现代数据库系统概论 Project 1 实验报告

网络爬虫与信息抽取

聂浩 2017312153

任务一 网络爬虫

本实验使用 python 3.5 完成，依赖的包有 re,bs4, urllib3, json , xmltodict。

其中使用 urllib3 获取 html 信息（没有考虑 js 的解析问题），使用 beautiful soup 进行 htnl 解析，再利用 re 进行正则匹配，最后将类转换为 dict 并存储为 json。

整个流程中先从网站（戏考网）获取艺术家列表，之后遍历列表进行爬取。由于网站会将大量爬取视为恶意流量，因此实验中分七次进行爬取，获得的数据 data1.json~data7.json 可见 json/目录，代码可见 分任务代码/spider,py 。

最后通过 分任务代码/Xmlgen.py 进行合并，并通过 xmltodict 转换为 xml 文件，生成的 data.json 和 data.xml 分别可见 json 和 xml 文件夹。

任务二 命名实体识别

本实验使用 python 3.5 和 Stanford CoreNlp 完成。依赖于

python 包 stanfordcorenlp ，同时需要将 corenlp（https://github.com/Lynten/stanford-corenlp）对应的三个 jar 文件放在 lib 文件夹下（这一 python 接口对中文包名的匹配规则为 stanford-chinese-corenlp-YYYY-MM-DD-models.jar，必须改为这一形式）。

该任务通过调用 corenlp 的 ner 模块实现，可见 分任务代码/name.py,生成的 name.json 与 name.xml 分别可见 json 和 xml 文件夹。

Stanford CoreNlp 基于 java 构建而成，因此在前期，我基于 maven 进行了相程序 ,但是 java 版本使用 annotators 进行解析时，会执行已定义的所有解析方法，效率很低；与之相对的是，python 接口可以一次只调用一种解析方法。同时 java 的 xml 和 json 的解析速度也原逊于 python，最后不得不换成 python 环境，这浪费了大量的时间。这部分工作可以参见 java 文件夹。

任务三 实体关系抽取

本实验仍基于 python 3.5 和 CoreNLP。

这里使用了一种较为笨拙的方案——使用正则表达进行关键字过滤，再通过 CoreNLP 识别相应词性（ner 方法）的词语，以此获得目标关系。有趣的是，在学校/科班信息处理时，由于 CoreNLP 的组织名识别能力很差，但是考虑到大部分的学校名中不含有动词和助

词，通过 CoreNLP 的 pos 方法可以很大程度提高过滤的准确性。实现代码见 分任务代码/relate.py,相应 relate.json 和 relate.xml 可见 json 和 xml 文件夹。

CoreNLP 提供了 parse 功能 ,通过 parse 树进行语义判断显然会有更高的准确度。然而 CoreNlp 提供的 parse tree 转换成了字符串，难以进行解析，由于时间所限，没有进行进一步的研究。

根目录下的 test.py 即为测试程序。

心得

目前开源 NLP 项目并不是非常成熟，CoreNLP 准确性较高，但是内存、CPU 占用非常大 ;NLPIR 开发难度较高 ;它们的 API 和 wiki 文档建设都有待提高。这一专业方向还有很大的发展空间。

京剧是我国国粹，但是通过各网站可以看出名家在 300~500 人左右（本实验中抽取的大部分为业余票友），传统文化的脆弱远胜于我们的想象。

最后，在程序开发中平台的选区很重要，良好高效的开发环境可以事半功倍，前期的调研非常重要。