

文生视频演示项目

关键词生成文物解说视频

汇报人：董相宏

时间：2025/09/21



CONTENTS

目录

01 项目架构

03 项目成果

02 难点与解决方案

04 未来优化方向

01

项目架构



顶层pipeline依次调用各功能模块

...

01

深度搜索

使用wikipediaapi工具抓取文物的中文维基百科作为解说词来源。

使用ddgs工具下载相关图片用于后续视频生成。

02

解说词生成

使用ollama部署本地大模型，将上一步搜索得到的信息按章节整合为完整的解说词。

03

文字转语音

使用Coqui TTS文字转语音工具将解说词分章节转为音频

04

语音+图片拼接视频

使用moviepy工具将图片与音频拼接成完整的解说视频

pipeline接口设计

```
from deepsearch_module import run_search_module
from script_module import run_script_module
from audio_module import run_audio_module
from video_module import run_video_module
```

```
if __name__ == "__main__":
    query = input("请输入要生成视频的关键词: ").strip()
    if not query:
        raise ValueError("关键词不能为空")
```

```
sections_json, images = run_search_module(query)
```

```
script_list, script_json = run_script_module(sections_json)
```

```
audio_files = run_audio_module(script_json)
```

```
output_video = run_video_module(script_json)
```



02

难点与解决方案



短时间，弱基础下完成项目



核心 难点

资源紧张

在共计不到8h时间内需要熟悉工具，分块调试，时间非常紧张。

无GPU，仅在CPU上运行，无法使用效果更好的工具。

解决 方案

选择低成本工具

本着以最小成本完成项目的核心目标，在检索模块只进行了维基百科抓取和图片下载。在解说词生成模块选择使用轻量的本地qwen模型。在视频生成模块选择使用moviepy进行简单拼接。

03

项目成果



完成基本要求，设计好优化方案

生成解说视频

输入关键词后等待约5min，即可生成一版解说视频。全流程自动化，步骤进度可见。最终生成的视频按章节划分为若干幕。

便于优化的架构

采用模块化思想将各个功能模块隔离，该架构便于后续的优化。

04

未来优化方向



使用agentscope框架



尝试与探索

在开始项目时，第一时间去学习agentscope框架。然而因为缺乏项目经验，在跑最小demo时常常因为agentscope规定的formatter格式，reactagent对象，toolkit工具函数等概念而出错，后来选择放弃使用agentscope架构。



未来优化

agentscope架构中的多agent沟通架构和messagehub消息路由机制非常完善，快捷，在时间充裕的情况下做到掌握该工具，对于这一类agent开发项目大有裨益。



资源充足情况下使用更好的开发工具

...

检索

用RAG等更强大的检索工具来搜集原始信息。



模型

改用更强的本地大模型或者云端部署。



视频生成

使用diffusion工具生成高品质视频。



细节完善



01

添加字幕

为视频添加内嵌或外挂字幕

02

优化解说词

优化prompt，增强解说词的真实性，逻辑性，描述性

03

提高画质

当前视频画质为720p，还可进一步提升



Thank you.

感谢您的观看

汇报人：董相宏

时间：2025/09/21

