**重庆理工大学毕业设计（论文）**

文 献 综 述

学 院 计算机学院

班 级 116030704

学生姓名 刘世闯 学 号 11603990830

**摘要**  以微博为代表的新兴社交媒体，为人们进行意见的表达和信息的交流提供了媒介，具有开放性、实时性、互动性和低门槛性，以极快的速度影响着社会的信息传播格局。对微博文本进行情感分析有助于了解公众的情感状态，及时获取公众的观点、态度，对于舆情控制、产品营销、民意调查等具有重要的意义。

情感分析，也称倾向性分析、意见抽取和意见挖掘。主要是通过对带有情感色彩的主观性文本进行分析、处理、归纳然后进行推理的过程。而微博，人口基数大，涉及的话题广泛，对人们的日常生活产生了不可估量的影响，而对微博的情感分析，更是有着十分重要的意义。为此，本文针对了微博文本的情感分析进行了如下几个工作。这些带有情感信息的微博是非常宝贵的资源，通过情感分析可以获取网民的此时的心情，对某个事件或事物的看法，可以挖掘其潜在的商业价值，还能对社会的稳定做出一定的贡献。

**关键词**：深度学习 情感分析 自然语言处理 数据挖掘 舆情分析

**1、功能需求**

系统的预期功能是通过网页展示根据用户需要进行分析的某一条微博，可以由用户输入或也可以展示出最近的热门微博由用户选择，用户选择完成后由服务器进行处理，然后通过网页展示分析结果，包括正向情感、中性情感和负向情感的分布情况(即所占比例)、热门词汇、热门表情等等，系统的核心是对用户所选择的微博情感分析结果的可视化展示。

1. **性能需求**

因为本系统使用的是网页来展示结果，运行程序主要由服务器来完成，所以不太需要用户使用时的性能要求和运行环境的要求，用户只需要登录相应的网页即可。但开发时应注意以下四个方面的性能需求：

（1）系统具备可扩展性。能够在本系统的基础上进行后续的开发。

（2）系统须具备一定的并发性。为了系统的正常使用和后续调试开发，系统必须能够支持同时多名用户的并发访问。

（3）系统应具有兼容性。网页应兼容不同的浏览器、系统例如安卓、ios、windows等等。

（4）系统须具备一定的安全性。因为使用了服务器，所以需要具备一定的安全性，防止黑客的攻击。同时要防范用户在输入框输入非法字符进行数据库注入操作。

**3、主要实现方法**

实现步骤主要如下：

1. 新浪微博爬虫
2. 将爬取的微博评论保存到数据库
3. python对数据库的读取并保存到列表
4. 对爬取到的数据进行处理，去掉用户名，无用的url等等，对无关性的微博文本进行过滤，微博已经为热点事件的传播平台，同时，也带来了大量无关的广告、重复字词（“分享图片”）等
5. 利用分词工具例如jieba等对于微博评论进行分词
6. 利用经过深度学习的卷积神经网络模型对经过处理后的文本信息进行分析处理，主要包括特征提取、计算特征向量权值、训练分类、利用分类器对数据进行分类、统计分类结果
7. 做一个网页来展示分析结果，也可以通过网页输入要进行分析的微博

针对微博数据的抓取和存储，目前主要有两种方式：

（1）根据微博官方提供的API接口。这些接口基于REST实现的HTTP协议，以JSON结构化的方式作出响应。但是，这种接口需要一定的权限，还有接口的请求频次限制次数，甚至对于接口的限制的速率限制。

（2）基于微博网页的解析。这种方式需要解析网页，如果网页代码有改动，响应的抓取方式也要有改变，对于大量的抓取，需要破解微博的反爬虫机制（使用代理ip，不同账号等）。

微博的反爬虫机制

防止爬虫一般从三个方面入手：分析网页请求的headers，监督用户访问网站的行为，调整网站中的目录和数据加载的方式。前面两种比较常见，大部分网站都是从这两个角度来反爬虫。第三种会应用一些ajax来反爬虫。

1. 通过Headers反爬虫

目前，一般网站都会检测网页请求中Headers的User-Agent，有的甚至还要检查网站的Referer。如果遇到这类反爬虫机制，我们可以直接在代码中添加Headers和Referer以此来绕过检查。对于这些网站，在代码中添加或修改其中的Headers和Refer就能很好的绕过。

1. 基于用户行为的爬虫

用户访问网站的行为也是目前主流网站常用的检测手段，比如：同一IP在短时间内多次访问了同一个页面，还有的事同一个账户短时间内多次进行相同操作。对于这种情况，我们可以使用IP代理来解决。现在网上有收费的和免费的IP代理，我们可以爬去这些IP代理存储起来，然后每请求几次就更换一次IP。

1. 动态页面的反爬虫

还有一部分网站，数据是通过ajax或者js请求生成的。我们可以使用浏览器对访问网站中的请求的进行分析。如果能找到ajax请求，分析其含义后可以使用上面两种方法解决，获取对应的数据。

如果不能获取ajax的请求，可以调用selenium+phantomjs框架，调用其浏览器内核，来模拟人为操作以及触发页面的js脚本。

1. 微博的反爬虫

微博中的反爬虫使用了以上三种机制，验证客户端的Headers，同时对访问量多大的同意IP进行禁止访问，使用Ajax进行数据传输。要想破解此类爬虫，必须使用IP代理，同一账号不同时间访问，添加Headers等。微博中的Headers添加如下：

conn.request('post', '/oauth2/authorize', postdata,  
 {'referer': self.\_author\_url,  
 'content-type': 'application/x-www-form-urlencoded',  
 'user-agent': 'mozilla/5.0 (windows nt 10.0; wow64) applewebkit/537.36 (khtml, like gecko) chrome/50.0.2661.102 safari/537.36',  
 'cookie': 'your cookie'})

本实验中，主要是基于微博开放平台提供API，然后申请一个statuses/public\_timeline接口，获取到最新的公共微博。

数据库主要使用navicat数据库对爬取的数据进行保存

删除URL

垃圾微博中的内容一般较短，而且一般文字后面都带有链接，由此才能将用户导向网页的入口，如下面几条：

【领 10 元 优惠 券】【券 后价 19.9元】【包邮】 粉丝福利购：http://t.cn/R6j6YyX ​"

"申马 帆布鞋 =26.8 领券拍：http://t.cn/R6lLnsV http://t.cn/R6n9kRO ​"

"玩游戏？找优惠？上莫莉幻想！http://t.cn/RxmHa1i ​"

"蓝牙耳机 =49 领券拍：http://t.cn/R6nIheq http://t.cn/R6nI2Fp ​"

"赫拉 气垫bb =58 领券拍：http://t.cn/R6CjMuM http://t.cn/R6QsX8f ​"

由上述内容可以知道，一般情况下的普通URL链接都是较长的字符串，如果保持原链接会占用微博的资源。因此，微博希望能够将原本的“长链接”变成缩短的短链接。微博中使用散列（hash）索引，将原始链接对应成一个较短的、一一对应的字幕、数字串组合。如：

<http://t.cn/R6QsX8f>

其原链接为：

<http://shop.sc.weibo.com/h5/goods/index?iid=110054105972300003317973>

URL中带有有用的信息很少，一般都是广告的导向和用户的定位。我们使用SQL从数据库中查找含有URL的微博数量统计。

表2.1 统计结果

|  |  |  |
| --- | --- | --- |
| 统计项 | | 结果 |
| 含有URL的微博数量 | 1756（总量：4780） | |
| 平均引用次数 | 2.72 | |

由此可见，URL在微博中的引用量是特别大的，在情感分析前，要对微博文本进行适当的清理，从而去除这些无用的URL，降低这些URL对情感分析的影响。

删除用户名

微博文本中的用户名一般用来提醒该用户，但是，大部分微博用户的用户名毫无规律性，如：@real\_\_pcyyyyy、@CloverH静、@baekhyunee7永远像25岁一样年轻等，对分词器来说有较大的影响，比如JIEBA分词会把“@baekhyunee7永远像25岁一样年轻”拆分成['@', 'baekhyunee7', '永远', '像', '25', '岁', '一样', '年轻']，其中“年轻”会对用来构建的词性特征造成影响，所以，对于用户名的出去也是极其重要的。

去除停用词

停用词（STOP WORDS），在词典中的意思为：对文本中表达的意义并不起什么作用的词语。在SEO中，为了节省存储空间和提高检索速度，搜索引擎会在搜索时自动忽略某些字或词，这些字或词便是停用词。

停用词在一定程度上相当于过滤词，但是过滤词的范围比较大，通常包含色情、政治、暴力等敏感信息，停用词则没有这个限制。通常情况下，停用词可分为两类：

（1）使用广泛，过于频繁的一些单词。比如“我”、“你”之类的词几乎在很多文档中都会出现，对于搜索引擎来说，这类词无法保证准确的搜索结果，还会降低效率。

（2）文档中使用频率很高，但是实际意义不大的词。主要包括语气助词、副词、介词、连词等，在文本表达中没有变现出任何意义。为了增加情感分析的准确性，我们需要对微博文本中去除停用词。

分词

中文的分词是将一个汉字序列变成一个个单独的有意义的词汇。文本挖掘首先要以中文分词为前提。目前常用的中文分词软件主要有以下几种：

（1）BosonNLP：玻森的中文语义开放平台，主要提供了一个方便、性能强大的中文自然语言分析的平台。

（2）NLPIR：是中国科学院计算机研究所的一个产品，积累了多年研究工作，暂时是目前世界上最好的中文分词工具。

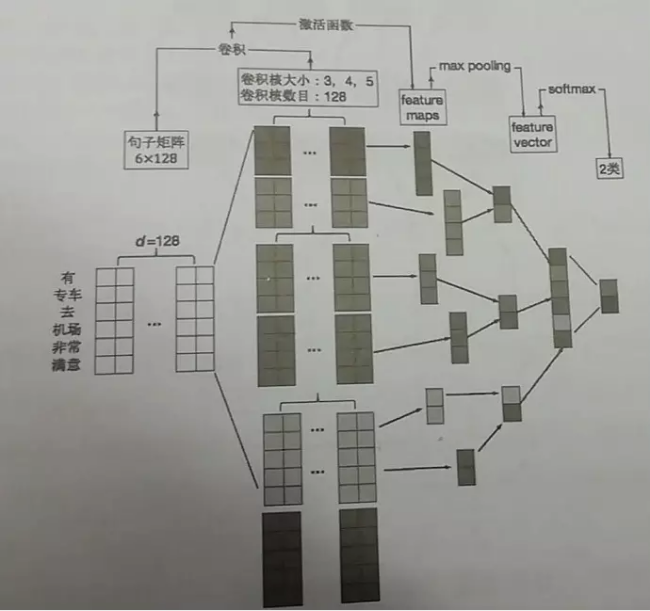
（3）结巴分词：一款开源在GitHub上面的中文分词工具，提供了python、java等多语言的接口，而且能够识别繁体字，立志成为最好的中文分词工具。

选用合适的模型对数据进行分析

**1、TextCNN**

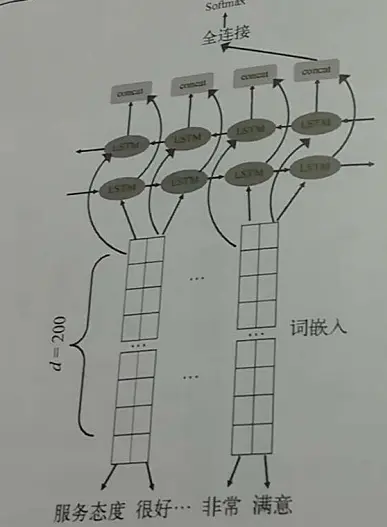
句子 maxlen 的确定（一般选定为语料中最大长度或平均长度为佳）

卷积核大小（2/3/4/5）



**2、TextRNN**

循环神经网络弥补卷积神经网络中卷积核大小固定，导致了卷积神经网络无法抽取到与当前词距离更长的词信息表达。经验表明，循环神经网络更适于自然语言处理问题，能够更好表达文本或语句上下文信息。



def text\_cnn(maxlen=15, max\_features=10000, embed\_size=300):

# input layers

comment\_seq = Input(shape=[maxlen], name='x\_seq')

# Embeddings layers

emb\_comment = Embedding(max\_features, embed\_size)(comment\_seq)

# conv layers

convs = []

filter\_sizes = [2, 3, 4, 5]

for fsz in filter\_sizes:

l\_conv = Conv1D(filters=100, kernel\_size=fsz, activation='relu')(emb\_comment)

l\_pool = MaxPooling1D(maxlen - fsz + 1)(l\_conv)

l\_pool = Flatten()(l\_pool)

convs.append(l\_pool)

merge = concatenate(convs, axis=1)

# out = Dropout(0.1)(merge)

output = Dense(64, activation='relu')(merge)

output = Dense(units=1, activation='sigmoid')(output)

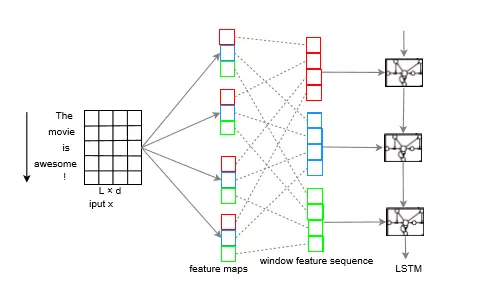
model = Model([comment\_seq], output)

model.compile(loss="binary\_crossentropy", optimizer=optimizers.RMSprop(lr=1e-4), metrics=['acc'])

return model

**3、CRNN**

CRNN 模型是基于 CNN 和 RNN 的拼接式神经网络，其中 CNN 主要用于文本特征抽取，RNN 主要用于后续基于全局序列的情感分类。



def text\_c\_lstm(num\_labels=1,maxlen=15, max\_features=10000, embed\_size=300):

model=Sequential()

model.add(Embedding(max\_features, embed\_size, input\_length=maxlen))

model.add(Conv1D(256, 3, padding='same', strides=1))

model.add(Activation('relu'))

model.add(MaxPooling1D(pool\_size=2))

model.add(GRU(32, dropout=0.2, recurrent\_dropout=0.1, return\_sequences=True))

model.add(GRU(32, dropout=0.2, recurrent\_dropout=0.1))

model.add(Dense(num\_labels,activation='sigmoid'))

model.compile(loss="binary\_crossentropy", optimizer=optimizers.RMSprop(lr=1e-4), metrics=['acc'])

return model

CRNN 结构与编码器-解码器（Encoder-Decoder）结构非常相似，用 CNN 的卷积 + relu + 池化操作作为编码器，其输出作为解码器 LSTM 的输入，最终仍然经过 Softmax 得出最终概率值。

# 参考文献

1. 杜永萍,陈守钦,赵晓铮.   [基于特征扩展与深度学习的短文本情感判定方法](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=JSJA201710051&dbcode=CJFD&dbname=CJFD2017" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 计算机科学 2017(10)
2. 刘小明,张英,郑秋生.   [基于卷积神经网络模型的互联网短文本情感分类](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=JYXH201704015&dbcode=CJFD&dbname=CJFD2017" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 计算机与现代化 2017(04
3. 陈龙,管子玉,何金红,彭进业.   [情感分类研究进展](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=JFYZ201706003&dbcode=CJFD&dbname=CJFD2017" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 计算机研究与发展 2017(06)
4. 吴冬茵,桂林,陈钊,徐睿峰.   [基于深度表示学习和高斯过程迁移学习的情感分析方法](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=MESS201701022&dbcode=CJFD&dbname=CJFD2017" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 中文信息学报 2017(01)
5. 何炎祥,孙松涛,牛菲菲,李飞.   [用于微博情感分析的一种情感语义增强的深度学习模型](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=JSJX201704001&dbcode=CJFD&dbname=CJFD2017" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 计算机学报 2017(04)
6. 李阳辉,谢明,易阳.   [基于深度学习的社交网络平台细粒度情感分析](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=JSYJ201703025&dbcode=CJFD&dbname=CJFD2017" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 计算机应用研究 2017(03)
7. 张庆庆,刘西林.   [基于深度信念网络的文本情感分类研究](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=GDSH201601013&dbcode=CJFD&dbname=CJFD2016" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 西北工业大学学报(社会科学版) 2016(01)
8. 王科,夏睿.   [情感词典自动构建方法综述](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=MOTO201604002&dbcode=CJFD&dbname=CJFD2016" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 自动化学报 2016(04)
9. 陈钊,徐睿峰,桂林,陆勤.   [结合卷积神经网络和词语情感序列特征的中文情感分析](http://gb.oversea.cnki.net/KCMS/detail/detail.aspx?filename=MESS201506024&dbcode=CJFD&dbname=CJFD2015" \t "http://gb.oversea.cnki.net/kcms/detail/frame/_blank)[J]. 中文信息学报 2015(06)
10. Lei Zhang,Shuai Wang,Bing Liu ****Deep Learning for Sentiment Analysis : A Survey****
11. Jordan Prosky,Xingyou Song,Andrew Tan,Michael Zhao ****Sentiment Predictability for Stocks****
12. Pedram Hosseini,Ali Ahmadian Ramaki,Hassan Maleki,Mansoureh Anvari,Seyed Abolghasem Mirroshandel ****SentiPers: A Sentiment Analysis Corpus for Persian****
13. Shuai Wang,Mianwei Zhou,Geli Fei,Yi Chang,Bing Liu ****Contextual and Position-Aware Factorization Machines for Sentiment Classification****
14. W. James Murdoch,Peter J. Liu,Bin Yu ****Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs****