系统的架构图：

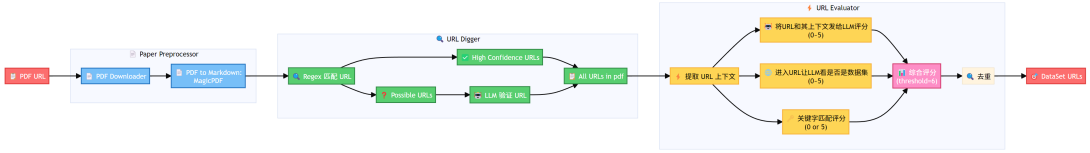

图 1: 整体系统架构

项目从逻辑上拆分成了 Paper Preprocessor, URL Digger 和 URL Evaluator 三个部分。各部分的作用如下文所述。

1.3 Paper Preprocessor

1.3.1 核心目标

完成 PDF 下载任务和文字提取任务。
对应代码中的 src/pdfurl2md.py。

1.3.2 架构图

下图为 Paper Preprocessor 的子架构图：

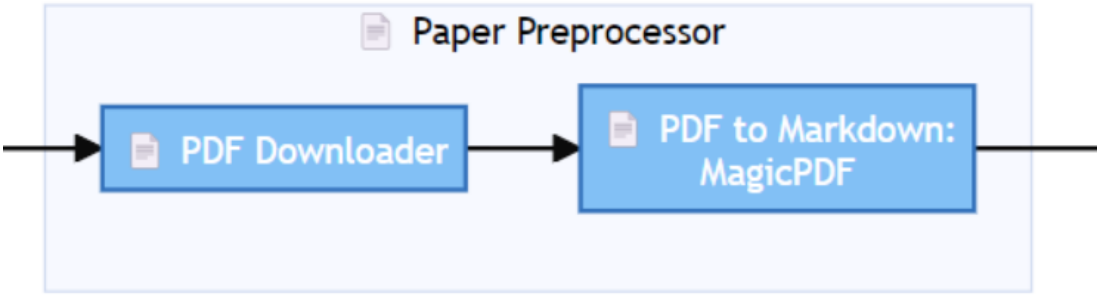

图 2: Paper Preprocessor 架构图

1.3.3 技术实现

- **PDF 下载**：直接使用 requests 库逐字节地下载 pdf 即可。
- **文字提取**：使用 magic-pdf 转换 PDF Markdown

我们使用 magic-pdf 的原因是其使用高准确率视觉模型识别 pdf，精度高，利于我们后续在此基础上提取所有 URL 并相应评测出数据集 URL。

1.3.4 存在问题

跨行 URL 处理问题 magic-pdf 转换时，跨行的 URL 可能在换行处插入额外空格，影响 URL 完整性识别。

在聆听了其他小组的汇报后，知悉了可以通过直接提取 pdf 的 href 属性而定位链接，然而我们从实践上发现一些没有标注协议头（https://, http://）的缩略版 URL 缺少了 href 属性，因而这一方法并不通用，目前暂无特别优质的方法解决这一问题。

可能的解决思路有：- 后续通过智能匹配算法处理 - 大模型验证机制补偿

1.4 URL Digger

1.4.1 核心目标

从 PDF 文字中提取所有的 URL，不管是不是数据集 URL。

对应代码中的 src/urldigger.py。

1.4.2 架构图

下图为 URL Digger 的子架构图：

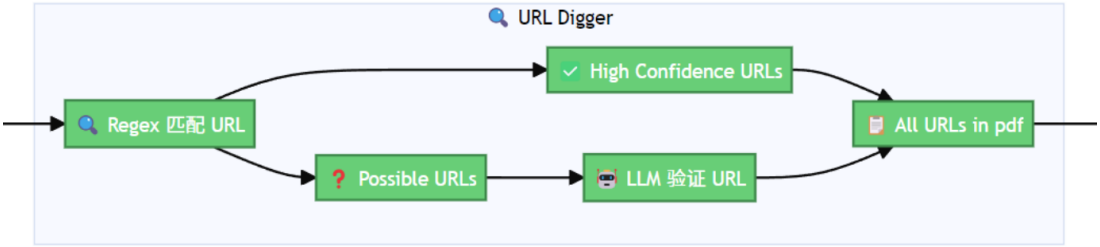

图 3: URL Digger 框架

1.4.3 双重匹配策略

URL Digger 的处理流程分为四步，核心为双重匹配策略：

1. 正则表达式扫描全文
2. 分类：高置信度 vs 可能 URL
3. LLM 验证可能 URL 的正确性
4. 高置信度 URL 与正确的可能 URL 合并生成完整 URL 列表

其中第二步需要详细说明：应对有 https://，http://头和没有该协议头的 URL，我们设计了两套正则表达式分别匹配：

```
high_confidence_pattern = r'(https?:\/\/(?:www\.|(?!www))[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[^\s]{2,}|www\.[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[^\s]{2,}|https?:\/\/(?:www\.|(?!www))[a-zA-Z0-9]+\.[^\s]{2,}|www\.[a-zA-Z0-9]+\.[^\s]{2,})'
possible_url_pattern = r'([a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[^\s]{2,}|[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[^\s]{2,})'
```

在可能的 URL 被匹配出来后，我们做了一些简单的基于规则的筛选，然后将剩下不确定的给 AI 让其判断到底是不是合法的 URL 链接并让 AI 做一些可能的修正。最后与高置信度 URL 合并成为论文中所有的 URL 列表。

1.5 URL Evaluator

1.5.1 核心目标

通过三重评分机制，从论文提取出的所有 URL 中筛选出真正的数据集 URL。

对应代码中的 src/urlprober.py。

1.5.2 架构图

下图为 URL Evaluator 的子架构图：

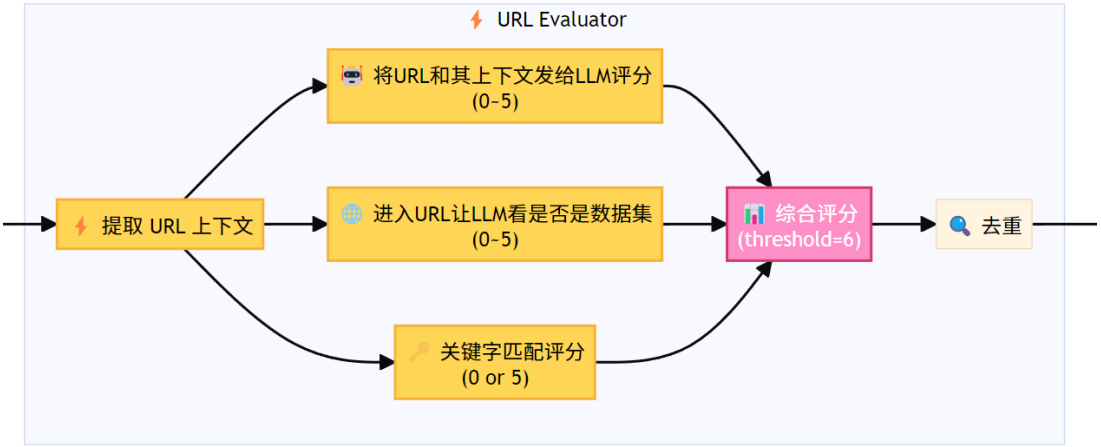

图 4: URL Evaluator 框架

1.5.3 三重评分机制

三重评分机制的详细说明如下：

上下文分析 LLM 评分 (0-5 分)

- 分析 URL 周围文本
- 让大模型判断与数据集的相关性

相应的 AI prompt 如下：

```
prompt = f"""
Given the following URL and its context from an academic paper:
URL: {url}
Context: {context}

Please determine from the context if this URL provides access to actual dataset files or data repositories that can be downloaded or accessed.

Consider these criteria and platform-specific guidelines: (The higher the score, the more likely it is to provide datasets. Don't hesitate to rank very high if the context suggests it is a dataset link, and don't hesitate to rank very low if it obviously is not a dataset link.)

1. DATASET REPOSITORIES (Score 1.0-5.0):
- HuggingFace datasets: https://huggingface.co/datasets/...
- Kaggle datasets: https://kaggle.com/datasets/...
```

```
- UCI ML Repository: https://archive.ics.uci.edu/ml/datasets/...
- Zenodo data repositories: https://zenodo.org/record/... (with data files)
- Direct download links: ending with .zip, .tar.gz, .csv, .json, .parquet for datasets
- Government/academic data portals with actual dataset downloads

2. CODE REPOSITORIES WITH DATASETS (Score 1.0-5.0):
- GitHub repositories that specifically host datasets in their repo (data/ folder, dataset files), and that they host datasets can be extrapolated from the context

3. RESEARCH/DOCUMENTATION PLATFORMS/PUBLISHER/JOURNAL WEBSITES (Score 0.0-1.0):
- ArXiv papers: https://arxiv.org/...
- Research project homepages without direct data access
- Papers With Code project listings (unless direct dataset link)
- General documentation or tutorial websites
- Social media or blog posts
- Journal article pages (Nature, IEEE, ACM, etc.)
- Publisher websites without dataset access
- Paywalled content without data downloads

4. UNCERTAIN/INACCESSIBLE (Score 0.0-3.0):
- URLs returning 404/403 but context suggests they were dataset links
- Ambiguous URLs where purpose cannot be clearly determined
- Private/restricted access sites where dataset nature is unclear

IMPORTANT:
- GitHub repositories should score higher if they clearly host datasets, not just code
- Consider the context: if paper mentions "we used dataset X from GitHub repo Y", it might be legitimate
- Zenodo and institutional repositories should generally score high if they contain data
- Focus on whether DATA is accessible, not just whether it's a "proper" dataset platform

Rate this URL from 0 to 5 (decimal scores encouraged), and provide a brief explanation.

Respond in the following format:
Score: [decimal number between 0-5]
Explanation: [your explanation]
"""
```

网站内容分析 **BeautifulSoup + LLM (0-5 分)**

- 访问实际网站
- 让大模型分析页面内容
- 确认是否为数据集

相应的 AI prompt 如下：

```
prompt = f"""
You are an expert in identifying dataset websites. Please analyze the following webpage content and determine if this website provides access to datasets or data repositories.

URL: {url}
Page Title: {page_title}
Number of potential data file download links: {download_links}
Data-related keywords found: {data_keywords}

Page Content (first 3000 characters):
{page_text}

Please evaluate this webpage based on the following criteria: (The more likely it is to provide datasets, the higher the score)
- Does the page provide direct access to downloadable datasets?
- Are there clear instructions or links to access datasets?
- Is the content focused on datasets or data repositories?
- Is the page from a reputable dataset platform or repository?
- Does the page have clear documentation or metadata about the datasets?

Please rate this webpage on a scale from 0.0 to 5.0 based on its usefulness for obtaining datasets, using the following scoring system:

DATASET REPOSITORIES (Score 2.5-5.0):
- Dedicated dataset platforms (HuggingFace, Kaggle, UCI ML Repository, etc.)
- Government/academic data portals with downloadable datasets
- Research data repositories (Zenodo, Figshare) with actual data files
- Pages with direct download links for data files (.csv, .json, .zip, etc.)
- Dataset documentation with clear access instructions
- Database dumps or API endpoints for data access

CODE REPOSITORIES WITH DATA (Score 2.5-5.0):
- GitHub/GitLab repositories specifically hosting datasets
- Research projects with data folders and dataset files
- Open source projects primarily for data sharing
- Repositories with dataset releases or data downloads

NON-DATASET CONTENT (Score 0.0):
- General software repositories without data focus
- Commercial websites with limited data offerings
- Blog posts or news articles mentioning datasets
- Social media or forum discussions about data
- Educational content not specifically about datasets
- Completely unrelated content (entertainment, personal blogs, etc.)
- Error pages or broken websites
- Paywalled content without clear data access
- General business websites
- Spam or malicious content

```

```
EVALUATION GUIDELINES:
1. Focus on whether actual data/datasets can be obtained from this page
2. Higher scores for direct download capabilities
3. Consider the quality and relevance of the dataset content
4. Academic and research contexts should be weighted positively
5. Clear documentation and accessibility increase the score
6. Multiple data formats or large datasets indicate higher value

Rate this webpage from 0.0 to 5.0 and provide a brief explanation focusing on what makes this page useful (or not useful) for obtaining datasets.

Respond in exactly this format:
Score: [number between 0.0-5.0]
Explanation: [your reasoning in 1-2 concise sentences]
"""
```

关键词匹配　规则评分 (0 分或 5 分)

- dataset, kaggle, data 等关键词匹配
- github.com/datasets
- 其他数据集平台标识

最终决策　阈值设定：将所有分数相加，总分 >= 5 分 -> 判定为数据集 URL。（在实操上，下面的关键词匹配融合进了网站内容分析中，因而阈值只设为了 5 分）

在实践中，我们发现一些 URL 可能受 Cloudflare 保护无法登入，抑或就是已无法连接上。对于这些 URL，它们的网站内容评分作废了，相应地，阈值也会降低一半。

1.5.4 去重

提取出的数据集 URL 可能有重复, 如:

```
https://www.robots.ox.ac.uk/~vgg/data/fgvc-aircraft/
www.robots.ox.ac.uk/~vgg/data/fgvc-aircraft/
```

解决方案　我们设计的算法如下：

1. 设定一个相似度阈值 (实操中设为了 0.8)，对每两个 url 做检验，如果两个 url 最长相同子串长度占比超过这个阈值就记录下来
2. 记录下来的 url 对让 ai 看看是不是真重复了
3. 如果真重复了就删掉其中任意一个

另外一个较明显的思路是进入两个 URL 分别的网页比较相不相同，但这样的方法是有局限性的：比如两个 URL 分别指向了同一个项目的首页和代码页面，这也算重复，但是这种方法就判断不出来。

因此，我们自行设计了这样一个能更好地保证正确率的算法。

1.6 成果展示

1.6.1 代码的其他实现细节

我们使用了 logging 库代替 print 已更美观地使程序输出。同时，实施了完备的 checkpoint 机制，使得程序无论跑到哪里终止了，下一次都能从上次没跑完的地方开始接着跑。

1.6.2 运行示例截图

下图为处理一个 pdf 时的示例截图。由于我们之前已使用了 magic-pdf 将该 pdf 转文本，所以此处没有 magic-pdf 的输出：

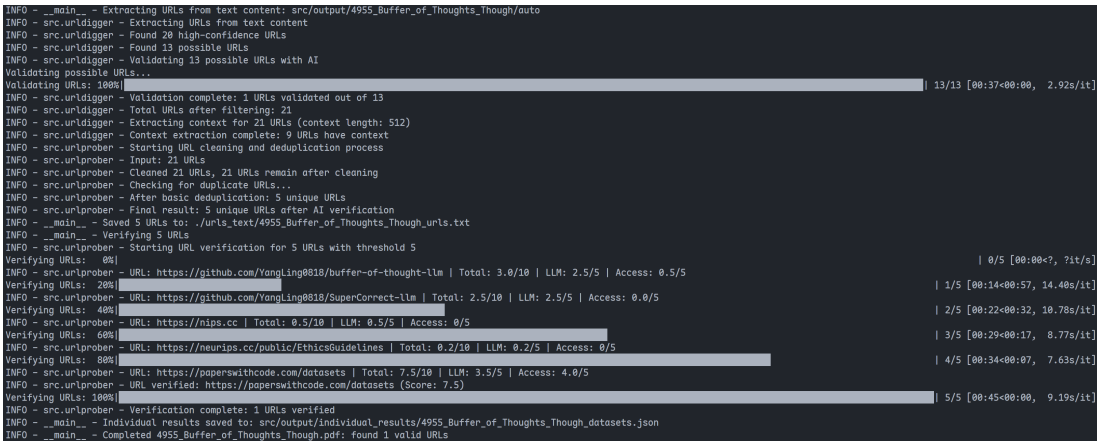

图 5: 运行截图

1.6.3 提取结果示例

我们的架构从”Flipped Classroom: Aligning Teacher Attention with Student in Generalized Category Discovery”一文中提取出的结果:

```
https://www.cs.toronto.edu/~kriz/cifar.html
https://www.kaggle.com/c/imagenet-object-localization-challenge/overview/
https://www.image-net.org/download.php
https://www.vision.caltech.edu/datasets/cub_200_2011/
https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset
https://www.robots.ox.ac.uk/~vgg/data/fgvc-aircraft/
www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset
www.cs.toronto.edu/~kriz/cifar.html
```

经验证, 基本符合人工寻找数据集的结果。

最终运行时的一些问题 在运行提供的五十篇论文时，我们发现有两个 URL 难以判定到底算不算数据集 URL：

```
https://huggingface.co/datasets
https://paperswithcode.com/datasets
```

它们没有提供具体的数据集，但确实是大批数据集的索引。因此我们最终还是判定这两个 URL 属于我们想要的结果。

此外，有一些链接指向了某个项目的主页，然后在这个项目内部的某个子目录下有数据集。对于这样的链接，我们也认为其符合要求。

1.7 代码的运行

准备工作 在 src/apikey.txt 中直接放入 openai apikey，在代码根目录下放入“课程作业论文”文件夹。

此外，使用./install.sh 安装好环境：

```
./install.sh
```

实际运行 使用项目根目录下的`run.py`即可一键跑通：

```
python run.py
```

1.8 总结

1.8.1 三大部分

- **Paper Preprocessor:** 尽可能准确地将 pdf 转文本
- **URL Digger:** 混合匹配策略, 正则 +AI 双重保障
- **URL Evaluator:** 三维评分机制 + 去重, 得到真正的数据集 URL

1.8.2 未来展望

- 使用更好的大模型进一步增加准确性
- 支持更多文档格式（Word、LaTeX 等）
- 优化 LLM 调用策略，降低成本
- 优化算法，提升整体处理效率

1.9 小组分工

- **王思宇** 撰写 URL Digger，提出了项目整个架构，运行调试，跑测试，最终整合与修改，撰写实验报告，绘制项目流程图，制作汇报 PPT，汇报人
- **薛佳音** 撰写 URL Evaluator，运行调试，跑测试
- **王镜凯** 撰写 Paper Preprocessor，运行调试，跑测试