**基于空间向量模型的简易搜索引擎实现**

目录

[1 引言 3](#_Toc533508988)

[1.1实验目的 3](#_Toc533508989)

[1.2开发环境 3](#_Toc533508990)

[2应用技术 3](#_Toc533508991)

[3实验内容 4](#_Toc533508992)

[3实验结果分析 7](#_Toc533508993)

[4实验中遇到的问题 13](#_Toc533508994)

[5实验总结 14](#_Toc533508995)

[6参考文献 14](#_Toc533508996)

# 1 引言

## 1.1实验目的

近几年在线社交网络分析是计算机领域炙手可热的综合课题，涉及到自然语言处理、机器学习、统计学等多方面技术，可应用于社交推荐、舆情分析、隐私保护、谣言检测、可视化等多个领域。本实验以信息检索课程内容为理论基础，以自然语言处理技术为重点，设计了一个基于空间向量模型的简易搜索引擎。希望我通过该实验能进一步理解搜索引擎的工作原理以及提高对复杂爬虫的处理能力和掌握计算文本相似度的三种方法。

## 1.2开发环境

硬件环境：PC机

开发环境：PyCharm

Anaconda(Python3.6)

工具包：jieba-0.39

BeautifulSoup

flask\_script

sklearn

urllib3

# 2应用技术

1. 爬虫技术

对采取JavaScript渲染的网页利用api进行爬取.

1. 分词处理基础

采用Jieba中文分词工具对爬取的数据进行分词。

1. 计算相似度

内积法，Jaccard法，余弦法。

1. 文本向量化技术

TF-IDF模型。

# 3实验内容

1.爬取网页

（1）本实验爬取了网易新闻作为搜索引擎的“数据库”，作为一个友好爬虫，在爬取数据之前先查看网易新闻的Robots协议，如图1所示。可以看出网易新闻是禁止今日头条新闻的爬虫爬取数据的。

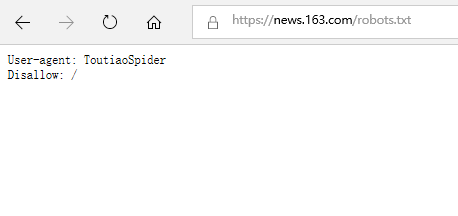


图 1

（2）网易新闻的标题及内容是经过JavaScript渲染的，简单的下载网页源代码是没有任何文字信息的，如图2所示。因此我在浏览器中通过网易新闻首页找到控制台的 Network选项找到api接口。<http://news.163.com/special/0001220O/news_json.js>,从中获取数据，如图3所示。



图 2



图 3

（2）采用requests库进行数据爬取，再对得到json格式数据进行处理分类，最后爬取到142个新闻类型文本文档，结果如图4所示。

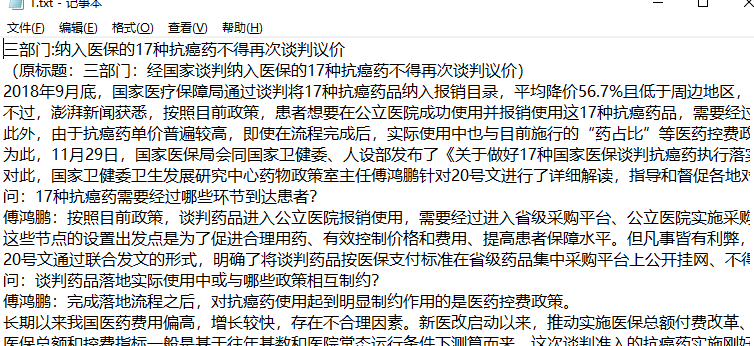


图 4

核心代码如下：

|  |
| --- |
| dir = 'D:/quick/data/' url = 'http://news.163.com/special/0001220O/news\_json.js' request = requests.get(url) request.encoding = request.apparent\_encoding data = request.text[9:-1] internal = json.loads(data)['news'][0] external = json.loads(data)['news'][1] society = json.loads(data)['news'][2] news\_list = {} for i in internal:  news\_list[i['t']] = i['l'] for e in external:  news\_list[e['t']] = e['l'] for s in society:  news\_list[s['t']] = s['l'] index = 1 for key in news\_list:  url = news\_list[key]  request = urllib.request.urlopen(url)  html = BeautifulSoup(request.read().decode('gb2312',"ignore"), 'html5lib')  content = html.find('div', {'class': 'post\_body'})  if isinstance(content, bs4.element.Tag):  content = content.find\_all('p')  with open(dir + str(index) + '.txt', 'a', encoding='utf-8') as news\_file:  news\_file.write(key + '\r\n') |

2.文本分词处理

采用jieba工具对语料库进行了预处理，其中使用了搜索引擎模式jieba.cut\_for\_search（）。这可以使文本在精确模式切分的基础上，对长词再次切分，提高召回率。 搜索引擎构建倒排索引的分词时经常会使用该方法。分词结果如图5所示。

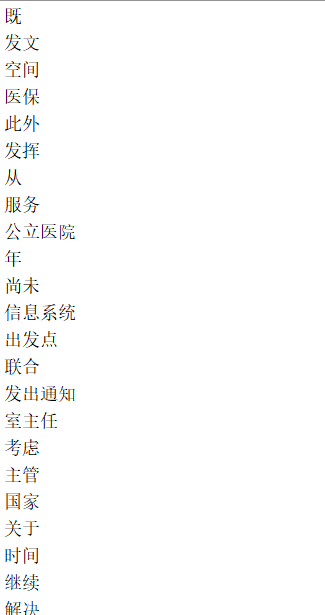


图 5

3.搭建Flask框架进行搜索引擎开发

当浏览器把请求发给 Web 服务器时，Web 服务器把请求发给 Flask 程序，Flask 程序保存了一个 URL 到 Python 函数的映射关系，即视图函数和路由。使用Flask框架可以让我们的代码更容易维护，里面的Jinja2组件让我们的数据可视化。

# 3实验结果分析

1.根据关键词搜索

当用户在浏览器端输入关键词后，后台对关键词进行分词和停用词处理，如图6所示“中央”二字被切分为“中”“央”。



图6

2.查询结果

（1）首先计算语料库中每个文章每个分词的tf值，然后使用公式进行归一化处理。如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类，其计算公式为C:\Users\fenghui\Desktop\tf.png，nij表示该词在文章中的出现次数，nkj表示文章中所有词出现的次数。

（2）接着计算语料库中有多少文章出现了搜索词， 并直接计算搜索词的tfidf值。

（3）然后将每个文档按照权值排序后构建向量，利用向量內积计算文章的相似度。

（4）将文章按照相似度从大到小排序，讲排序结果返回给浏览器。

“中央”的查询结果如图7所示。



图 7

核心代码如下：

|  |
| --- |
| # 计算query每个词的tfidf seg\_list = jieba.cut\_for\_search(query, HMM=False) words\_in\_query = list(seg\_list) # 对query中每个词，计算在query中出现的次数 tfidf\_in\_query = {} for word in words\_in\_query:  word = word.strip()  if len(word) > 0:  tfidf\_in\_query[word] = tfidf\_in\_query.get(word, 0.0) + 1.0 a\_pow = 0.0 # 搜索词向量模长的平方 # 计算语料库中有多少文章出现了搜索词中的这个词， 最后直接计算这个词的tfidf for key in tfidf\_in\_query:  news\_count = 1.0  for i in range(142):  if words\_count[i].get(key, -1) != -1:  news\_count += 1.0  tfidf\_in\_query[key] = tfidf\_in\_query[key] / len(words\_in\_query) \  \* math.log10(142.0 / news\_count)  a\_pow += tfidf\_in\_query[key] \*\* 2 for i in range(142):  ab = 0.0 # 向量内积  b\_pow = 0.0 # 文章向量模长的平方  # 计算文章每个词的tfidf  for key in words\_count[i]:  sum = 1.0  for j in range(142):  if words\_count[j].get(key, -1) != -1:  sum += 1.0  words\_count[i][key] \*= math.log10(142 / sum)  ab += tfidf\_in\_query.get(key, 0.0) \* words\_count[i][key]  b\_pow += words\_count[i][key] \*\* 2  tfidf = ab / (a\_pow \* b\_pow) \*\* 0.5  results[i].append(tfidf) # 将文章按照相似度从大到小排序 results = sorted(results, key=lambda news: news[-1], reverse=True) |

3.查看新闻内容

查询结果为与“中央”一词相似度比较高的新闻标题，点开标题链接可查看文章内容，内容来源为网易新闻中爬取的文本文档。如图8所示。



图 8

4.计算新闻中每个分词的TFIDF值

该搜索引擎是基于空间向量模型（VSM）实现的，TF-IDF值计算是搜索引擎实现的重要步骤之一。以2018年12月17日新闻“重磅央企再有前高管被查 此前还有一位高管落马”为例，各个分词的TF-IDF值计算结果如图9所示。

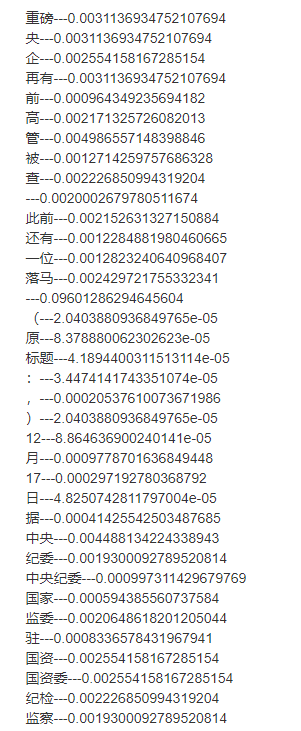


图 9

5.计算两个句子的相似度

计算文档相似度有多种方法，这里采用了內积法、余弦法、Jaccar法三种求相似度的算法进行了实验，实验结果如图10、图11所示。这里我选用了“辽宁纪委原副书记杨锡怀严重违纪违法”和“中国中化集团公司总经理蔡希涉嫌严重违纪”两个新闻的内容进行了相似度计算。內积值无上下限，相对而言数值越大相似度越高。余弦值介于-1和1之间，越接近1相似度越高，适合数据稀疏的情况。Jaccard系数值越大，样本相似度越高。实验结果对比说明，两篇新闻相似度较高，都属于反腐倡廉的新闻内容。



图 10



图 11

# 4实验中遇到的问题

(1)实验过程中乱码问题：爬取数据中文显示乱码，文档内容正常时回传给网页显示乱码。

解决方法：网易新闻编码方式为GB2312，爬虫返回数据设置为GB2312，当使用BeautifulSoup对html网页进行解析时，将获取内容解码成GB2312格式，写入文档时设置为utf-8格式。具体实现如下：

|  |
| --- |
| request.encoding = request.apparent\_encoding  html = BeautifulSoup(request.read().decode('gb2312',"ignore"), 'html5lib')  open(dir + str(index) + '.txt', 'a', encoding='utf-8') |

(2) Flask工程中无法获取表单数据，路由和视图函数与对应的URL无法匹配成功，蓝本注册出现了循环导入依赖的错误。

解决办法：使用普通方法获取表单时若提交数据不合理后台无法接收到，我改用[Flask-WTF](http://packages.python.org/Flask-WTF) 方法提示用户输入不正确。将views.py的导入语句放在程序最后可避免循环导入依赖。具体实现如下：

|  |
| --- |
| //使用flask\_wtf进行表单创建  from flask\_wtf import FlaskForm from wtforms import StringField, SubmitField from wtforms.validators import Required class SjetForm(FlaskForm):  query = StringField(u'请输入你想查询的新闻关键词', validators=[Required()])  submit = SubmitField(u'Google一下')  #避免循环导入依赖  from flask import Blueprint sjet = Blueprint('sjet', \_\_name\_\_) from . import views, errors |

# 5实验总结

纸上得来终觉浅，绝知此事要躬行。通过实验我熟悉了搜索引擎的工作流程，爬取网页、建立索引、搜索词处理、搜索结果排序四部分。通过查阅资料和实践我明白了计算文本相似度各个方法的特点。同时，我对VSM模型的原理有了更深的理解，能够熟练计算文档的TF-IDF值，利用词频把文本转化成空间向量进一步求相似度。

# 6参考文献

[1]高远. 动态网页爬取的两种策略.2016-12-04.   
[https://blog.csdn.net/wgyscsf/article/details/53454910.](http://www.bjyouth.com.cn/Bqb/20000412/GB/4216%5ED0412B1401.htm.)

[2] [judewen\_hi](https://me.csdn.net/u010598982). vsm向量空间模型实现.2016-03-26.

<https://blog.csdn.net/u010598982/article/details/50985994>.

[3]Yoangh. 推荐算法基础--相似度计算方法汇总.2017-09-4. <https://blog.csdn.net/y990041769/article/details/77837915>.

[4]lyy0905. python文本相似度计算.2017-04-28.

<https://www.jianshu.com/p/edf666d3995f>.