# Python爬取拉勾网数据挖掘岗位信息分析

## 目的与要求

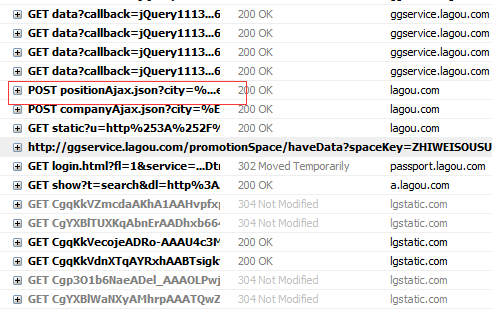
这篇文章主要是进行一次简单的Python爬虫实战演练。目的是分析数据挖掘岗位目前的一些情况。由于拉勾网是目前比较热门的互联网IT行业的招聘平台，所以小编就以拉勾网为我们爬虫的主要爬取对象。这里我们要获得一些招聘公司的基本信息，然后对其进行处理分析以及可视化。

## 分析网站的请求过程以及数据格式

我们知道所有的网站请求都是基于HTTP协议的，这里我们借助强大的Firefox浏览器来帮助我们分析网站的请求过程以及我们所需要的数据格式。



上面是拉勾网的首页，下面我们输入数据挖掘，并且利用Firebug来分析网站的请求过程。



我们通过Firebug工具，查看网络可以看到一系列的Get和Post请求链接，这里我们需要数据挖掘岗位的信息，所以只需要查看上面红色部分的Post请求链接即可即positionAjax.jsom。它的URL为：

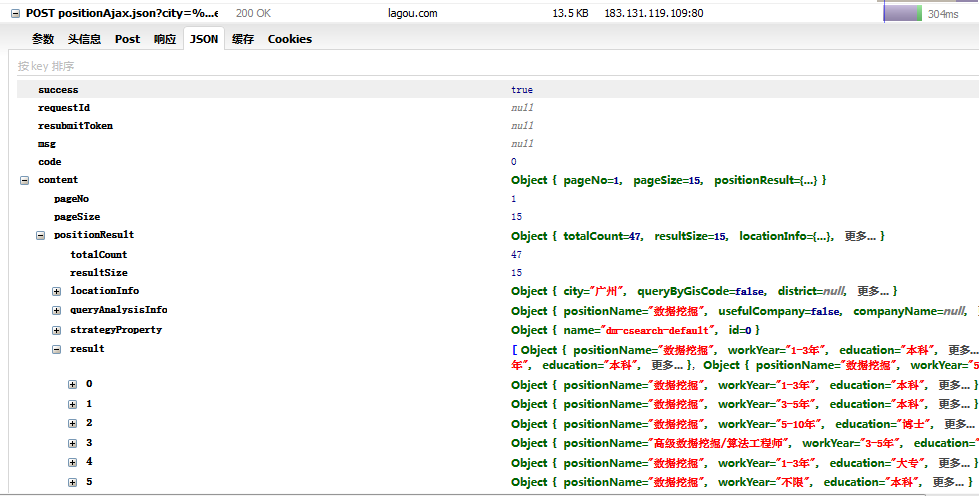
<http://www.lagou.com/jobs/positionAjax.json?city=%E5%B9%BF%E5%B7%9E&needAddtionalResult=false>

下面是它的请求头信息以及Post请求需要传递的参数：





下面是相应回来的JSON数据，里面包含我们需要的岗位信息，以及其中一条岗位的具体信息，我们可以很容易看出，每页显示15条数据，在该条件下总共有47条岗位信息，岗位信息都保存在result中：





## Python爬取数据以及保存数据到Excel

根据上面的分析，下面我们利用Python来一步步获取我们的数据。

### 3.1请求网站，获取网站信息

这里我们要利用Python模拟浏览器进行Post请求，以得到我们想要的网站数据。我们需要用到的模块有urllib和urllib2，urllib是用来编码请求参数的，urllib2用来封装头信息并且获得request请求对象，最后获得相应网站信息。

*# 得到当前页的json内容,url为爬取的链接，page\_num是爬取网站的第几页，keyword是需要爬取的关键字***def** get\_page(url,page\_num,keyword):  
  
 page\_hearders = { *# 模拟浏览器，封装头部hearder信息* **'Accept'**:**'application/json, text/javascript, \*/\*; q=0.01'**,  
 **'Content-Type'**:**'application/x-www-form-urlencoded; charset=UTF-8'**,  
 **'Host'**:**'www.lagou.com'**,  
 **'User-Agent'**:**'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0'**,  
 **'Connection'**:**'keep-alive'**,  
 **'X-Requested-With'**:**'XMLHttpRequest'** }  
  
 **if** page\_num==1:  
 flag = **'true'  
 else**:  
 flag = **'false'** data = { *# Post请求的三个参数* **'first'**:flag,  
 **'kd'**:keyword,  
 **'pn'**:page\_num  
 }  
  
  
 page\_data = urllib.urlencode(data)  
 request = urllib2.Request(url,headers=page\_hearders)  
 file = urllib2.urlopen(request,data=page\_data.encode(**'utf-8'**))  
 page\_contents = file.read().decode(**'utf-8'**)  
 file.close()  
 **return** page\_contents

### 3.2数据处理，得到所需数据

根据上面得到的网站信息，下面我们处理得到我们需要的数据。这里我们要用到的模块主要是json模块，对传进来的原始网站的json信息进行解析成Python可以解析的数据结构。

*# 得到当前页相应标签的内容,page\_content是爬取的网站内容，tag是需要爬取的标签***def** get\_real\_info(page\_content,tag):  
  
 data = json.loads(page\_content)  
 jdata = data[**'content'**][**'positionResult'**][**'result'**]  
 ls = [] *# 把我们需要的数据处理到list里,list里面封装的是字典类型数据* **for** jd **in** jdata:  
 company = {}  
 **for** t **in** tag:  
 **if 'companyLabelList'** == t:  
 **if** jd[t]:  
 company[t] = **','**.join(jd[t])  
 **else**:  
 company[t] = **'no label'  
 elif 'positionAdvantage'** == t:  
 **if** jd[t]:  
 company[t] = jd[t]  
 **else**:  
 company[t] = **'no advantage'  
 else**:  
 company[t] = jd[t]  
 ls.append(company)  
 **return** ls

### 3.3保存数据到Excel

最后我们将我们处理后的数据保存到Excel中，以便后面分析。这里我们要用到的模块主要是xlsxwriter模块：

*# 保存所有数据的信息到Excel中,all\_info是我们需要保存的数据,file\_name是需要保存的文件名***def** save\_in\_excel(all\_info,file\_name):  
  
 *# 这里我们默认保存到桌面*book=xlsxwriter.Workbook(**r'C:\Users\Administrator\Desktop\%s.xls'** % file\_name)  
 worksheet = book.add\_worksheet()  
 row\_num = len(all\_info)  
 **for** i **in** xrange(1,row\_num+2):  
 **if** i==1:  
 tag\_pos = **'A1'** worksheet.write\_row(tag\_pos,all\_info[0].keys())  
 **else**:  
 con\_pos = **'A%s'**%i  
 content = all\_info[i-2].values()  
 worksheet.write\_row(con\_pos,content)  
 book.close()

**完整代码：**

最后一个小的完整的爬虫程序已经完成，完整代码如下所示：

*# coding=utf-8*\_\_author\_\_ = **"WJW"  
  
import** urllib  
**import** urllib2  
**import** json  
**import** xlsxwriter  
**import** math  
  
*# 得到当前页的json内容***def** get\_page(url,page\_num,keyword):  
  
 page\_hearders = {

**'Accept'**:**'application/json, text/javascript, \*/\*; q=0.01'**,  
 **'Content-Type'**:**'application/x-www-form-urlencoded; charset=UTF-8'**,  
 **'Host'**:**'www.lagou.com'**,  
 **'User-Agent'**:**'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0'**,  
 **'Connection'**:**'keep-alive'**,  
 **'X-Requested-With'**:**'XMLHttpRequest'** }  
  
 **if** page\_num==1:  
 flag = **'true'  
 else**:  
 flag = **'false'** data = { **'first'**:flag,  
 **'kd'**:keyword,  
 **'pn'**:page\_num  
 }  
  
  
 page\_data = urllib.urlencode(data)  
 request = urllib2.Request(url,headers=page\_hearders)  
 file = urllib2.urlopen(request,data=page\_data.encode(**'utf-8'**))  
 page\_contents = file.read().decode(**'utf-8'**)  
 file.close()  
 **return** page\_contents  
  
  
*# 得到当前页相应标签的内容*

**def** get\_real\_info(page\_content,tag):  
  
 data = json.loads(page\_content)  
 jdata = data[**'content'**][**'positionResult'**][**'result'**]  
 ls = []

**for** jd **in** jdata:  
 company = {}  
 **for** t **in** tag:  
 **if 'companyLabelList'** == t:  
 **if** jd[t]:  
 company[t] = **','**.join(jd[t])  
 **else**:  
 company[t] = **'no label'  
 elif 'positionAdvantage'** == t:  
 **if** jd[t]:  
 company[t] = jd[t]  
 **else**:  
 company[t] = **'no advantage'  
 else**:  
 company[t] = jd[t]  
 ls.append(company)  
 **return** ls  
  
  
*# 保存所有数据的信息到Excel中***def** save\_in\_excel(all\_info,file\_name):  
  
 *# 这里我们默认保存到桌面* book = xlsxwriter.Workbook(**r'C:\Users\Administrator\Desktop\%s.xls'** % file\_name)  
 worksheet = book.add\_worksheet()  
 row\_num = len(all\_info)  
 **for** i **in** xrange(1,row\_num+2):  
 **if** i==1:  
 tag\_pos = **'A1'** worksheet.write\_row(tag\_pos,all\_info[0].keys())  
 **else**:  
 con\_pos = **'A%s'**%i  
 content = all\_info[i-2].values()  
 worksheet.write\_row(con\_pos,content)  
 book.close()  
  
*# 得到总共多少页***def** get\_totalPages(page):  
 data = json.loads(page)  
 totalCount = data[**'content'**][**'positionResult'**][**'totalCount'**]  
 totalPage = math.ceil(totalCount/15.0)  
 **return** totalPage  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 url = **'http://www.lagou.com/jobs/positionAjax.json?px=default&needAddtionalResult=false&city=全国'** tag = [**'companyFullName'**, **'city'**, **'companySize'**, **'education'**, **'financeStage'**, **'industryField'**, **'jobNature'**,**'positionAdvantage'**,**'positionName'**,**'salary'**,**'workYear'**,**'positionId'**,**'companyLabelList'**]  
 totalPages = get\_totalPages(get\_page(url,1,**'数据挖掘'**))  
 allinfo = []  
  
 **for** x **in** xrange(1,totalPages):  
 pageinfo = get\_page(url,x,**'数据挖掘'**)  
 info = get\_real\_info(pageinfo,tag)  
 allinfo.extend(info)  
 **print '爬取成功第%s页'**%x  
  
 save\_in\_excel(allinfo,**'lagouInfo'**)  
  
 **print '抓取完毕'**

## 数据挖掘岗位信息分析

数据挖掘目前是一个新型的行业，在很多人眼里它也是一个很高大上的职业，被充斥着各种头衔，那么下面就让我们来瞧瞧它到底是一个怎样的职业。

### 4.1 Excel进行基本信息的统计分析

我们通过上面Python程序的爬取，总共爬取到了全国2250条关于数据挖掘岗位的信息，包含十几个变量，我们对其进行了简单的数据清洗，进行如下的统计分析：

1. 数据挖掘岗位在全国各地招聘人数的分布

通过上图的分析，我们可以知道，数据挖掘岗位主要分布在北上广深，此外杭州、成都也有部分人数。其中北京所需要的人才超过了上海、深圳和广州的总和，高高位居榜首，由此可知，如果想要从事数据挖掘方面的工作，就离不开北上广深，它们对数据挖掘人才求贤若渴，特别是首都北京供不应求的场景十分严重，如果条件允许的话，大家大可以去北京闯一闯。

1. 数据挖掘职业类型分析

由上图可知，数据挖掘主要存在于移动互联网行业，其次是信息服务行业，电子商务和金融行业。这也不奇怪，互联网时代的到来，积存了许多数据，产生各行各业的生态圈，其中每个行业都存在着十分有用的信息有待我们去挖掘和分析，这是一个很大的趋势，数据挖掘职业是分门派的，但是其本质是不变的。

1. 数据挖掘职业招聘学历与经验分析

由上图分布可知，数据挖掘学历主要在本科，所需经验在1到5年之间，这相对其他行业来说，要求较高，而相对博士和研究生不太感冒，这主要是因为数据挖掘在商业应用上所涉及的内容和知识体系只是一些相对成熟的一块，而且大部分企业需要掌握技能和实践性较强的人才。

1. 数据挖掘岗位薪资分析

由上图分布可知，目前数据挖掘职业的薪资10K到30k居多，这大大高于其他行业的薪资水平，数据挖掘可真是一个高薪行业，不过这高薪行业可不是每一个人都能得到的，它对我们的学历和行业的经验要求很高，所需掌握的知识也比较高深。

1. 拥有数据挖掘职业的公司融资情况的分析

由上图可知，招聘数据挖掘职业最多的公司是上市公司，其次主要是成熟型的公司，这也映射了数据挖掘行业高薪背后主要是工作在一些大型的互联网企业，也只有成熟型的企业才有较大的能力和资金可以用于新领域和新知识的挖掘和研发。

1. 数据挖掘职业所需技能的分析

这里我们需要重新爬取数据，对刚刚我们爬取出来的每一条招聘信息进入岗位信息的具体爬取，因为所需技能信息在每一个岗位信息的详情页中，具体代码见附录，这里主要是分析它的信息.

有分析我们可知，sql技能是最基本的，其次需要的技能包括Python、

hadoop、java、R以及Excel，所以想从事数据挖掘行业的小伙伴门，可以从这几方面入手，当然光会技能是没用的，一些基本数据挖掘基础知识是必须的，再来就是积累经验了，不过数据挖掘是个比较难掌握的学科，大家可以慢慢来，掌握一两个工具和技能可以达到事半功倍的效果。

### 4.2 利用词云对一些文本信息进行分析

下面我们将对我们爬取出来的一些企业优势和从事数据挖掘工作的发展和福利进行分析，由于这些数据主要是文本信息，所以我们采用Python的词云来对其进行分析和可视化。这里要用到的模块主要是jieba、wordcloud和matplotlib模块，jieba主要是用来分词，wordcloud和matplotlib主要只用来绘制词云。

#coding=utf-8

import jieba

from wordcloud import WordCloud

import sys

reload(sys)

sys.setdefaultencoding('utf-8')

# alldata.txt是我们处理出来的文本信息

f = open('/home/wjw/pythonProject/lagouProject/alldata.txt','r')

content = f.read()

f.close()

s = {}

contents = jieba.cut(content)

for w in contents:

if len(w)>1:

pre\_count = s.get(w,0)

s[w] = pre\_count+1

words = sorted(s.iteritems(),key=lambda d:d[1], reverse = True)

words = words[1:100]

wordclouds = WordCloud(font\_path='/usr/share/fonts/wqy-zenhei/wqy-zenhei.ttc').fit\_words(words)

import matplotlib.pyplot as plt

plt.imshow(wordclouds)

plt.axis('off')

plt.show()

绘制出来的词云如下所示：



由分析出来的词云，我们可知绩效奖金是最基本的福利，其次是五险一金，岗位晋升还有带薪节假日，总体来说福利还不错。

最后通过上面的一系列分析，我们可以总结如下几点：

1. 数据挖掘职业主要存在于北上广深这样经济比较发达一线的城市，二三线城市还有待发展。而且招聘数据挖掘职位的公司大都是成熟型和上市公司，只有具备一定经济条件和规模才具有能力拥有数据挖掘岗位。
2. 数据挖掘职业主要招聘本科生，经验在一到五年之间，而博士和研究生不吃香的原因是数据挖掘受行业知识经验的影响，并且它还处于较为简单的商业应用方面。
3. 数据挖掘主要包含在移动互联网，电子商务，信息服务和金融行业，这正是互联网带来的大趋势。而其所需要的技能则包括sql、hadoop、python、java、R、Excel等，所以从事数据挖掘岗位可以从这些技能开始。
4. 数据挖掘的薪资、福利以及发展空间相对其他行业都有明显优势，但是相对而言它需要招聘较高水平的综合型人才。

**附录：**

爬取每个岗位技能信息代码：

#coding='utf-8'

import urllib2

from bs4 import BeautifulSoup

import re

# CompanyId.txt是我们处理的每一个岗位信息的页面Id

fs = open('/home/wjw/pythonProject/lagouProject/CompanyId.txt','r')

contents = fs.read()

fs.close()

ls = contents.splitlines()

addr = ['http://www.lagou.com/jobs/%s.html' %l for l in ls]

# 得到相应链接下的页面信息

def get\_page(url):

page\_hearders = {

'Accept':'text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8',

'Host':'www.lagou.com',

'User-Agent':'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0',

'Connection':'keep-alive'

}

request = urllib2.Request(url,headers=page\_hearders)

file = urllib2.urlopen(request)

page\_contents = file.read().decode('utf-8')

file.close()

return page\_contents

# 提取页面信息里面的岗位信息

def get\_realinfo(pageContent):

if pageContent:

soup = BeautifulSoup(pageContent, 'lxml')

job\_description = soup.select('dd[class="job\_bt"]')

job\_description = str(job\_description[0])

rule = re.compile(r'<[^>]+>')

result = rule.sub('', job\_description)

return result

else:

return None

# 找到所有的技能信息

def search\_skill(realinfo):

if realinfo:

rule = re.compile(r'[a-zA-z]+')

skills = rule.findall(realinfo)

return skills

else:

return None

# 将统计后的技能信息保存在Excel

def save\_in\_txt(all\_info,file\_name):

fs = open(r'/home/wjw/%s.txt' %file\_name,'w')

for info in all\_info:

fs.write(str(info)+'\n')

fs.close()

for a in xrange(0,len(addr)):

print addr[a]

try:

pageContent = get\_page(addr[a])

except Exception as e:

pageContent = None

print 'fail %s' %addr[a]

continue

realinfo = get\_realinfo(pageContent)

skills = search\_skill(realinfo)

for s in skills:

s = s.lower()

pre\_count = skill\_dict.get(s,0)

skill\_dict[s] = pre\_count+1

skill\_dict = sorted(skill\_dict.iteritems(),key=lambda d:d[1],reverse=True)

print skill\_dict

save\_in\_txt(skill\_dict,'skills')

print 'success'