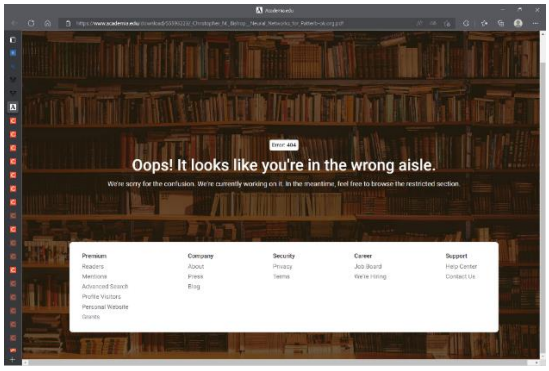
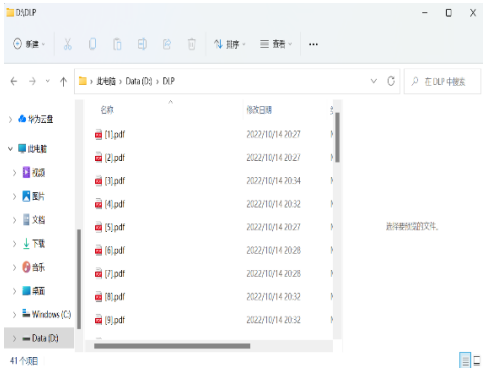
一． 任务说明

本次任务实现的主要功能为论文解析器。即：输入一篇文章的 PDF 文件，从中解析出所有的参考文献，并将每篇文章的 PDF 文件自动下载到选定的目录。具体可见演示视频。

注：演示视频中，存在部分论文未能下载的情况。这与网络不稳定等因素有关，程序中进行了异常处理，且通过手动查看，可以证明这不是程序设计缺陷造成的。（见下图）通过多次点击“爬取论文”按钮，多次执行程序，可以得到所有的参考文献。





二． 实验细节：

1. 文件解析：

使用第三方库 PDFminer.six 实现相关功能。

PDFminer.six 简要介绍：

PDFMiner 是一个可以从 PDF 文档中提取信息的工具。与其他 PDF 相关的工具不同，它注重的完全是获取和分析文本数据。PDFMiner 允许你获取某一页中文本的准确位置和一些诸如字体、行数的信息。

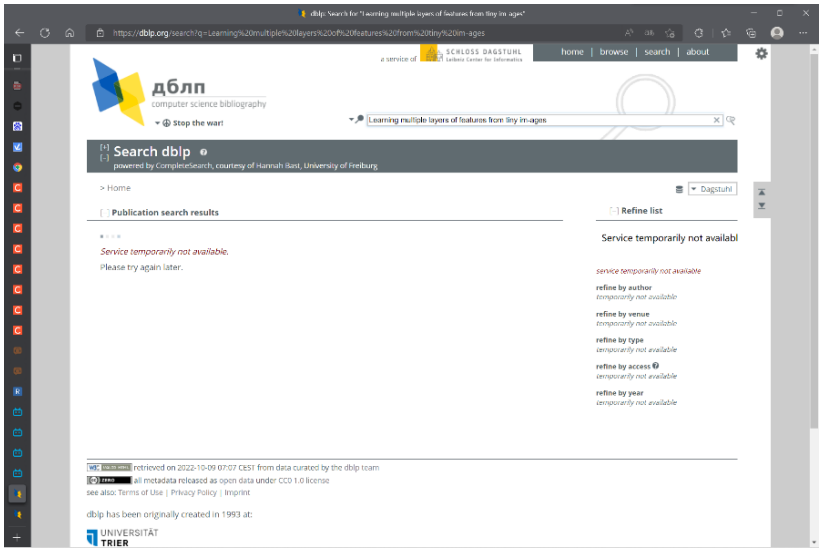
但是 PDFminer 只适用于 python2，且从 2020 年开始不再维护。PDFminer.six 是 PDFminer 的一个分支，适用于 python3，故本次作业选择 PDFminer.six 实现文件解析功能。

2. API

在项目实现过程中，我一共尝试了三个 API

a .DBLP:

最初，我尝试使用实验要求文档中所提供的 API：https://dblp.org/下载参考文献的 bib 文件。直接在该网站搜索解析出的引用条目，发现检索不到匹配项。经过询问助教，最后找到原因：该网站的字符检索过于严格（垃圾）。去除引用文献中的连字符，并调整首字母的大小写后，大部分文献可以检索到，但仍有部分文献由于网站收录不全、文章的作者在罗列参考文献时，题目与原有的题目有偏差等原因找不到匹配项。另外，该网站在使用过程中出现不稳定现象，检索 Publication 功能不能正常运行（我强烈怀疑它针对我，见下图）最终决定选用其他 API。



b. Google Scholar

由于谷歌学术有强大的关键字检索能力以及更广泛的文章收录，我尝试从 Google Scholar 爬取参考文献的 PDF 文件。但发现 request 返回的内容，与网页源代码不同。从部分博客中查找原因，某些网站会因为不是 js 解析等原因，会出

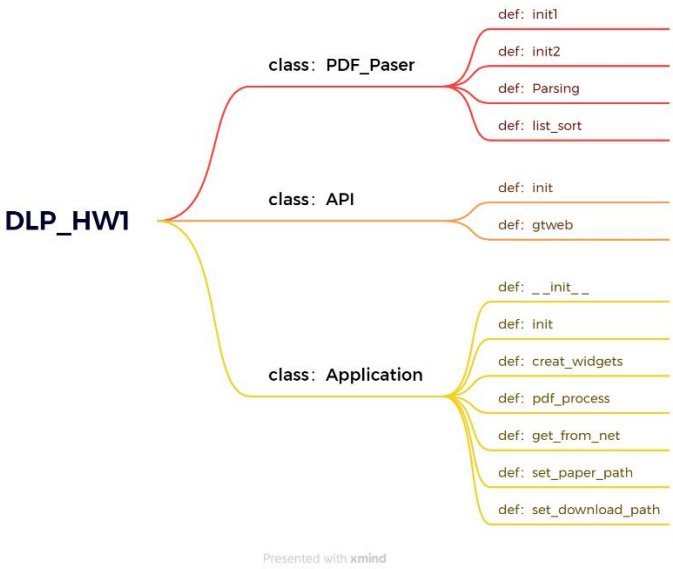
现类似情况。按照博客在 headers 中添加 Accept_Language 和 cookie，问题仍未得到解决。进一步了解之后，发现 google scholar 需要使用 selenium 库并配备相应的驱动，故放弃。

c. 熊猫学术

熊猫学术为谷歌学术的镜像网站，url 为：https://sc.panda321.com。该网站具有较强的检索能力，并且使用 request 库可以成功爬取网页数据并进行数据解析。故最终使用这个网站，爬取参考文献的 PDF 文件。

3. 代码框架

代码主要结构如图：



（1） PDF_Paser：

调用 PDFminer.six 库，解析出 PDF 文件中的文本信息，使用者正则表达式，将所有的 reference 解析出来。

（2） API：

使用 requests 库，使用 get 方法向 url 请求数据（get 方法的参数包括要搜索的内容）。得到的返回值为对应的网页数据（text/html），使用 bs4 解

析数据，抽取出 PDF 文件资源对应的 url。向新的 url 发起 get 请求，得到对应的 PDF 文件。

（3） Application:

使用 tkinter 库，实现项目的图形用户界面。并调用上述两个类，实现相关功能。

三．心得体会

1. 爬虫和正则表达式:
   通过此次作业，我初步的了解了一些有关爬虫的概念和简单方法，包括。
   此外，PDF 解析功能使我初步掌握了简单的正则表达式，并了解到他在算法中的广泛应用。
2. 关于图形用户界面
   按照我的最初构想，由于本项目的界面需要有大面积的空白用以清晰的显示文献信息，不适合使用背景图片等手段过度美化，所以我觉得 PyQt5 有点大材小用，选择了比较简单的 tkinter。直到把图形界面完成我都还十分自信，但当把各个类整合在一起，并通过 GUI 调用各个模块实现相关功能时，tkinter 中缺少线程，信号、参数传递困难的问题就显露了出来，必须通过相对复杂的逻辑转换来弥补，因此增加了好几个函数来进行参数传递和相互调用。这让我对 PyQt5 和 tkinter 之间的差距有了深深的体会。由于时间原因（还有好多其他 ddl），本次没有改用 PyQt5，下次一定！（不要扣我分，卑微）。