今天来聊聊求职需要的 Python 技能

原创 某某白米饭 Python技术 1周前

每年的 3、4 月份都是金三银四跳槽季，企业一般也会选择在这个时期调整职工的薪资，小伙伴在这个时候也会心里痒痒，在招聘网站上看看是否有合适的机会，需要的 Python 技能是否符合年限等等情况。这里以招聘网站为例抓取魔都近一个月的招聘数据，生成柱状图与词云。

抓取招聘网站数据

首先将魔都 近 1 个月的招聘职位都抓取出来，使用 requests 模块和 BeautifulSoup 模块

```
# -*- coding: utf-8 -*-
import requests
from bs4 import BeautifulSoup
import time
import random

urlFileName = 'urls.txt' # 存放招聘信息详情的URL文本
contentFileName = 'context.txt' # 存放抓取的内容
def getUrls2Txt(page_num):
    p = page_num+1
    for i in range(1, p):
        urls = []
        # 抓取魔都的
        url = 'https://search.51job.com/list/020000,000000,0000,00,2,99,Python,2,'+str(i)+'.html?lan

        html = requests.get(url)
        soup = BeautifulSoup(html.content, "html.parser")
        ps = soup.find_all('p', class_='t1')
        for p in ps:
            a = p.find('a')
            urls.append(str(a['href']))
        with open(urlFileName, 'a', encoding='utf-8') as f:
            for url in urls:
                f.write(url+'\n')
        s = random.randint(5, 30)
        print(str(i)+'page done,'+str(s)+'s later')
        time.sleep(s)

def getContent(url, headers):
    record = ''
    try:
        html = requests.get(url, headers=headers)
```

```
        soup = BeautifulSoup(html.content, 'html.parser')
        positionTitle = str(soup.find('h1')['title']) # 标题
        salary = soup.find_all('strong')[1].get_text() # 薪资
        companyName = soup.find('p', class_='cname').get_text().strip().replace('\n','').replace('查
        positionInfo = soup.find(
            'div', class_='bmsg job_msg inbox').get_text().strip().replace('\n', '').replace('分享',
        record = positionTitle + '&&&' + salary + '&&&' + companyName + '&&&' + '&&&' + positionInfo
    except Exception as e:
        print('错误了')
    return record


def main():
        page_num = 93
        getUrls2Txt(page_num)
        user_Agent = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_3) AppleWebKit/537.36 (KHTML, lik
        headers = {'User-Agent': user_Agent}
        with open(urlFileName, 'r', encoding='utf-8') as f:
            urls = f.readlines()
        i = 0
        for url in urls:
            url = url.strip()
            if url != '':
                record = getContent(url, headers)
                with open(contentFileName, 'a', encoding='utf-8') as f:
                        f.write(record + '\n')
                i += 1
                print(str(i)+'详情抓取完成')
                time.sleep(1)

        print('完成了')


if __name__ == '__main__':
    main()
```

抓取网站内容结果图

分词

在这一步需要对招聘信息中的职位信息进行人工的初步删选，过滤掉常用字存入 filterWords 变量中，然后利用结巴分词(https://github.com/fxsjy/jieba)基于TF-IDF算法将职位信息进行分词，并统计技术词语出现的次数。

```
from jieba import analyse

fenCi = {}

```

```
def main():

    # 负责过滤的词语，这里只列出了几个
    filterWords = ['熟悉', '熟练', '经验', '优先', '应用开发', '相关', '工作', '开发', '能力', '负责', '技

    # 结巴分词基于 TF-IDF 算法的关键词
    tfidf = analyse.extract_tags

    for zpInfo in open('context.txt', 'r', encoding='utf-8'):

        if zpInfo.strip() == '':
            continue
        # 详情数据是用&&&分割的
        infos = zpInfo.split("&&&")
        words = tfidf(infos[-1])

        words = [x.upper() for x in words if x.upper() not in filterWords]

        for word in words:
            num = fenCi.get(word, 0) + 1
            fenCi[word] = num

    print(sorted(fenCi.items(), key=lambda kv: (kv[1], kv[0]), reverse=True))
    print('分出了' + str(len(fenCi)) + '了词语')


if __name__ == '__main__':
    main()
```

分词结果图

技能图表

在分词中，分出了 12663 个词这些词大多都是常用字，需要进一次筛选出多个高频的 Python 技能利用 matplotlib 模块画出柱状图。

```
import matplotlib.pyplot as plt
import numpy as np

plt.rcParams['font.sans-serif'] = ['Arial Unicode MS']
params = {
    'axes.labelsize': '14',
    'xtick.labelsize': '14',
    'ytick.labelsize': '13',
    'lines.linewidth': '2',
    'legend.fontsize': '20',
```

```
    'figure.figsize': '26, 24'
}
plt.style.use("ggplot")
plt.rcParams.update(params)

# 筛选分词中高频的
barDir = {
    'PYTHON': 2283,
    'LINUX': 981,
    '算法': 658,
    '运维': 530,
    '数据库(MySql,Sql,Redis等)': 1021,
    'SHELL': 996,
    '数据分析/挖掘': 695,
    'WEB': 454,
    '测试用例': 515,
    'MATLAB': 221,
    'PERL': 209,
    'HIVE': 122,
    'HADOOP': 176,
    'SPARK': 146,
    'TENSORFLOW': 136,
    '多线程': 127,
    'AI': 106,
    'SAS': 104,
    '视觉/图像处理': 180,
    '人工智能': 170,
    'HTTP': 90,
    'DOCKER': 82,
    'DJANGO': 82,
}

fig, ax = plt.subplots(figsize=(20, 10), dpi=100)

# 添加刻度标签
labels = np.array(list(barDir.keys()))
ax.barh(range(len(barDir.values())), barDir.values(), tick_label=labels, alpha=1)

ax.set_xlabel('Python技术词的次数', color='k')
ax.set_title('Python工作高频技术词')


# 为每个条形图添加数值标签
for x, y in enumerate(barDir.values()):
    ax.text(y + 0.5, x, y, va='center', fontsize=14)

# 显示图形
plt.show()

```

图表

词云

最后将分词数据生成一个词云，将 Python 图标作为底图使用。

```
def getWorldCloud():
    # 底层图片路径
    path_img = "python.jpg"
    background_image = np.array(Image.open(path_img))

    wordcloud = WordCloud(
        # 字体路径
        font_path="/System/Library/Fonts/STHeiti Light.ttc",
        background_color="white",
        mask=background_image).generate(" ".join(list(fenCi.keys())))
    image_colors = ImageColorGenerator(background_image)
    plt.imshow(wordcloud.recolor(color_func=image_colors), interpolation="bilinear")
    plt.axis("off")
    plt.show()
```

最后生成的词云图

总结

本文主要是从招聘网站抓取 Python 工作职责并生成柱状图和词云，展示企业需要哪些 Python 技能，从而在面试前学会并运用这些技能。在生成最后结果的过程中存在 2 点不完美的情况，一点是存在人工筛选另一个是在分词中没有完全过滤掉通用字。随着小编的 Python 技能树的增长，有理由相信在不久这 2 种情况将完全避免。

> 示例代码：求职需要的 Python 技能