work record

这周的任务

1. 日期需要格式化
2. 表格需要加索引
3. 代理池ip需要优化
4. 中小企业的代码修改
5. 东方财富和公总号更新爬取 东方财富,交易所数据的更新爬取
6. 数据页面展示
7. 推荐字段名(mongodb?) 代码 公司 产品 行业 概念 关联企业 关键词库
8. 新浪财经 公司资讯(个股资讯 行业资讯) 研究报告 行业研究 同花顺 A股 金融界 7个字段
9. 类似财新网数据爬取的问题
10. 搜狗行业网站的xpath生成

4/28

1. 定向 自媒体+APP 行业
2. 今日头条 微博 微信 东方财富
3. 选股宝 右上角 股票 标签
4. APP数据的抓取

5/6

1. 标题 摘要 正文 关联
2. 原创 早晚报 每日复盘 龙虎榜 公告精选 (选股宝,周四前)

5/8

1. 公司的板块及其url抓全
2. 对图片文字识别
3. 板块 板块链接 公司名称 代码 关联理由 图片url

5/20

1. (图片识别) 搜索用文字, 展示用图片

问题:

1. (ok) 类似https://xuangubao.cn/theme/24898553网站板块不一样 忽略类似网站即可

2. (ok) 把linux下面的teseract更新一下 已更新到4.0, 确实识别率提升很多

5/24

问题:

1. bidding_reason_str的varchar长度设计不合理 * 将bidding_info_xgb_plus的bidding_reason_str的varchar的长度加到400

2. 有些看似正常的中文字识别的比较差

> pytesseract.image_to_string(im0, lang='chi_sim+eng', config='--psm 6 --oem 1'), config配置为--psm 6的时候 对单个文本块的识别非常友好, 无论文本块是单行还是多行的, 配置--oem 1的意思为使用神经网络引擎的LSTM模型 --psm 6: Assume a single uniform block of text. --oem 1: Neural nets LSTM only. 经过试验比对, 传统的光学识别和神经网络引擎差距有点大!

    - 在ubuntu下百分号识别得一塌糊涂
        - 尝试替换, %常见的误识别做个列表, 如果这个字在数字旁边则替换掉

5/30

总结:

问题:

1. 在写同花顺图谱爬取的程序中遇到了selenium的一些问题, 主要是too many open files, 这意味着在初始化 webdriver 的时候会出错, 导致程序跳出, 从而会有webdriver没有referenced的错误.
    - 终于找到原因了, webdriver的关闭最好用quit, 不要用close, 否则chromedriver驱动会一直停留在进程里面.
    - lsof -p 116447 | wc -l: 可以查看进程打开的文件数
    - lsof: 显示该进程打开文件
    - wc: 计算文件数
    - (我的解决方式是给webdriver的初始化套上一个循环, 但是给其一个十秒钟的等待时间, 这样就可以避免too many open files 的错误, 并且确保了webdriver一定可以初始化成功.)

5/31

1. 同花顺产品企业图谱的逻辑分析

    1. 企业与产品之间的关系: 企业 --主营产品--> 产品
    2. 产品与产品之间的关系:

2. 制作基础表

    - 每一行为 企业点 线(类型) 产品点 线(类型) 产品点 线(类型) 产品点 ...

    - 可能出现的问题:

        1. (ok) 可能会出现无限循环的情况:
            - 产品A 上位(出) 产品B 下位(进) 产品A 上位(出) 产品B ...
            - 解: 只需要判断将要新添加的产品有没有在该行出现过, 如果已经出现过则没有必要再添加了.

6/3

1. 制作基础表的问题:

> 制作一家公司的基础表, 例子: 机器人 边长: 1 --- 条目: 23 边长: 2 --- 条目: 154 边长: 3 --- 条目: 2829 边长: 4 --- 条目: 11645 边长: 5 --- 条目: 42837

2. vim J 可以将当前行与下一行连接

6/5

- 知识:
    1. 可以用散列表的方式检查元素是否重复, dic.get(item) == True, 则是重复的, dic.get(item) == None, 则是 没有重复的, 散列表的键名为item, 值为True的方式储存元素.
    2. mkdir -p: no error if existing, make parent directories as needed
    3. chown: Change the owner and/or group of each FILE to OWNER and/or GROUP
    4. export: 用于设置或者显示环境变量

6/12

- 知识:
    1. linux:
        - nohup python3 *.py >out.log 2>&1 &: 即使关闭终端, 程序依然可以在服务器上运行, 并将日志记录在out.log
        - htop: 友好的系统进程查看命令
        - kill -9 'python3': 关闭所有包含进程名python3的进程
        - 双核 1 g: 最佳是跑5个selenium
- 任务:
    1. 学习mongodb, 将数据插入到mongodb中, 最终比较在mysql和mongodb中查询数据的速度.
    2. 思考在上亿条数据中检索的好方法
        - 解: 建一个哈希表储存对应公司的id, 这样取部分公司的时候可以根据所在id直接取出来.

6/13

- 任务:
    1. 采集与展示页面校对, 是否重复
    2. 浏览展示页面的功能是否欠缺, 是否有bug
    3. 在monodb中分别针对每个公司建立一个表
- 问题:
    1. 标题链接无法点进去
    2. 无法翻页
    3. 公告点进去会弹出报错窗口但是会有内容加载出来
    4. 公告的内容和资讯的内容重复
    5. 公告可以翻页但是公告翻页之后的内容为空
    6. 研报里面列表链接点击会弹出报错窗口且无法加载进去
    7. 观点的内容似乎有所缺失
- 总结:
    1. 自动化爬取工具
        - get
            1. 成功备份统计
                - 网站 | url | xpath | 翻页
            2. 全新测试

    - js post
2. 代码整理
3. 列个爬虫数据清单(找个软件可以画脑图也可以写markdown)数据跟踪统计
4. 去重
5. ftp 100g 删掉原来的虚拟机, 新建一个虚拟机专门用来ftp
6. mysql 机械盘扩展
7. 循环ip丢失 spider_request_file.py

6/14

- linux:
    1. 打开mongo服务:
        - cd /usr/local/mongodb/bin/
        - ./mongod --smallfiles
    2. 清屏命令:
        - printf "\033c"

6/17

- python3:
    1. python 使用 pymysql DBUtils 创建连接池, 提升性能

```
import pymysql
from DBUtils.PooledDB import PooledDB
pool = PooledDB(pymysql, 5, host='ip', user='user', passwd='passwd',
db='db', port=3306, setsession=['SET AUTOCOMMIT = 1']) # 5为连接池最小连
接数, setsession=['SET AUTOCOMMIT = 1']是用来设置线程池是否打开自动更新的配
置，0为False，1为True
conn = pool.connection()
```

        1. 在程序创建连接的时候, 可以从一个空闲的连接中获取, 不需要重新初始化连接, 提升获取连接的速度
        2. 关闭连接的时候, 把连接放回连接池, 而不是真正的关闭, 所以可以减少频繁地打开和关闭连接
- linux:
    1. 磁盘分区与自动挂载:
        1. fdisk -l: 查看磁盘分区情况
        2. fidsk /dev/sda: 对磁盘进行分区
        3. mkfs -t ext4 /dev/sda4: 对分区格式化, 只有格式化之后才能进行挂载
        4. mount /dev/sda /new_empty_dir: 手动挂载 umount /dev/sda
        5. vim /etc/fstab: 设置开机自动挂载分区
            - /dev/sda4 /home ext4 defaults 0 2: 将分区挂载在/home目录下, 之前的挂载磁盘内容会被覆盖, 所以最好提前做好备份工作
        6. fdisk默认为mbr文件系统分区限制为2TB, 可以用parted对gpt文件系统分区
        7. mbr文件系统默认主分区加上扩展分区不超过4, 扩展分区不超过1, 扩展分区包含逻辑分区, 逻辑分区的个数没有限制

2. cp -a: 复制常用的命令相当于cp -pdr, 即连同文件的属性一起复制, 若来源为连结档的属性则复制连结档的属性而非本身, * 递归的持续复制

- 问题: 1.(ok) 5亿条数据, 如何提高数据的查询速度? * 解：在建表之初就添加想要查询字段的索引

6/19

- 任务:
    1. 展示校验
    2. 查重校验
    3. 爬虫 - 去重 - 展示 整个流程的审核
    4. 展示爬虫程序和服务器运行状态
    5. 展示数据的更新状态
    6. 对发放任务的历史纪录
    7. ths_tp图表的展示

6/20

- 任务：
    1. 准备接手去重
    2. 取数据的循环逻辑最好借鉴同花顺的循环连接
- 问题 1.(ok) os.path.dirname(__file__)

6/21

- 任务:
    1. 固定ip代理服务器开selenium最多只能并发五个
    2. 故而一定要确保一个服务器的爬虫最多并发五个
    - 方案
        1. 给每台服务器标记一个值, 用于标记每台服务器爬虫的并发运行个数
        2. 每台服务器的标记值不得超过5, 初始值为0
        3. 爬虫优先获取标记值最小的服务器代理ip
        4. 每有一个爬虫获取了代理ip, 则对应代理ip的服务器的标记值+1

        ---

        1. 可以在每个服务器终端开启五个selenium客户端, 将每个selenium客户端对象添加一个属性flag, 如果在访问url, 则flag为繁忙, 反之为空闲
        2. 爬虫随机取空闲的selenium客户端
        3. 如果selenium客户端都繁忙, 则堵塞
    - 测试
        1. 对于一台服务器, 不管你怎么取代理, 确保该台服务器只有不超过5个爬虫并行操作
        2. 在redis的键值对表示服务器的标记值
            - {'服务器1': flag, '服务器2': flag, ...}
            - 初始化键值对为0
            - 当有一个爬虫任务时, 则值加1
            - 当该爬虫任务结束时, 则值减1
            - 当值为5时, 不可再有爬虫任务使用这台服务器

        - 当多台服务器运行时, 则优先取标记值小的服务器给该爬虫使用

6/24

小结:

- 可以用取模函数将爬虫均摊到每个服务器上面

页面展示问题记录

- **注**: 这里列举的是个别的股份公司, 具有一般性, 即每个公司都会有相似的问题
- 宝钢股份公告有重复内容
- 宝钢股份研报内容没有换行, 看着不是很舒服
- 宝钢股份研报内容格式有点问题
  - 例如: 研报 '*宝钢股份:2019年5月期货出厂价点评,五月出厂价稳中有涨,有望利好Q2业绩*' 里面行首有'xa0xa0xa0'这样的字符
- 宝钢股份观点的时间问题
  - 例如 '*9小时前 来自...*' 爬取的内容没有错误, 但是相对现在的时间来说明显是有问题的
- 宝钢股份观点内容有点乱
  - 例如 '*来自*' 后面的来源内容缺失
- 宝钢股份的观点没办法点进去看详细的内容
- 600782 新钢股份 资讯内容有水印
  - 例如: '*钢铁限产松动打压利润，钢贸商拿货谨慎将导致钢厂库存继续累积*' 有很难看的水印
- 新钢股份咨询标题内容有乱码部分:
  - 例如: '*基金重仓分析：1Q19钢铁减仓，建材持仓比重环比提升 - 投一把股票网*'
- 新钢股份研报内容有重复
- 新钢股份观点内容有点乱码
  - 例如: '*8小时前 来自 用4万块看了一下新钢的底牌，太烂啦。(SH)@阿懒猫...* ', 有很多条消息都有'(SH)'这样的东西.
  - 例如: 37页的 '*u003cemu003e新钢股份u003c/emu003e2018年净利55亿-62.5亿元 同比增长77%至101%*', 另外它好像不是观点的内容, 因为可以点进去, 且点进去感觉和正常的观点不一样.