ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.16, No.8, March. 2020
E-mail：eduf@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

基于Python的商品评论文本情感分析

曾小芹,余宏

(豫章师范学院 数学与计算机学院,江西 南昌 330103)

摘要:文本情感分析是自然语言处理的重要过程。研究首先运用Selenium爬虫抓取评论文本,通过Jieba分词工具对文本进行分词、词性标注及关键词词云的生成,再选用适用于中文文本处理的snowNLP库对评论文本进行情感计算和结果可视化,并通过准确率和召回率验证了研究结果,将对结果进行了详细分析。最后,给出了进一步研究方向。

关键词:中文文本情感分析;SnowNLP;Python;NLP

中图分类号:TP3 **文献标识码:**A

文章编号:1009-3044(2020)08-0181-03

开放科学(资源服务)标识码(OSID):

DOI:10.14004/j.cnki.ckt.2020.0941

Sentiment Analysis of Merchandises' Comment Based on Python

ZENG Xiao-qin,YU Hong

(Mathematics and Computer College,Yuzhang Normal University, Nanchang 330103, China)

Abstract: Text sentiment analysis is an important process of natural language processing.Above all, Selenium spider is utilized to crawl comment text, and then applying Jieba segmentation tools to do word segmentation,part-of-speech tagging and using WordCloud to generate the wordcloud of keywords, then making use of snowNLP library which is suitable for Chinese text processing to calculate text' sentiment and results' visualization.The results was verified through the accuracy and recall rate and analyzed in detail.At last,.Finally, the further research direction is presented.

Key words: Chinese text sentiment analysis; SnowNLP; Python; NLP

文本情感研究是计算机语言学、人工智能、机器学习、信息检索、数据挖掘等交叉领域的研究热点,具有高度综合性和实用性。文本情感分析就是对文本的显性主观性或隐性关联性信息进行分析,让机器感受人类的感情从而更深入地理解人类语言,为自然语言处理(NLP)的研究提供有效帮助。而NLP在诸多实际应用如个性化推荐系统、信息安全过滤系统、网络用户兴趣挖掘、网线预警系统等的决策制定中都有产生带来巨大的经济、社会效益。为此,对文本情感识别具有重要的学术研究意义及社会经济价值。

1 主要相关技术简介

1.1 SnowNLP库

Python有多个工具库用于自然语言处理,但绝大多数是针对英文的处理。由于中英文的诸多差异,很多库不能直接拿来使用,要扩展也不简单。SnowNLP库是一个用Python语言编写的专门处理中文文本的类库,与其他类库不同的是它的实现没有依靠NLTK,所有算法均是自主实现,且自带语料库和情感字典。SnowNLP支持多种中文文本处理操作包括:中文分词、词性标注、情感分析、文本分类、转换成拼音、繁体转简体、提取文本关键词、提取文本摘要及计算文本相似度等[1],功能十分全面。

1.2 Selenium爬虫

传统的爬虫通过直接模拟HTTP请求来爬取站点信息,但当网络爬虫被滥用后,互联网上就出现太多同质的东西,原创版权得不到保护,且传统的方式和浏览器访问差异比较明显,当前很多网站网页都有反爬虫机制,要想获取网页内容就没有那么轻松了。Selenium是自动化浏览器技术,通过驱动浏览器来模拟真实浏览器完成网络行为,最终拿到网页渲染后的信息。整个过程就如真正的用户在操作,为此,它用于爬虫躲避反爬机制再合适不过。Selenium爬虫支持多种语言及多种浏览器,不用去分析每个请求的具体参数,比起传统的爬虫开发起来更容易,它唯一的不足是速度慢,如果对爬虫的速度没有要求,那使用Selenium是个非常不错的选择。当然,在实际的操作过程中,还会遇到很多意外,比如,虽然Selenium完全模拟了人工操作,给反爬增加了困难,但如果网站对请求频率做限制的话,Selenium同样会被封杀,所以通常还得给浏览器设置代理技术等。

2 SnowNLP库情感分析

文本情感分析又称为文本倾向性分析和意见挖掘,是对带

收稿日期:2019-12-11
基金项目:2017年度江西省教育厅科学技术研究项目青年项目:基于Python NLP库的中文文本情感识别研究(编号:GJJ171193)
作者简介:曾小芹(1986—),女,江西人,讲师,硕士,主要研究方向为大数据及其处理等。

有情感色彩的主观性文本进行分析、处理、归纳和推理预测的过程，其中情感分析还可以细分为情感极性(倾向)、情感程度及主客观分析等。一般来说，情感可以从多层次、多角度进行分类，中国传统文化中就有“七情”——好、恶、乐、怒、哀、惧和欲。在实际的语言应用场景中，不能对一个文本进行一分为两的划分，且情感倾向存在极性和强度，不能单纯地将情感词归为某一类等问题。为此，为了精准计算文本情感值，文本情感分析行为应该建立在当前基准情感词典基础上，关注文本词语在不同情感类别中的强度值，计算出不同语境下词语情感，进而得到文本情感值。当然，本文为简化处理，暂时将文本情感划为两类，关于情感极性与强度的衡量问题将作为下一研究阶段的重点。

利用SnowNLP库的文本情感分析的基本流程如下：

(1) 自定义爬虫抓取信息保存至文件。根据研究需要可抓取不同字段信息保存到不同类型的文件。Selenium爬虫首先要创建浏览器句柄即加载浏览器驱动，再定位页面元素来获取信息文本。

(2) 中文分词、词性标注及可视化显示关键词。中文分词工具也有很多，snowNLP本身能完成，但其准确率不如Jieba分词，为此，本次研究采用Jieba完成分词等工作。为了直观表现、对比文本关键词的重要性，采用WordCloud生成词云的形式来完成结果的可视化步骤。

(3) 情感计算，结果可视化。snowNLP中情感计算实现过程很简单，只要文本调用sentiments方法即可，具体使用见实例。但其实在sentiments方法中调用了sentiment下的分类方法，Sentiment对象首先调用load方法加载训练好的数据字典，然后再调用classify方法，在classify方法中实际调用的是Bayes对象中的classify方法，具体可查看SnowNLP库中的sentiment文件夹下的__init__文件等。再运用Matplotlib生成情感分数分布柱状图和情感波动曲线图来实现结果可视化。

(4) 结果评估。结果可以使用准确率、召回率来衡量。召回率用来度量完整性或灵敏度。较高的召回意味着更少的假负，而较低的召回意味着更多的假负。提高召回率往往就会降低精确度，两者往往难以同时达到最理想的状态。

3 SnowNLP库情感分析实例

本研究目的是用SnowNLP库来分析下豆瓣上书籍的短语文本情感，详细过程如下：

(1) 获取评论文本。其实无论使用什么方式抓取文本，都要仔细分析所需信息存在于页面的哪些标签以及在标签的哪些属性里。本研究使用Selenium来爬取所需实验文本，主要抓取的是豆瓣上的关于《Python编程快速上手》这本书的所有短评评论，将所有评论存储在txt文本中，同时也将其他与评论相关的字段如评论用户名、用户链接、评论时间、推荐星数、评论等信息一同存储在CSV文件，以备后续研究所用。部分代码[2]如下所示：

```
c = open("book-douban.csv", "w", encoding='utf-8') #写文件
writer = csv.writer(c,dialect=("excel"))#写入对象
writer.writerow(["序号","用户名","评分","评分标题","有用数","日期","评论"])
driver = webdriver.Firefox()
i = 1
while i<20:
url = "https://book.douban.com/subject/26836700/comments/hot?p=" + str(i)
driver.get(url)
elem1 = driver.find_elements_by_xpath("//div[@class='avatar']/a")
elem2 = driver.find_elements_by_xpath("//span[@class='comment-info']/span[1]")
elem3 = driver.find_elements_by_xpath("//span[@class='comment-vote']/span[1]")
elem4 = driver.find_elements_by_xpath("//span[@class='comment-info']/span[2]")
elem5 = driver.find_elements_by_xpath("//span[@class='short']")
tlist = []
k = 0
while k<20:
num = i*20+k+1 #序号
name = elem1[k].get_attribute("title") #用户姓名
score = elem2[k].get_attribute("class") #用户评分及内容
content = elem2[k].get_attribute("title")
useful = elem3[k].text#有用数
date = elem4[k].text#日期
shortcon = elem5[k].text #评论
templist = []
templist.append(num).append(name).append(score)
templist.append(content).append(useful).append(date).append(shortcon)
writer.writerow(templist)
k = k + 1
i = i + 1
c.close()
```

当然，在爬取文本信息过程中遇到的相关问题有：(1)网页上的用户名及评论中出了特殊字符即中文文字以外的一些符号，简单解决方案：对出现特殊字符字段，使用代码shortcon=shortcon.translate(non_bmp_map)来去掉特殊字符；(2)爬取过程中虽然暂未出现被封杀情况，但豆瓣页面内容本身有折叠隐藏的部分信息，即使代码正确也无法完全抓取，此时需要加入异常处理代码，折叠部分文本需要手动收集；(3)评论文本并不每页都是同样的数量，为此，不能固定每页爬取的数量，应该灵活计算。

(2) 中文分词。本研究是采用结巴分词方法对上述评论文本进行分词，具体代码在作者的其他论文中可查看。再运用WordCloud生成文本分词后的词云，核心代码及词云图如下所示：

```
with open(path+'cut_word.txt', encoding='utf-8') as f:
comment_text = f.read()
color_mask = imageio.imread(path+"book.jpg") # 读取背景图片
Stopwords =stopwordslist("stopword.txt")#添加自定义停用词表
cloud = WordCloud(font_path= "FZYTK.TTF", background_color='white',
```

```
max_words=200, max_font_size=200, min_font_size=4, mask=
color_mask,sto pwords=Stopwords)
word_cloud = cloud.generate(comment_text) # 产生词云
image_colors = ImageColorGenerator(color_mask)
plt.imshow(cloud.recolor(color_func=image_colors))
plt.figure()
plt.imshow(color_mask, cmap=plt.cm.gray)
plt.axis("off")
plt.show()
```

图1　评论文本关键词生成的词云

在词云中，显示越大的词越能体现评论文本的核心思想，可以看到关于本书的主要讲解的是python的基础知识，适于入门使用，并且也注重编程的训练。

(3) 情感分析。基本过程：按行读取评论文本，利用循环每行文本调用sentiments方法计算得到每行(条)评论的情感分数(0-1的一个值)，将其他存入文件。核心代码[3]如下：

```
source = open("comment.txt","r",encoding="UTF-8")
line = source.readlines()
score1=[]#情感分数数组
i=0
for i in line:
s=SnowNLP(i)
r=s.sentiments
score1.append(r)
mpl.rcParams['font.sans-serif'] = ['SimHei']#显示中文
matplotlib.rcParams['axes.unicode_minus'] =False#显示负号
plt.hist(score1,np.arange(0, 1, 0.01), facecolor = 'b')
plt.xlabel("情感分数")
plt.ylabel("用户评论")
plt.title("用户评论情感分析柱状图")
plt.show()
source.close()
```

在情感分析的时候，为了更清晰的判断情感的极性分类，通常可以将情感区间从[0, 1.0]转换为[-0.5, 0.5]，位于0以上的是积极评论，反之消极评论。从上图可知，柱状图直观展示所有评论的分数分布情况，曲线图说明评论的情感波动情况。评论情感波动性较小，大部分评论情感比较集中、接近。

上述代码会从文件中逐行读取文本进行情感分析并输出最终的结果。SnowNLP库中训练好的模型是基于商品的评论数据，此次研究就不需要重新训练模型。但它也可用于其他类别文本的情感分析，只不过在实际使用的过程中，需要根据自己的情况，重新训练模型。

(4) 结果验证。首先人工标记每一条评论的情感极性——

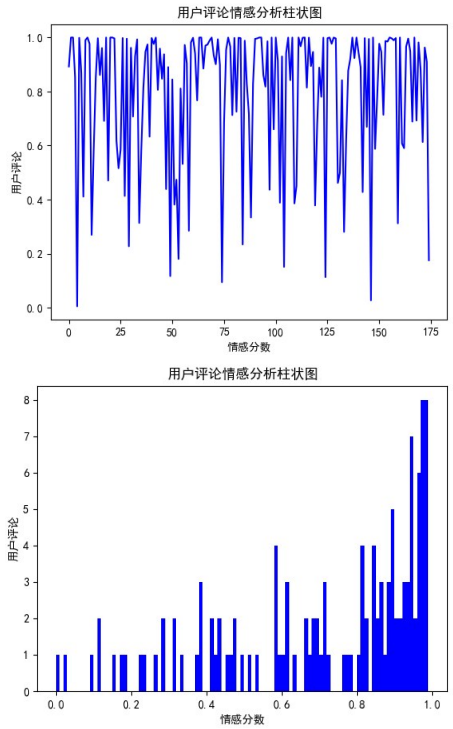

图2　文本情感分数分布柱状图及情感波动曲线图

1表示积极评论，-1表示消极评论。本次评论文本情感分析结果：积极评论准确率及召回率分别为0.784和0.978，消极评论准确率及召回率分别为0.935和0.507。从结果可看出：positive类文本被正确识别的有97.8%的召回率，说明在positive类文本中极少存在识别错误的情况，但positive类却只有78%的准确率，准确率有待提升。negative类准确率达到93%，这意味着负类情感识别十分精准，但召回率却只有50%，说明存在很多负类情感文本被错误地分类。对于负类文本来说，由于中文的博大精深，通常在使用语境中用负面词汇来表达正面评价，如“不是很好”。当然，中文也存在很多中性词或极性强度根据语境决定的词汇，但分类器却只能将词汇情感一分为二。

4 结论

本文完成了使用snowNLP库进行文本情感的分析，总结如下：第一、语料库十分关键，如果构建了精准的类别语料库来替换默认语料库，准确率可能更高；第二、情感分析通常需要与评论的具体时间段结合起来，这样就能在特定的场合发挥情感分析更多的应用，比如舆情预测预警等[3]；第三、情感分析经常不应该一分为二地划分两个不相干的集合，因为词语在不同语境会产生不同的情感极性或强度，这样会降低情感分析的准确性。相关问题将作为后续研究进行。

参考文献：

[1] zhiyong_will. 情感分析—深入snownlp原理和实践[EB/OL]. https://blog. csdn. net/google19890102/article/details/80091502, 2018,6.

[2] Eastmount. 基于SnowNLP的豆瓣评论情感分析[EB/OL].https://blog.csdn.net/Eastmount/article/details/85118818,2018,12.

[3] 王树义,廖桦涛,吴查科. 基于情感分类的竞争企业新闻文本主题挖掘[J]. 数据分析与知识发现,2018,2(3):70-78.

【通联编辑：代影】