

项目总览

本项目旨在45天内复现论文“SciCapenter: Supporting Caption Composition for Scientific Figures with Machine-Generated Captions and Ratings”中提出的科学图表标题辅助编写系统^{1 2}。SciCapenter系统允许用户上传 PDF 文档，自动提取其中的所有图表及原始标题，并显示引用该图表的段落和上下文¹。系统使用先进的 AI 模型生成两个长度不同（长/短）的候选标题²，并利用 GPT-3.5-turbo 对原始及生成标题进行质量评分和解释³。用户可以在界面上直接编辑标题，并查看智能生成的标题、多维度质量检查表（如是否提及图表元素、是否包含关键结论等）及评分反馈，从而迭代完善标题写作^{1 2}。

复现此系统需要实现以下关键模块：PDF 处理与图表提取（如使用 AllenAI 的 pdffigures2 工具⁴）；图像 OCR 提取和图表分析；调用 AI 模型生成标题（如使用 GPT-3.5-turbo 或 Pegasus 模型生成短标题²）；调用 GPT-3.5-turbo 进行标题评分及解释³；实现 Web 前端（如 Streamlit 或 Flask）和后端，支持文件上传、交互操作及结果展示；以及数据存储（如 MongoDB）和系统部署等。本计划将对项目进行详细拆解，每天安排目标、实践内容、学习资源、工具安装和关键词，使学习紧密对齐项目需求，逐步搭建出完整系统原型。

每日计划表

天数	当天目标	实践内容	推荐学习资源	工具安装说明
第 1 天	明确项目任务，搭建开发环境	- 阅读项目概述，熟悉 SciCapenter 的功能模块； - 在本地安装 Python（建议 3.8+ 版本）和 Git； - 配置 Python 虚拟环境（ <code>python -m venv env</code> ），安装基本包： <code>pip install --upgrade pip</code> 。	- 观看 Python 环境搭建教程（B 站） - SciCapenter 论文摘要 ¹	Python 官网下载安装 Python 安装 Git Git 官网
第 2 天	学习 API 调用基础：GitHub 代码管理、OpenAI API 概览	- 学习使用 Git 进行代码管理； - 阅读 OpenAI 官方文档，注册获取 API Key； - 安装 OpenAI Python 库： <code>pip install openai</code> ； - 编写简单脚本调用 GPT-3.5 创建聊天示例。	- 小昇博客：OpenAI API 调用教程 ⁵ - B 站 ChatGPT Python 教程	安装 OpenAI Python 库： <code>pip install openai</code>
第 3 天	探索 GPU 资源：购买 VS 云服务	- 调研深度学习常用 GPU 型号（如 NVIDIA 3090/4090）的性能与价格； - 调研云 GPU 服务（如阿里云、AWS、谷歌云、腾讯云等）的入门流程和费用； - 比较使用免费试用资源（Google Colab、Kaggle 免费 GPU）和自购 GPU 的利弊。	- 阿里云 GPU 云服务器购买指南 ⁶ - 知乎文章：GPU 购买与云 GPU 推荐（知乎搜索关键词“云 GPU 购买”）	（如需租用）登录阿里云等平台，根据需求选择 GPU 实例配置 ⁶
第 4 天	PDF 解析基础：使用 PyMuPDF 提取图像	- 安装 PyMuPDF： <code>pip install PyMuPDF</code> ； - 学习使用 PyMuPDF（fitz 模块）打开 PDF 文件； - 练习提取 PDF 每页或指定页的图像并保存 ⁷ ； - 验证提取效果。	- CSDN 教程：PyMuPDF 提取 PDF 图像 ⁷ - PyMuPDF 官方文档或示例	安装 PyMuPDF： <code>pip install PyMuPDF</code>

天数	当天目标	实践内容	推荐学习资源	工具安装说明
第 5 天	图表提取工具：安装 AllenAI pdffigures2	- 了解 pdffigures2：功能是提取 PDF 中的图表、表格和标题； - 安装依赖：Java JDK（建议版本 8+），并下载 SBT； - 克隆 pdffigures2 仓库（ <code>git clone https://github.com/allenai/pdffigures2</code> ）； - 编译并测试命令行工具。	- 博客：pdffigures2 使用教程 ⁴ - pdffigures2 官方 GitHub 示例	安装 JDK（如 AdoptOpenJDK ）和 SBT 克隆并编译 pdffigures2 ⁴
第 6 天	应用 pdffigures2：从 PDF 提取图表和标题	- 使用 pdffigures2 命令行提取样本论文的图表和标题： <code>sbt "run-main org.allenai.pdffigures2.FigureExtractorBatchCli /path/to/pdf -o /output/path"</code> ⁴ ； - 检查输出文件夹，获取图像文件和对应的标题 JSON； - 编写简单 Python 脚本调用 pdffigures2（使用 <code>subprocess</code> ）自动化批量处理。	- pdffigures2 官方文档[0]（GitHub 主页） - 博客：pdffigures2 Python 调用示例 ⁴	—
第 7 天	图像 OCR：提取图表中的文本	- 学习 Tesseract OCR：安装 Tesseract OCR ； - 安装 Python 包： <code>pip install pytesseract pillow</code> ； - 使用 pytesseract 对 pdffigures2 提取的图像进行 OCR 识别，获取图表上的文字信息； - 实践处理两个样例图表。	- B 站教程：Tesseract OCR 安装与使用 - CSDN/Tutorial：pytesseract 使用示例	安装 Tesseract（官网二进制或包管理器安装） 安装 pytesseract： <code>pip install pytesseract</code>
第 8 天	PDF 文本处理：提取引用图表的段落	- 继续使用 PyMuPDF：遍历 PDF 文档文本； - 设计方法：查找图表编号（如“Figure 1”）并提取相应上下文段落； - 编写代码，提取每个图表引用所在的段落文本。	- PyMuPDF 文档、示例代码 - 知乎专栏：Python 提取 PDF 文本	—
第 9 天	GPT-3.5 调用：生成图表标题（长、短两种）	- 设计 ChatGPT 提示词：让模型根据引用段落和图表信息生成描述（长标题和短标题）； - 编写调用脚本：使用 <code>openai.ChatCompletion.create</code> 提交提示； - 针对一个示例图表，分别生成长度不同的两个标题； - 调整提示、参数（如 <code>max_tokens</code> ）以获得合适结果。	- OpenAI 官方文档（Chat API 使用指南） - 博客：基于 GPT 的文本生成教程	—
第 10 天	GPT-3.5 调用：标题评分与解释	- 阅读 SciCapenter 附录中的评分提示词 ³ ； - 编写调用脚本：将引用段落和标题作为上下文，调用 GPT-3.5-turbo 进行评分和解释； - 测试：使用生成的标题和手写标题，获取模型评分和文字说明； - 评估输出合理性，调整提示语。	- SciCapenter 附录（GPT 提示词示例） ³ - B 站：GPT 评价文本质量教程	—
第 11 天	Web 前端入门：Streamlit 基础	- 了解 Streamlit：安装 <code>pip install streamlit</code> ； - 学习基本组件：如 <code>st.file_uploader</code> 、 <code>st.image</code> 、 <code>st.text_area</code> 等； - 编写第一个界面：支持 PDF 文件上传和显示； - 在本地运行 <code>streamlit run app.py</code> ，测试上传功能。	- B 站：Streamlit 入门教程 ⁸ - Streamlit 中文文档	安装 Streamlit： <code>pip install streamlit</code>

天数	当天目标	实践内容	推荐学习资源	工具安装说明
第 12 天	Web 功能：显示提取的图表和原始标题	- 在 Streamlit 界面添加文件上传后处理：调用 <code>pdffigures2</code> 或 <code>PyMuPDF</code> 提取图表； 在界面上显示提取到的第一个图表图片和原始标题文本； 添加导航（如下拉菜单）让用户切换查看其他图表； 确保每次上传都能处理多个图表。	- B 站：Streamlit 图像展示与下拉菜单示例 - ChatGPT 示例：Streamlit 文件上传	—
第 13 天	Web 功能：可编辑的标题输入区	- 在界面中为当前图表添加标题编辑框（ <code>st.text_area</code> ），预填原始标题； 用户可修改文本并提交； 添加按钮触发：将编辑后标题传回后端； 显示用户编辑后的标题文本。	- B 站：Streamlit 文本输入组件教程 - Streamlit 文档：Input Elements	—
第 14 天	Web 功能：展示 AI 生成的长/短标题	- 在界面中添加按钮：点击后调用 GPT 生成长标题和短标题； 将生成的两个标题显示在界面； 可视化效果：使用 <code>st.success</code> 或 <code>st.info</code> 区分不同标题； 验证多次调用结果的可用性。	- GPT API 调用参考博客 - B 站：Streamlit 弹窗与多列布局示例	—
第 15 天	Web 功能：GPT 评分展示	- 在界面中添加另一个按钮：提交修改后的标题给 GPT 评分； 调用评分接口，获取分数和解释； 在界面上用星级或进度条可视化评分（可用 <code>st.slider</code> 模拟）； 直接显示 GPT 给出的文字解释。	- GPT 提示词参考 3 - Streamlit 图表组件教程	—
第 16 天	后端：MongoDB 数据存储	- 学习 MongoDB：安装社区版或使用 Atlas（云端）； 安装 Python 驱动： <code>pip install pymongo</code> ； 设计数据结构：保存上传用户、PDF 信息、图表索引、原始标题、生成标题、评分等； 在后端代码中连接 MongoDB，插入和查询数据。	- B 站：Python 操作 MongoDB 教程 9 - MongoDB 官方文档	安装 MongoDB（ MongoDB 官网 ） 安装 PyMongo： <code>pip install pymongo</code>
第 17 天	Web 功能：数据持久化与历史记录	- 修改接口：每次生成标题或评分后，将结果保存到 MongoDB； 在界面上添加“查看历史”功能，展示当前文档的已有标题与评分记录； 确保刷新界面后数据依然可见； 练习查询与展示多条记录。	- Python+MongoDB 数据持久化示例 - Streamlit 表格展示教程	—
第 18 天	增加可选功能：自动检查标题要素（选做）	- 可选：使用简单规则或 GPT 检查标题是否包含关键词（如是否提及图表元素、给出结论）； 例如：调用 GPT 提问“标题是否包含图表关键元素？”并解析回答； 在界面显示检查结果（勾/叉或进度条）。	- SciCapenter 视觉要素检查表格示例 1 - 文本分类相关教程	—
第 19 天	界面优化：整理和美化 Web 界面	- 对 Streamlit 界面布局进行优化：使用列布局（ <code>st.columns</code> ）或选项卡（ <code>st.tabs</code> ）； 添加系统 Logo 或项目标题； 添加提示文字和帮助链接； 处理异常输入，提升用户体验。	- B 站：Streamlit 进阶组件教程 - UI 设计参考（论文界面截图）	—
第 20 天	集成测试：端到端演示	- 从上传 PDF 到输出各项结果，完整运行一遍系统； 测试多个示例论文和不同类型图表； 修复整合过程中发现的 Bug 和性能问题； 确保每个模块稳定运行。	- 测试用例收集文档 - Streamlit 调试指南	—

天数	当天目标	实践内容	推荐学习资源	工具安装说明
第 21 天	系统部署准备	- 研究部署选项：Streamlit Sharing、Heroku、阿里云服务器等； - 如使用 Heroku：安装 Git、Heroku CLI； - 创建项目的 requirements.txt； - 准备上线 Demo 版本。	- Streamlit 分享平台文档 - Heroku 部署 Python 教程	安装 Heroku CLI (可选)
第 22 天	部署尝试 (选做)	- 将项目推送到 GitHub； - 尝试在 Streamlit Sharing 或 Heroku 上部署； - 解决依赖和环境变量问题； - 确认云端可访问系统 Demo。	- Streamlit 部署实例案例 - Heroku 日志排查教程	—
第 23 天	撰写示例文档与代码注释	- 完善代码：添加注释、模块说明； - 撰写 README：项目简介、功能说明、运行步骤； - 准备项目演示 PPT：系统架构图和主要功能示例； - 练习演示操作流程。	- Markdown/README 编写指南 - PPT 制作技巧视频	—
第 24 天	Demo 准备：选择案例论文	- 在自己的研究领域选取1-2篇有代表性的论文； - 运行系统生成图表标题和评分； - 核对生成结果与原文标题，准备差异讨论； - 截图并记录流程，用于组会汇报。	- 参考 SciCapenter 用户研究思路 2 - 论文图表分析方法	—
第 25 天	演示与答辩准备：演示案例完善	- 继续制作 PPT：添加示例论文片段、系统界面截图与流程图； - 准备说明：系统各模块功能、调用的 AI 模型及其作用； - 练习演示：控制流程介绍和演示问答。	- PPT 制作参考 (学术报告模板) - 组会演讲技巧博客	—
第 26 天	功能完善：根据反馈改进	- 邀请同学/导师试玩系统，收集反馈； - 根据反馈对界面/功能做最后优化（如响应速度、提示文字等）； - 修复剩余问题； - 最终测试完整流水线。	- Feedback 收集方法 - 项目优化迭代案例	—
第 27 天	项目复盘与总结材料准备	- 整理项目过程文档：记录关键技术选型和实现细节； - 完成项目报告或组会文档：说明项目目标、方法、结果、心得； - 准备答辩时可能的问题和答案。	- 科研报告写作指南 - 相关论文笔记模板	—
第 28 天	系统优化 (可选)：尝试更强模型	- 如有余力：尝试使用更强的模型（如 GPT-4V、Huggingface 图表摘要模型）优化标题生成； - 比对不同模型效果，记录差异； - 准备分析结果。	- Hugging Face 模型库 - GPT-4V 视觉示例博客	—
第 29 天	系统文档完善	- 整理代码注释与说明：确保每个功能模块都有说明； - 更新 README 与开发笔记； - 整理数据和示例：确保演示用数据完整； - 整理参考资料与引用。	- 文档写作规范 - 项目发布指南	—
第 30 天	最终汇报彩排	- 进行完整的汇报彩排：演示系统操作，讲解设计思路； - 收集导师或同学的建议，做最后调整； - 准备好线上/线下组会所需资源（PPT、代码链接、环境说明等）。	- 组会汇报技巧视频 - 演示提问技巧	—

天数	当天目标	实践内容	推荐学习资源	工具安装说明
第 31 天	预留调整日：修复遗漏或深入学习新技术	- 根据彩排反馈，对系统或演示内容进行补充改进； 可学习高级功能：如图像分析模型或前端特效； 持续测试系统稳定性与完整性。	- 相关技术文档或进阶教程	—
第 32 天	预留调整日：检查部署与兼容性	- 如果已部署云端，检查访问情况； 解决不同平台的兼容性问题（Python 版本、依赖等）； 确保项目在演示环境正常运行。	- 部署故障排查指南 兼容性测试方法	—
第 33 天	预留调整日：优化用户体验	- 根据实际使用情况，进一步优化界面文字、布局； 添加必要帮助提示（如操作说明）； 提高交互流畅度和反馈。	- UI/UX 设计教程	—
第 34 天	预留调整日：复现报告要求	- 检查复现结果与论文描述的差异：总结偏差原因； 补充或标注不同之处； 准备在汇报中说明可行改进方向。	- 学术写作指导	—
第 35 天	预留调整日：任务回顾与时间管理	- 回顾整个项目进度：总结阶段成果； 优化时间分配计划，确保剩余时间充足； 调整最后两周重点（演示完善、论文阅读等）。	- 时间管理技巧博客	—
第 36 天	预留：研究相关新功能或工具	- 如有兴趣：研究相关领域最新进展（如图表理解模型、可视化分析工具）； 记录有价值的想法或工具，以备今后扩展。	- 相关领域综述文章	—
第 37 天	预留：系统性能优化（可选）	- 如果使用的模型较慢，可尝试优化代码或使用加速工具（如异步调用）； 调研 GPU 加速或并行处理技术，必要时应用。	- Python 性能优化教程	—
第 38 天	预留：阅读其他相关论文	- 查阅 SciCap 相关文献，例如 SciCap 数据集、模型、竞赛等； 吸收灵感，为系统后续改进做准备。	- SciCap 竞赛网站 相关论文综述	—
第 39 天	预留：答辩准备	- 制定答辩提纲：预想问题及答案； 复习系统技术细节和相关背景知识； 确保逻辑清晰地向他人讲解项目。	- 科研答辩指南	—
第 40 天	预留：最终检查与整理	- 最后核对项目文件：代码、文档、示例数据是否齐全； 确认所有链接（如 GitHub、Demo）正常； 备份项目成果。	- 项目交付检查清单	—
第 41 天	预留：休整与轻松复习	- 放松心情，回顾学习内容； 轻度复习项目重点，以便精神饱满地准备汇报。	- 学习放松技巧	—
第 42 天	预留：参加组会汇报	- 将项目介绍给同学/导师：演示系统功能，说明技术要点； 记录听众提问，为后续改进提供线索。	- PPT、演示系统	—
第 43 天	预留：根据组会反馈最后完善	- 根据演示反馈，对 PPT 或系统小问题作最后调整； 写下演示心得和改进计划； 准备总结文档以留存。	- 项目总结博客指南	—

天数	当天目标	实践内容	推荐学习资源	工具安装说明
第44天	预留：项目归档与文档提交	- 整理所有代码和文档到项目仓库； - 提交报告或源码至课题组指定平台； - 完成项目周期总结。	- 代码托管指南	—
第45天	预留：缓冲与放松	- 若前面步骤均完成，进行适当放松； - 如果有遗漏点最后补充； - 为下阶段研究储备精力。	—	—

工具清单

- **Python 3.8+**：用于开发环境（官网下载安装 [Python](#)）。
- **pip**：Python 包管理工具，用于安装库（随 Python 自带）。
- **openai Python 库**：调用 ChatGPT API（安装 `pip install openai`）。
- **PyMuPDF (fitz)**：PDF 解析与渲染库（`pip install PyMuPDF`）⁷。
- **AllenAI pdffigures2**：提取 PDF 图表的工具（需安装 Java JDK 和 SBT，参考 [17 + L34-L43]）。
- **Tesseract OCR**：图像文字识别引擎（安装 Tesseract 二进制，Python 调用库 `pip install pytesseract`）。
- **Streamlit**：快速构建 Python Web 界面框架（安装 `pip install streamlit`）或 **Flask**（可选）（`pip install flask`）。
- **MongoDB**：文档型数据库，用于存储结果（可使用 MongoDB Atlas 云服务或本地社区版，[MongoDB](#)）。
- **Git**：版本控制工具，用于代码管理（[Git](#) 下载）。
- **(可选) Heroku/云服务器 CLI**：部署工具，如 Heroku CLI，可按需安装。

推荐链接列表

- **SciCapenter 论文 (CHI EA 2024)**：论文 PDF 和 HTML ¹ ² ³。
- **pdffigures2 GitHub 仓库**：[allenai/pdffigures2](#) 和使用教程 ⁴。
- **OpenAI 官方文档**：OpenAI API 使用指南（ChatGPT、GPT-3.5）(<https://platform.openai.com/docs>)。
- **Python+OpenAI 教程**：小昇博客 ChatGPT 调用教程 ⁵。
- **Streamlit 入门教程**：B 站视频《入门即可搭建AI应用的框架 Streamlit》及文档。
- **Flask 框架教程**：B 站快闪 Flask 项目实战 ¹⁰。
- **MongoDB 教程**：B 站《Python 保姆级 MongoDB 快速入门》¹¹。
- **Tesseract OCR 教程**：B 站 Tesseract 安装与使用视频 ¹²。
- **GPU 云服务**：阿里云 GPU 实例创建指南 ⁶。
- **PPT 制作**：GitHub 学术报告模板或 B 站演示技巧视频（关键词：科研PPT制作）。
- **项目管理**：GitHub/其他代码托管平台使用指南。

以上计划充分覆盖了 SciCapenter 系统复现所需的各个技术模块，从基础环境搭建到模型调用、前后端开发、部署和汇报，每天有针对性的实践任务，并配以优质学习资源和检索关键词，帮助用户循序渐进地完成项目目标 ¹ ² ³。

- 1 2 3 [2403.17784] SciCapenter: Supporting Caption Composition for Scientific Figures with Machine-Generated Captions and Ratings
<https://ar5iv.labs.arxiv.org/html/2403.17784>
- 4 Using pdffigure2 to extract images from publications
<https://titipata.github.io/2016/11/06/pdffigure2.html>
- 5 ChatGPT 教程 (Python 调用 OpenAI API) - 小昇的博客
<https://xiaosheng.blog/2023/05/04/how-to-use-chatgpt>
- 6 购买GPU实例_GPU云服务器(EGS)-阿里云帮助中心
<https://help.aliyun.com/zh/egs/user-guide/create-a-gpu-instance>
- 7 Python PyMuPDF 提取PDF文件中的图片到指定目录保存，PDF将每一页都保存为图片_python提取pdf中的图片并按标题保存-CSDN博客
<https://blog.csdn.net/x1131230123/article/details/128390999>
- 8 入门即可搭建AI应用的框架Streamlit | 文生文、图生文 - Bilibili
<https://www.bilibili.com/video/BV1T5CJY7ECQ/>
- 9 11 Python保姆级MongoDB快速入门教程_哔哩哔哩
<https://www.bilibili.com/video/BV1gV411H7jN/>
- 10 【最快速度搞定Flask-框架教程】用5小时讲完的python-flask项目实战 ...
<https://www.bilibili.com/video/BV11PoTYkEE1/>
- 12 文本的检测、识别实战：使用Tesseract 进行OpenCV OCR 和文本识别
<https://www.bilibili.com/read/cv14112976/>