Python与金融数据挖掘(8)

文欣秀

wenxinxiu@ecust.edu.cn

大数据定义

大数据（big data）： 一种数据规模大到在获取、存储、管理、分析方面大大超出了传统数据库软件工具能力范围的数据集合。它包括**结构化、半结构化和非结构化数据**，非结构化数据越来越成为数据的主要部分。

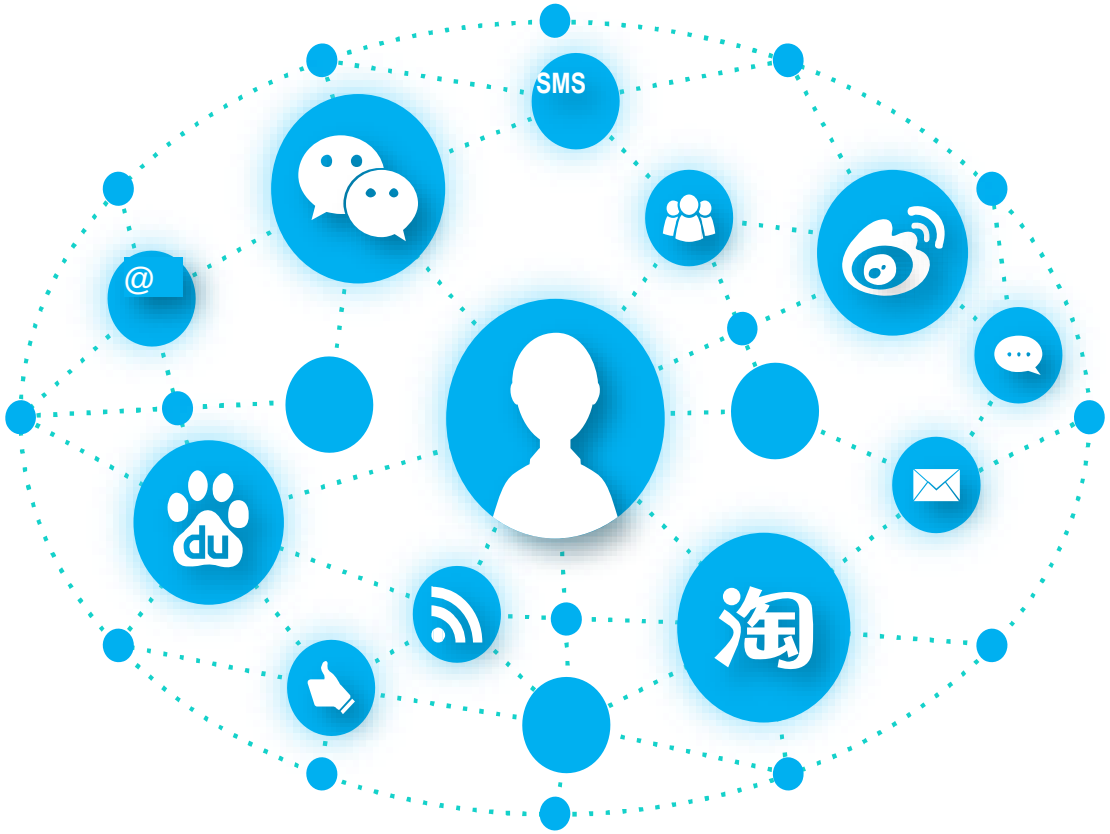
结构化数据

	A	B
1	学号	姓名
2	20002518	邹轩敏
3	21012973	李一凡
4	21012974	王绘雯
5	21012975	黄佳妮
6	21012976	李雨杉
7	21012978	胡凯
8	21012979	党嘉懿
9	21012980	马睿
10	21012981	赵骏飞
11	21012982	袁洲力

半结构化数据

```
<ul>
  <li><a href="https://qs.dfcfw.com/1606">期货手机开户</a></li>
  <li><a href="https://qs.dfcfw.com/1607">期货电脑开户</a></li>
  <li><a href="https://qs.dfcfw.com/1608">期货官方网站</a></li>
</ul>
```

非结构化数据



大数据的特点

Volume(大量)： 存储单位至TB、PB、EB级别

Velocity(高速)： 处理速度**快**、时效性要求**高**

Variety(多样)： **结构化**、**半结构化**及**非结构化**

Value(价值)： 数据价值密度**低**、需要**算法**挖掘

Python支持的数据库

- SQLite
- MySQL
- MongoDB
- Redis
- Microsoft SQL Server 2000
- ….

常用数据库

SQLite：是一个开源的关系型数据库，具有零配置、自我包含、便于传输等优点。它将整个数据库的表、索引、数据都存储在一个**单一的.db文件**中，不需要网络配置和管理，没有帐户和密码，数据库访问依赖于文件所在的操作系统。

SQLite数据库连接

- 和数据库建立连接
- 执行sql语句，接收返回值
- 关闭数据库连接

常用SQL语句

- 创建一个新的数据表

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''create table stock (code char(8) not null,
        name char(10),price float, primary key("code"))'''
conn.execute(SQL)
conn.commit()
conn.close()
```

常用SQL语句

- 往一个表中插入数据

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''insert into stock (code, name, price)
                    values('2349', '精华制药' , 10.49)'''
conn.execute(SQL)
conn.commit()
conn.close()
```

常用SQL语句

- 更新数据表中的数据

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''update stock set price=11.5
                        where code='2349' '''
conn.execute(SQL)
conn.commit()
conn.close()
```

常用SQL语句

- 从一个表中删除数据

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL=""delete from stock where code='2349' ""
conn.execute(SQL)
conn.commit()
conn.close()
```

常用SQL语句

- 删除表

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''drop table stock'''
conn.execute(SQL)
conn.commit()
conn.close()
```

交易表数据存入数据库中

链接数据库并创建表

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL= '" drop table if exists stock'"
conn.execute(SQL)
conn.commit()
SQL='"create table stock (code char(8) not null,
        name char(10),price float, primary key("code"))'"
conn.execute(SQL)
conn.commit()
```

从文件读取数据并存入数据库中

```
with open("C:\\trade.csv","r") as fobj:
    for i in fobj:
        if i[:4]=="code":
            continue
        i=i.strip();  info=i.split(",")
        SQL=""insert into stock (code, name, price)
           values('%s','%s',%f)"" %(info[0],info[1],float(info[2]))
        conn.execute(SQL)
        conn.commit()
conn.close()
```

东方财富网爬虫存入数据库中

爬取东方财富网链接和标题

```
import requests
import re
url="https://www.eastmoney.com/"
html=requests. get(url)
html. encoding=html. apparent_encoding
data=html. text
reg=r'<a href="(https://.*?)".*?>(.*?)</a>'
urls=re.findall(reg, data)
```

爬虫结果存入数据库

```
import sqlite3
conn=sqlite3.connect("web.db")
SQL='''drop table if exists information'''
conn.execute(SQL)
conn.commit()
SQL='''create table information(code integer not null,name char(30),
        link char(20), primary key("code"))'''
conn.execute(SQL)
conn.commit()
```

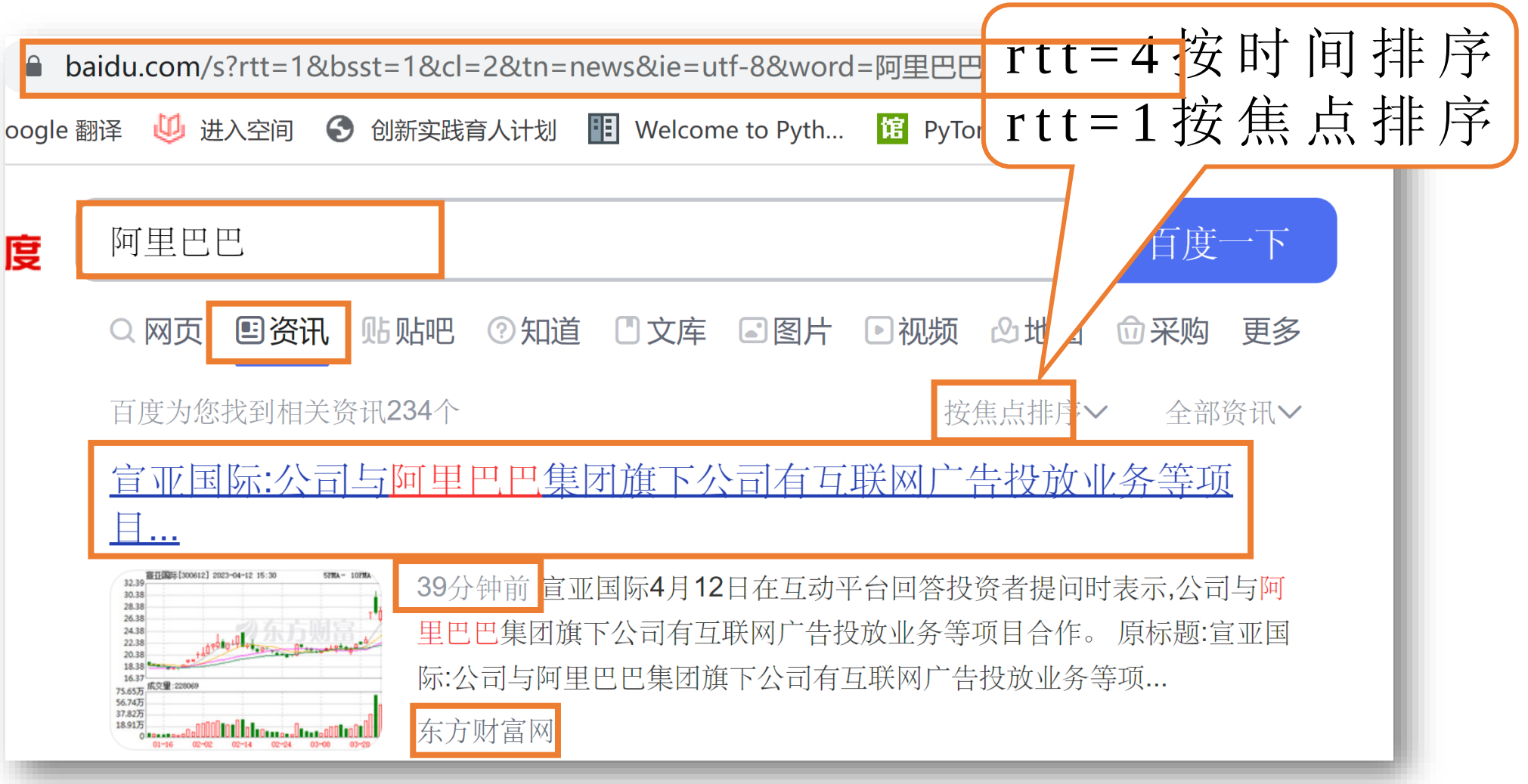
爬虫结果存入数据库

```
count=1
for item in urls:
    SQL='''insert into information(code,name,link)
                values(%d,'%s', '%s')''' %(count,item[1],item[0])
    conn.execute(SQL)
    conn.commit()
    count+=1
conn.close()
```

从数据库中查询部分记录

```
import sqlite3
conn=sqlite3.connect("web.db")
SQL=""select name,link from information where name like "东方%" ""
aList=list(conn.execute(SQL))
conn.commit()
for line in aList:
    print(line)
conn.close()
```

案例分析


rtt=4 按时间排序
rtt=1 按焦点排序
baidu.com/s?rtt=1&bsst=1&cl=2&tn=news&ie=utf-8&word=阿里巴巴
oogle 翻译
进入空间
创新实践育人计划
Welcome to Pyth...
PyTor
阿里巴巴
百度一下
网页
资讯
贴吧
知道
文库
图片
视频
采购
更多
百度为您找到相关资讯234个
按焦点排序
全部资讯
宣亚国际:公司与阿里巴巴集团旗下公司有互联网广告投放业务等项目...
39分钟前
宣亚国际4月12日在互动平台回答投资者提问时表示,公司与阿里巴巴集团旗下公司有互联网广告投放业务等项目合作。 原标题:宣亚国际:公司与阿里巴巴集团旗下公司有互联网广告投放业务等项...
东方财富网

正则表达式修饰符含义

修饰符	描述
re.I	使匹配对大小写不敏感
re.L	做本地化识别（locale-aware）匹配
re.M	多行匹配，影响 ^ 和 $
re.S	使 . 匹配包括换行在内的所有字符
re.U	根据Unicode字符集解析字符。这个标志影响 \w, \W, \b, \B.
re.X	该标志通过给予你更灵活的格式以便你将正则表达式写得更易于理解。

输出搜索到的全部链接

```
import requests
import re
import time
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36'}
def baidu(company):
    url = 'http://www.baidu.com/s?tn=news&rtt=4&wd=' + company
    res = requests.get(url, headers=headers).text
    p_href = '<h3 class="news-title_1YtI1 "><a href="(.*?)"'
    href = re.findall(p_href, res, re.S)
    print(href)
    …
baidu('阿里巴巴')
```

rtt=4按时间排序
rtt=1按焦点排序

输出搜索到的标题、日期、来源

```
    …
    p_title = '<h3 class="news-title_1YtI1 ">.*?>(.*?)</a>'
    title = re.findall(p_title, res, re.S)
    print(title)
    p_date = '<span class="c-color-gray2 c-font-normal c-gap-right-\
xsmall" .*?>(.*?)</span>'
    date = re.findall(p_date, res)
    print(date)
    p_source = '<span class="c-color-gray" .*?>(.*?)</span>'
    source = re.findall(p_source, res)
    print(source)
```

部分搜索结果展示

['<!--s-text-->宣亚国际:公司与阿里巴巴集团旗下公司有互联网广告投放业务等项目...<!--/s-text-->', '<!--s-text-->张勇:阿里巴巴所有产品未来将接入大模型全面改造<!--/s-text-->', '<!--s-text-->读特专稿|放权动真格?解读阿里巴巴“分家式”组织变革<!--/s-text-->', '<!--s-text-->小摩:阿里巴巴股价上行空间具吸引力 上季度经调整EBITA或升48%至...<!--/s-text-->', '<!--s-text-->概念动态|特发服务新增“阿里巴巴概念”<!--/s-text-->', '<!--s-text-->中金:维持阿里巴巴“跑赢行业”评级,目标价137港元<!--/s-text-->', '<!--s-text-->宣亚国际:公司与阿里巴巴集团旗下公司有互联网广告投放业务等项目...<!--/s-text-->', '<!--s-text-->中金:维持阿里巴巴跑赢行业评级 目标价137港元<!--/s-text-->', '<!--s-text-->国信证券维持阿里巴巴买入评级<!--/s-text-->', '<!--s-text-->阿里巴巴所有产品未来将接入「通义千问」,将推企业专属大模型|最...<!--/s-text-->']
['1小时前', '12小时前', '3小时前', '4小时前', '1小时前', '10小时前', '5小时前', '5小时前']

数据清洗常见方法

- 用strip()函数删除空格及换行符等非相关符号

```
>>> res='  华能信托本年实现利润32.05亿元  '
>>> res=res.strip()
>>> res
'华能信托本年实现利润32.05亿元'
```

数据清洗常见方法

- 用split()函数截取需要的内容

```
>>> date='2019-01-20 10:10:10'
>>> date=date.split(' ')[0]
>>> date     '2019-01-20'
```

数据清洗常见方法

- 用sub()函数进行内容替换

短语标签，用来呈现为被强调的文本

```
>>> import re
>>> title='阿里<em>巴巴</em>人工智能再发力'
>>> title=re.sub('<.*?>', '',  title)
>>> title     '阿里巴巴人工智能再发力'
```

搜索结果清洗及输出

```
for i in range(len(date)):
    title[i] = title[i].strip()
    title[i] = re.sub('<.*?>', '', title[i])
    if ('小时' in date[i]) or ('分钟' in date[i]):
        date[i] = time. strftime("%Y-%m-%d")
    else:
        date[i] = date[i]
    print(str(i + 1) + '.' + title[i] + '(' + date[i] + '-' + source[i] + ')')
    print(href[i])
```

如何将爬取4项数据存入数据库中？

谢　谢