**华中科技大学计算机科学与技术学院**

**机器学习报告**



专 业： 计算机科学与技术

班 级： 计算机1805班

学 号： U201814643

姓 名： 王俪晔

成 绩：

指导教师： 邹复好

**完成日期： 2020年 6 月 27 日**

# 课程项目一

## 一、实验题目：疫情期间网民情绪识别

## 二、实验要求

### 2.1 题目背景

2019新型冠状病毒（COVID-19）感染的肺炎疫情发生对人们生活生产的方方面面产生了重要影响，并引发国内舆论的广泛关注，众多网民参与疫情相关话题的讨论。为了帮助政府掌握真实社会舆论情况，科学高效地做好防控宣传和舆情引导工作，本题目针对疫情相关话题开展网民情绪识别的任务。

### 2.2 数据集

数据集依据与“新冠肺炎”相关的230个主题关键词进行数据采集，抓取了2020年1月1日—2020年2月20日期间微博数据，并对其进行人工标注，标注分为三类，分别为：1（积极），0（中性）和-1（消极）。

训练数据以csv格式存储在train.csv文件中，其中包含45000条微博数据，具体格式如下：

[微博中文内容，情感倾向]

1.微博中文内容，格式为字符串

2.情感倾向，取值为{1,0,-1}

### 2.3 任务描述

根据train.csv文件中的微博数据，设计算法对test.csv文件中的4500条微博内容进行情绪识别，判断微博内容是积极的（1）、消极的（-1）还是中性的（0）。

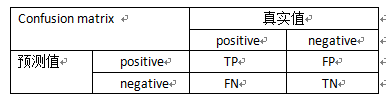
将结果存储在csv文件中，编码采用UTF-8编码，格式如下：

微博中文内容 情感倾向

新冠肺炎…… 1

### 2.4 评测标准

基于以下混淆矩阵(confusion matrix)，采用Precision，Recall，F1-score三个指标评价算法结果，要对比3种以上算法的结果，可进一步自由发挥，做算法参数敏感性的实验及对比分析等。



其中，TP是真阳例，TN是真阴例，FP是假阳例，FN是假阴例。

1. Precision

精确率( 查准率 )，即为在预测为1的样本中，预测正确（实际为1）的人占比，,用混淆矩阵中的字母可表示为：

Precision= TP/(TP+FP)

2. Recall

召回率（查全率），即为在实际为1的样本中，预测为1的样本占比,用混淆矩阵中的字母可表示为：

Recall= TP/(TP+FN)

3. F1-score

F1分数（F1 Score），是统计学中用来衡量二分类模型精确度的一种指标。它同时兼顾了分类模型的准确率和召回率。F1分数可以看作是模型准确率和召回率的一种加权平均，它的最大值是1，最小值是0。

F1=2\*Precision \* Recall /( Precision + Recall)

## 三、算法设计

### 3.1 数据整理

**（1）格式整理**

原始数据集中格式杂乱，为了情感分析更准确，我对原语句进行了下列格式整理：

a. 保留离本级用户最近的一级用户发言内容：微博内容有时包含多级转发，那么我们的模型判断情感时应该是判断哪一级的情感呢？经过对train数据集的观察，绝大多数的情感判断级，是最近的非空级的内容。例如，A大微博，被B转发，C再转发了B转发的A的微博。当C的发言非空时，则只取C的发言内容作为情感判断的目标文本，删除ABC的用户名以及BA的发言内容。这样做也有缺陷，因为有些最近级的发言内容过短，且情感与上几级一脉相承，如果留有上几级的发言内容会判断更准确。具体例如“//@给你史迪仔:北京大学科研所发表微博说患者病毒RNA和蛇的符合//@香蕉你个不林林:蛇尼玛个头，你这么厉害怎么不去当科学家？蛇都冬眠了你出去找条蛇,0”，本级用户没有发言，上级用户发言为中立，上上级发言为消极，最终数据集的所给分类是中立，可见我的分析无误。

b. 删除微博固定格式：删除的微博格式包括两种固定句式：一是“投票类”的“我参与了@罗志渊发起的投票【请老实回答：】，我投给了“没吃过”这个选项，你也快来表态吧~吃蝙蝠干嘛，疯了？”，通过观察所给分类值，确定其判断内容是投票主题和投票选项，故我只保留了投票主题和投票选项；二是“围观类”的“我围观了@意大利华裔郭医生的回答，该问题价值96.00元，围观仅1元，快来一起围观~O微博问答?”由于所给分类值均是0，故我只保留了“我围观了”。删除的微博格式还包括“展开全文”“查看图片”“Lxxx的xx视频”等固定格式词。

c. 统一大小写与简繁体：将微博符号统一为大写字母和简体中文以及其他的符号，不含有小写字母和繁体中文。

d. 合并数值和转发格式：微博中的数字拥有独立含义的很少，故我将所有十进制数表示为“数字”一词。微博中还有很多与转发有关的固定用词，但形式各异，故我将'转发', '快转', 'Repost', 'RepostWeibo'等词统一为“转发”。

**（2）整理分类值**

在对数据和分类结果的观察中，有一些所给标签值与其他大多数标签值的逻辑相悖，同时test集和train集之间相悖的语句较多。为了使数据同分布，我将原训练集9:1划分，作为新的训练集和测试集。同时根据从train集中观察到的普遍规律，提炼出以下原则，依照这些原则对train数据集的部分分类值做了修改，修改清单详见文末。

原则一：报告喜讯且文内有明显情感抒发词的新闻报道属于1

原则二：坚定拥护医护人员并且无其他指责、贬低等消极表达的个人微博属于1

原则三：报道文字内无情感表达的新闻报道或活动介绍词属于0

原则四：实事求是分析事件，无明显贬低或赞扬的个人微博属于0

原则五：“围观格式”微博和无情感表达的广告购物格式微博属于0

原则六：表达对疫情恐慌且没有任何正面情感表达的个人微博属于-1

原则七：凡是含有多种情绪，我无法百分之百下定论的微博，保留原值。

### 3.2 特征提取

在区分不同情感的文本时，富有差异性的词语是关键特征。为了在训练文本的十万有余的词语中提取出富有差异性的词语，降低维度，我在实验中设计了以下两种提取方案。

**（1）方案一：提取标准差率较大者**

第一步，计算三类文本中每个词语的出现频率；

第二步，计算每个词语的三种频率两两之间的标准差率，其公式为：

第三步，提取每类频率标准差率最大的前N个词语，即最终提取了N\*3个关键词语（含重复）作为特征；

第四步，将输入的文本，转换为关键词语出现频率的矢量，由此得到训练/测试的输入量。

该方案所得的特征词分为三组，可以有两种不同用途：一是可以分别提取前N个，再用于分别训练三个二分类器；二是可以分别提取前N个，得到三个集合的并集，作为三分类器的输入特征

**（2）方案二：合并差异性相似的词语**

第一步，计算三类文本中每个词语的出现频率；

第二步，计算每个词语中立频率与消极频率之比P(中立/消极)和积极频率与消极频率之比P(积极/消极)。为了避免消极频率更大时，两种频率接近零而比较性不强，故以P(中立/消极)为例，采用以下分段计算公式：

第三步，一定的粗略化使词语的合并更有效，并且随着两种比值绝对值的增大，其允许的粗略化程度就越大，故以P(中立/消极)为例，采用以下分段粗略化处理：

第四步，将词语按照P(中立/消极)的大小降序排列，按该顺序给词语归类，P(中立/消极)与P(积极/消极)均相等的词语归为一类。由此得到N类词语，即为N个特征；

第五步，将输入文本转换为记录各类词语出现频率的向量，由此得到训练/测试的输入量。

### 3.3 模型训练

训练模型时，我在试验中尝试了以下两种训练方案：

**（1）方案一：三个二分类器**

由于不同的差异性词语，它的差异性可能仅体现在某两类频率之间的差异极高，故该类词语对三类的分类效率可能比不上对二类的分类效率。同时由于筛除了一部分在指定两类间频率差异不大的词语，从而降低了单个训练器的维度，有利于模型收敛。

第一步，用消极频率和中立频率差异最大的前N个词语，将全部的消极训练集、全部中立训练集和随机一个积极训练集矢量化后，合并为第一个二分类器，预测全部测试集，得到预测概率分布。将概率分布中的积极概率清零；

第二步，用中立频率和积极频率差异最大的前N个词语，将全部的中立训练集、全部积极训练集和随机一个消极训练集矢量化后，合并为第一个二分类器，预测全部测试集，得到预测概率分布。将概率分布中的消极概率清零；

第三步，用积极频率和消极频率差异最大的前N个词语，将全部的积极训练集、全部消极训练集和随机一个中立训练集矢量化后，合并为第一个二分类器，预测全部测试集，得到预测概率分布。将概率分布中的中立概率清零；

第四步，将以上三次的概率相加，由最大概率决定最终的预测值。

**（二）方案二：一个三分类器**

直接用N个特征（N个差异词语或N个词语类）将输入文本矢量化，训练出一个三分类器，直接得到最终的预测值。

### 3.3 模型选择

训练模型引用了sklearn中的KNN、随机森林、决策树、三类朴素贝叶斯等六类模型。在评估时，伯努利朴素贝叶斯的训练结果最优，其主要原因可能有两点：一是提取特征时，两种方案的第一步都计算了单个词语在各类文本中的词频，顺应了朴素贝叶斯的频率计算；二是微博属于短文本，单条微博中词语的重复性低，一个词语的出现与不出现即代表了不同的分类可能性，这顺应了伯努利的二极化处理。

### 3.4 数据集选择

本次实验的数据集有两种使用方案：一是以train.csv作为训练集，提取特征，再以test.csv作为测试集；二是随机划分train.csv，九成数据作为训练集，提取特征，剩下一成数据作为测试集。

## 四、实验环境与平台

实验系统环境：windows 10 家庭中文版

实验python环境：anaconda提供的python 3.7

实验编程环境：pyCharm 社区版本2020.1.2

## 五、程序实现

### 5.1 文本数据清洗

# -\*- coding: UTF-8 -\*-

from sklearn.neighbors import KNeighborsClassifier #模型一：47%正确率

from sklearn.ensemble import RandomForestClassifier #模型二：52%正确率

from sklearn import tree #模型三：50%正确率

from sklearn import naive\_bayes #模型四：55%正确率（伯努利）, 48%(高斯), 53%（多项式）

from sklearn.externals import joblib #下载训练好后的模型

from sklearn.metrics import confusion\_matrix #混淆矩阵模型

import csv #按行读写csv文件

import jieba

import jieba.analyse

import pandas as pd

import numpy as np

import math #计算关键词赋分

import matplotlib.pyplot as plt

import os #判断文件是否存在

from langconv import \* #简繁体转换

（1）删除微博用户名

def deleteUserName(text):

index = text.find("@") #目标删除开头

indexLast = -1 #目标删除结尾

while index != -1: #循环寻找、删除@后面的用户名

index = index+indexLast

indexLast = index+1

indexMult = -1

for formatText in [':', '：']: #用户名结束的一般标志是中英文冒号

if (indexMult == -1): #indexMult意味着还没有找到结束标志

indexMult = text[indexLast+1:indexLast + 20].find(formatText)

if indexMult != -1: #找到结束标志后停止寻找其他的结束标志

indexMult += indexLast + 1

break

if (indexMult > indexLast): #如果有结束标志，则替换原来的目标删除结尾

indexLast = indexMult

text = text[:index+1] + text[indexLast+1:] #删除目标开头至目标结尾

indexLast = index

index = text[index+1:].find("@") #寻找下一个开头

return text.replace('/', '') #删除用户名的分隔符/

（2）删除微博投票通知格式

def deleteWeiboFormat(text):

index = text.find("我参与了@") #删除“投票动态”中的格式部分

if (index != -1): #只保留原投票标题和用户所选选项

textBackend=text[index+1:]

text = text[:index] + '' + text[textBackend.find("发起的")+index+1 + 3:textBackend.find("】")+index+1 + 1] + '' + text[textBackend.find(

"我投给了“")+index+1 + 5:textBackend.find("”这个选项")+index+1] + '' + text[textBackend.find("快来表态吧~")+index+1 + 6:]

return text

（3）小写字母转大写字母，繁体中文转简体中文

def textUpper(text):

text = text.upper()

text = Converter('zh-hans').convert(text)

return text

（4）删除##内的客观主题

def deleteTopic(text):

index = text.find("#") # 目标删除开头

indexEnd = 0

while(index != indexEnd+2):

indexEnd = text[index+1:].find("#")+index+1

if(indexEnd==index):

break

text = text[:index+1] + ' ' + text[indexEnd:]

indexEnd = index

index = text[index+3:].find("#") + index+3

return text

### 5.2 文本特征提取与文本矢量化

（1）标准差率方案：仅getMaxStdDev函数与相似词合并方案差异较大，其他地方与相似词合并方案类似。

def getMaxStdDev(countList, minFreq): #计算单个词在三种情况下的标准差率、较大频率值和赋分

retList = []

for i in range(3): #0、1、2依次表示-1消极、0中立、1积极

maxValue = max([countList[i % 3], countList[(i+1) % 3]])

#标准差频率=标准差/平均数

stdValue = np.std([countList[i % 3], countList[(i+1) % 3]]) / np.mean([countList[i % 3], countList[(i+1) % 3]])

score = (math.atan(maxValue)/math.pi + stdValue)/2 #赋分 = arctan(较大频率值)/PI与标准差率的平均值

retList.append([stdValue, maxValue, score])

return retList

（2）相似词合并方案

def textFormalize(path): #将原始训练集中的微博规范化，并划分为lable不同的三组

with open(path, 'r', encoding='UTF-8')as f: #path为原始训练集

csvReader = csv.reader(f)

dataRaw = list(csvReader)

dataNum = len(dataRaw) #dataRaw的长度=train微博条数

for row in range(1, dataNum):

text = dataRaw[row][0] #dataRaw的每一行的第一列是每一条微博的原文

text = deleteWeiboFormat(text) #删除投票转发格式

text = deleteUserName(text) #删除用户名

text = deleteTopic(text) #删除#内的话题

text = textUpper(text)

dataRaw[row][0] = text #记录下被规范后的每条微博

data = pd.DataFrame(data=dataRaw[1:], columns=["text", "lable"])

data = np.array(data.groupby(["lable"])["text"]) #根据'lable'将微博划分成三组

negText = list(data[0, 1])

neuText = list(data[1, 1])

posText = list(data[2, 1])

data = np.array([negText, neuText, posText]) #3\*15000\*2

return data

def getMaxStdDev(countList): #计算单个词在三种情况下的标准差率、较大频率值和赋分

retList = [0, 0]

minFreq = 0.0002 # 设定阈值，过滤频率值极低的词汇

minValue = min(countList)

print(minValue)

if minValue < minFreq:

return retList

for i in range(1, 3): #0、1、2依次表示-1消极、0中立、1积极

if(countList[i] > countList[0]):

retList[i - 1] = round(countList[i] / countList[0], 1)

else:

retList[i - 1] = -round(countList[0] / countList[i], 1)

if (abs(retList[i - 1]) > 4):

retList[i - 1] = (retList[i - 1]\*30//3)/10 # 对大数近似处理

if (abs(retList[i - 1]) > 7):

retList[i - 1] = round(retList[i - 1], 0) # 对大数近似处理

if(abs(retList[i - 1]) > 15):

retList[i - 1] = (retList[i - 1] // 4) \* 4 #对大数近似处理

return retList

def getGroupWordsFrequency(data): #统计得到各组自身词频和与相邻组的相关数据，用于提取重要的关键词

lables = ['countNeg', 'countNeu', 'countPos'] # 用于构造字符串

flag = 0

if (os.access('key.csv', os.F\_OK)):

print("已存在")

return

jieba.load\_userdict('addWords.TXT') #添加自定义词汇

# 明星人名的出现场合高度一致，对常见明星统称为“明星”能减少特征冗余

stars = ['王俊凯', '华晨宇', '肖战', '朱一龙', '玖', '烊', '张艺兴', 'TFBOYS', '韩红', '魏大勋', '李易峰', '王一博', '刘诗诗', '罗云熙', '吴亦凡']

# 以下停用词是成对出现的字符中的另一半，同样为了减少特征冗余

stop = ['全文', '链接', '”', '】', '）', '》', ')']

# 将以下常见微博用语统称为“转发”

repost = ['转发', '快转', 'Repost', 'RepostWeibo']

i = -1

diction = []

print('开始')

for lable in data: #data由textFormalize得到

i = i+1

wordNum = 0 #该组的词汇个数（含重复）

myDiction = dict()

for text in lable: #遍历处理该组的每一句

words = jieba.lcut(text)

for word in words: #遍历处理该句的每一词

wordNum = wordNum + 1

if(word in stop): #为停用词则跳过

continue

if (word.isdecimal()): #数字归为“数字“，减少冗余

myDiction['数字'] = myDiction.get('数字', 0) + 1

elif (word in stars): #明星归为“明星“，减少冗余

myDiction['明星'] = myDiction.get('明星', 0) + 1

elif (word in ['##']): #连在一起的#记作两个#

myDiction['##'] = myDiction.get('##', 0) + 2

elif (word in repost): #转发格式归为“转发“，减少冗余

myDiction['转发'] = myDiction.get('转发', 0) + 1

else: #其他词汇则按自身处理

myDiction[word] = myDiction.get(word, 0) + 1

temp = []

for key in myDiction.keys(): #遍历处理该组的字典

temp.append([key, myDiction[key]]) #转换为list

myDiction = pd.DataFrame(data=temp, columns=['word', lables[i]])#转换为df

myDiction[lables[i]] = myDiction[lables[i]] / wordNum \* 1000 #将频数转换后放大后的频率值

diction.append(myDiction)

myDiction = pd.merge(diction[0], diction[1], how='inner') #将三组的df按交集合并为一个df

myDiction = pd.merge(myDiction, diction[2], how='inner')

rating = []

for index, row in myDiction.iterrows():

retList = getMaxStdDev([row[lables[i]] for i in range(3)])

if (retList[0]==0 or (abs(retList[0]) ==1 and abs(retList[1]) ==1)):

continue

rating.append(retList)

tempDf = pd.DataFrame(data=rating, columns=['neu/neg', 'pos/neg'])

myDiction = myDiction['word']

tempDf = pd.concat([myDiction, tempDf], axis=1)

tempDf.sort\_values(by=['neu/neg', 'pos/neg'], ascending=[False, False], inplace=True) # 分数名列前茅者作为特征词

tempDf.reset\_index(drop=True, inplace=True)

classID = -1

oneKeep = -1

twoKeep = -1

refoIDList = []

classIDList = []

for index, row in tempDf.iterrows():

if (row[1] != oneKeep):

classID = classID + 1

oneKeep = row[1]

twoKeep = row[2]

elif (row[2] != twoKeep):

classID = classID + 1

twoKeep = row[2]

refoIDList.append(index)

classIDList.append(classID)

classIDList = pd.DataFrame(classIDList)

refoIDList = pd.DataFrame(refoIDList)

classIDList = pd.concat([refoIDList, classIDList], axis=1)

tempDf = pd.concat([classIDList, tempDf], axis=1)

with open('key.csv', 'w', newline='', encoding='UTF\_8\_sig') as f:

writer = csv.writer(f)

for index, row in tempDf.iterrows():

writer.writerow(row)

return

def getData(path): #用于读取训练集和测试集的原始csv文件

with open(path, 'r', encoding='UTF-8')as f:

csvReader = csv.reader(f)

dataRaw = list(csvReader)

dataNum = len(dataRaw)

Text = []

Lable = []

for row in range(1, dataNum):

lable = dataRaw[row][1] #读取每一条微博的分类

text = dataRaw[row][0] #读取并规范每一条微博的原文

text = deleteWeiboFormat(text)

text = deleteUserName(text)

text = deleteTopic(text)

Lable.append(lable)

Text.append(text)

f.close()

return Text, Lable #返回内容和标签

def getTextVec(dataPath, path): #用于获取训练集\测试集的输入特征矩阵

retText, retLable = getData(dataPath) #读取数据

data = retText

lenKey = 3883

if (os.access('vec\_' + dataPath, os.F\_OK)): #检验对应输入特征矩阵文件是否已存在

tempDf = pd.read\_csv('vec\_' + dataPath, header=0)

temp = []

for index, row in tempDf.iterrows():

temp.append(list(row)[1:]) #若输入特征矩阵文件已存在，则读取并返回

return retText, retLable, temp

tempDf = pd.read\_csv('key.csv', header=-1)

keyWords = list(tempDf.iloc[:lenKey, 2])

print(keyWords)

print(len(keyWords))

jieba.load\_userdict('addWords.TXT')

stars = ['王俊凯', '华晨宇', '肖战', '朱一龙', '玖', '烊', '张艺兴', 'TFBOYS', '韩红', '魏大勋', '李易峰', '王一博', '刘诗诗', '罗云熙', '吴亦凡']

stop = ['全文', '链接', '”', '】', '）', '》', ')']

repost = ['转发', '轉發', '快转', 'Repost', 'RepostWeibo']

textVec = []

for Text in data: #遍历数据的每一条文本

text = [0]\*lenKey

words = jieba.lcut(Text)

num = 0 #记录一条文本的三种总关键词数

for word in words:

if (word in stop):

continue

if (word.isdecimal()):

word = '数字'

elif (word in stars):

word = '明星'

elif (word in ['##']):

word = '##'

elif (word in repost):

word = '转发'

try:

word2 = keyWords.index(word)

word3 = tempDf.iloc[word2, 1]

text[word3] = text[word3] + 1 # 关键词出现一次，特征矢量对应位置则加一

num = num + 1

except ValueError:

continue

if (num != 0): #如果该句子的该类关键词数不为0，则将频数转换为频率

for row in text:

row = row/num

textVec.append(text)

print(textVec)

print(len(textVec))

temp = pd.DataFrame(textVec) #将特征矩阵存为csv文件，方便以后使用

temp.to\_csv('vec\_' + dataPath, header=-1)

return retText, retLable, textVec

### 5.3 混淆矩阵的可视化

def cmShow(shouldBe, testResult):

classes = ['消极', '中立', '积极'] #坐标刻度

confusion = confusion\_matrix(y\_true=shouldBe, y\_pred=testResult)

indices = range(len(confusion))

plt.xticks(indices, classes) #第一个表示坐标的显示顺序，第二个坐标刻度

plt.yticks(indices, classes)

plt.ylabel('Predicted label')

plt.xlabel('True label')

plt.title('Confusion matrix')

# plt.rcParams两行是用于解决标签不能显示汉字的问题

plt.rcParams['font.sans-serif']=['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

# 显示数据

for first\_index in range(len(confusion)): #第几行

for second\_index in range(len(confusion[first\_index])): #第几列

plt.text(first\_index, second\_index, confusion[first\_index][second\_index])

plt.imshow(confusion, cmap=plt.cm.Greens) #颜色风格为绿

plt.colorbar()

plt.show()

### 5.4 训练与预测

（1）三个二分类器：

# 获取C(2,3)个区分关键词CSV文件

groupData = textFormalize("train.csv")

getGroupWordsFrequency(groupData)

trainText, trainLable, trainVec = getTextVec('train.csv', 'keywords.TXT') #获取训练集

print('Train set has been loaded.')

testText, testLable, testVec = getTextVec('test\_labled.csv', 'keywords.TXT') #获取测试集

print('Test set has been loaded.')

clf = []

lables = ['countNeg', 'countNeu', 'countPos'] #用于构造字符串

for i in range(3):

trainTextVec = []

trainClass = []

flagOther = 0

for row in range(len(trainLable)):

if(trainLable[row]==((i+2)%3)-1): #当训练相邻两类时，剩下的那一类只取一个训练数据

if(flagOther!=0):

continue

else:

flagOther = 1

trainTextVec.append(trainVec[row][i])

trainClass.append(trainLable[row])

testTextVec = []

for row in range(len(testLable)):

testTextVec.append(testVec[row][i])

clf.append(naive\_bayes.BernoulliNB()) #模型选择

print("Training classifier", i+1)

clf[i].fit(trainTextVec, trainLable) #训练模型

joblib.dump(clf[i], lables[i]+'\_bnb.model') #下载模型

clf.append(joblib.load(lables[i]+'\_bnb.model'))

print("Predicting classifier", i + 1)

temp = clf[i].predict\_proba(testTextVec) #获得测试结果的概率值

temp[:, (i+2)%3] = temp[:, (i+2)%3] - temp[:, (i+2)%3] #剩下的那一类的预测概率清零

if(i==0):

score = temp

else:

score = score+temp #概率值相加

result = pd.DataFrame(score) #下载最终概率结果，方便人为分析误差原因

with open('result.csv', 'w', newline='', encoding='UTF\_8\_sig') as f:

writer = csv.writer(f)

for index, row in result.iterrows():

writer.writerow(row)

# score = pd.read\_csv('result.csv', header=-1)

# testText, testLable = getData('test\_labled.csv')

score = result

testResult = []

errScore = 0

shouldBe = [eval(x) for x in testLable] #实际值

for index, row in score.iterrows():

result = list(row).index(max(row))-1 #取概率和中的最大概率者

testResult.append(result)

if(result != shouldBe[index]):

errScore = errScore + 1

print(index, '预测值:',result, '实际值:' ,shouldBe[index], testText[index])

else:

print('OK')

errScore = errScore/len(score) #计算错误率

print('错误率', errScore)

cmShow(shouldBe, testResult) #打印混淆矩阵图

testResult = pd.DataFrame(testResult)

（2）一个三分类器：

groupData = textFormalize("train.csv")

getGroupWordsFrequency(groupData)

trainText, trainLable, trainVec = getTextVec('train.csv', 'keywords.TXT') #获取训练集

print('Train set has been loaded.')

testText, testLable, testVec = getTextVec('test\_labled.csv', 'keywords.TXT') #获取测试集

print('Test set has been loaded.')

clf = naive\_bayes.BernoulliNB() #模型选择

print("Training")

clf.fit(trainVec, trainLable) #训练模型

joblib.dump(clf, 'bnb.model') #下载模型

clf = joblib.load('bnb.model')

print("Predicting")

score = clf.predict(testVec) #获得测试结果的概率值

errScore = 0

score = [eval(x) for x in score]

shouldBe = [eval(x) for x in testLable] #实际值

for index in range(len(score)):

result = score[index]

if(result != shouldBe[index]):

errScore = errScore + 1

print(index, '预测值:', result, '实际值:', shouldBe[index], testText[index])

else:

print('OK')

errScore = errScore/len(score) #计算错误率

print('错误率', errScore)

cmShow(shouldBe, score) #打印混淆矩阵图

testResult = pd.DataFrame(score)

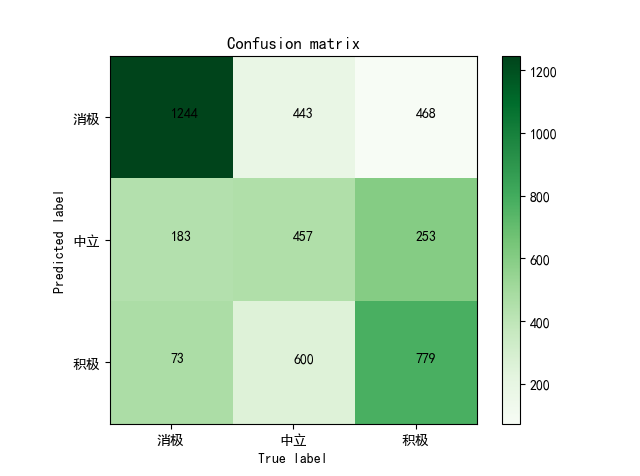
## 六、实验结果

### 6.1 不同训练模型的比较

当特征提取方案为标准差率特征词(800)、训练方案为三个二分类器、测试数据集是test.csv时，六个训练模型的准确率如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| KNN | 随机森林 | 决策树 | 伯努利NB | 多项式NB | 高斯NB |
| 47% | 52% | 50% | 55% | 48% | 53% |

以上六种模型的混淆矩阵类似，伯努利朴素贝叶斯的混淆矩阵如下图所示：



由此计算得到的评估值如下表所示：

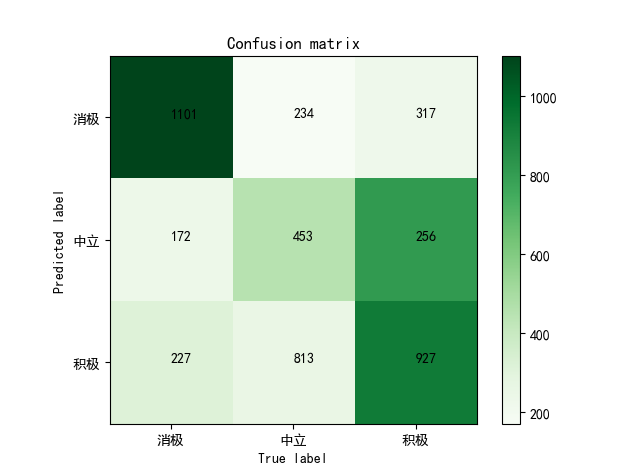
|  |  |  |  |
| --- | --- | --- | --- |
|  | 精准率 | 召回率 | F1 |
| 消极 | 57% | 82% | 67% |
| 中立 | 51% | 30% | 38% |
| 积极 | 54% | 52% | 53% |

### 6.2 不同方案的比较

当训练模型为伯努利朴素贝叶斯、测试数据集是test.csv时，不同方案的方案组合的准确率如下表所示：

|  |  |  |
| --- | --- | --- |
|  | 三个二分类器 | 一个三分类器 |
| 标准差率提取特征词 | 55% | 54% |
| 合并差异度相似词 | 53% | 55% |

其中合并差异度相似词+一个三分类器的混淆矩阵如下图：



由此计算得到的评估值如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 精准率 | 召回率 | F1 |
| 消极 | 67% | 73% | 69% |
| 中立 | 50% | 30% | 38% |
| 积极 | 47% | 60% | 55% |

### 6.3 多项式朴素贝叶斯对特征维度的敏感度

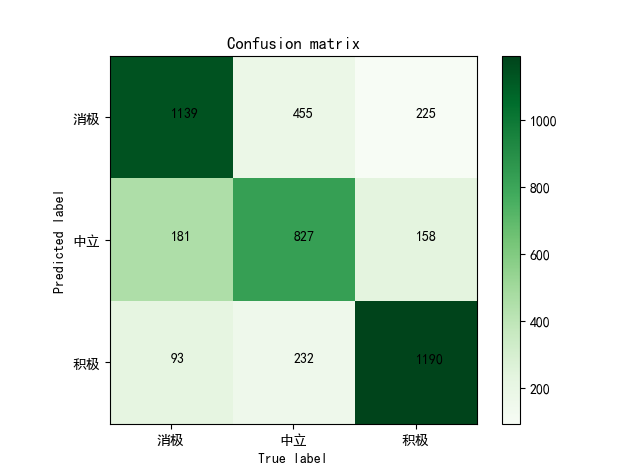
在6.1中，准确率55%的伯努利朴素贝叶斯明显优于准确率48%的多项式朴素贝叶斯。但是在调整输入特征维度时，伯努利朴素贝叶斯的变化很小，几乎忽略不计，而多项式朴素贝叶斯的准确率则有很大提升。多项式朴素贝叶斯的准确率与特征维度的关系如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 特征维度 | 800 | 3000 | 7000 |
| MNB准确率 | 48% | 53% | 57% |

由此可见，虽然低训练维度时多项式朴素贝叶斯没有伯努利朴素贝叶斯的表现好，但是在高训练维度时，多项式朴素贝叶斯的增长空间更大。

### 6.4 采用train数据集测试

对低准确度的结果进行分析后，我认识到test数据集中的部分分类标准与train数据集的分类标准明显不同，且train数据集的分类值更接近常理，所以我决定舍弃test数据集，将train进行随机划分，得到9:1的新训练集和测试集。在维度为7000左右、采用多项式朴素贝叶斯的模型时，得到的混淆矩阵如下图所示：



由此计算得到的评估值如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 精准率 | 召回率 | F1 |
| 消极 | 63% | 80% | 70% |
| 中立 | 71% | 50% | 59% |
| 积极 | 79% | 75% | 77% |

## 七、结果分析

本次实验的训练结果并不让人满意，以下是对工作的不足之处的分析：

（1）无法判定语义的转折和语义的重心：

口语表达中，除了直抑或直扬，常常还有欲扬先抑、欲抑先扬等语义转折。此时语句的重点放在后半段，前半段的表达并不能作为最终情感判断的参考。

想要识别语义重心，有两个明显的难关需要攻破：一是不能单纯以词语频率为特征，还需要包含语句中词语的顺序，目前了解到的是可以用Doc2Vec的办法建立段落矩阵记录词语顺序；二是抓住语义转折的关键词，往往为“不过”、“但是”，但有时也没有任何明显的转折特征词。

以某训练微博为例：#生活小记#要是仅从活在当下的角度来看，我现在过得还是蛮幸福的。但是一旦想长远了，就会焦虑不安以及沮丧。后来想想，未来也是由无数个当下所构成的，何必为了没有到来的未来而放弃了全身心享受当下的幸福呢？? 该微博转折有两次，从贝叶斯频率派的角度，能统计到消极和积极两方面的高频词，使分类变得困难。训练所给标签是中立，不过严格来说按照最后的语义重心应该是积极。

由上例可见，可以按照“但是”“后来”“何必”等关键词为界限，为靠后的词语赋予更高的权重。

（2）无法结合现实时事分析出讽刺含义

语言的表达，尤其是汉语的表达，常常会有来自高语境的晦涩。以本次实验的数据集为例，由于抓取时有“石正丽实验室疑似泄露病毒”事件正在爆发，所以可以分析以下训练微博//@六国虹://@康老狮:不妥，她只是居内夫人。//@章天瑜:不会做夫妻肺片的艺术生特招生不是中国特色正厅级别病毒研究所所长#王延轶舒红兵#?虽看似一本正经甚至有些夸赞，但实际上结合时事知，是在抨击和讽刺，所以所给训练标签是消极。

此类句子在包含高频词时，会由其他明显的文本带动分类结果；但如果用词过于隐晦小众，词频统计就会彻底失效。

（3）对不同的微博主体无法区分

无法区分微博主体的影响主要体现在无法判断是否需要删除/忽略##中的微博主题概括。具体的讲，在微博主体是普通个人用户时，##内的微博主题主要是其他微博大V所写的主题概括，该情感可能与用户本人的情感相悖，如果不忽略则会混淆训练数据；在微博主体是微博大V时，##内的微博主题一般含有用户本人的情感，如果忽略##内的内容，则可能会忽略关键的情感偏向关键词。

（4）难以识别苦乐夹杂中的情感倾向

微博上的个体用户有时用语混乱，消极情绪与积极情绪无规律地夹杂在一起。以下测试例子说明：钉钉打卡，我的噩梦笑死我了哈哈哈哈哈哈哈哈测试集所给的实际值是消极，我的预测值是积极。从该微博地后半句来看，该用户地苦乐夹杂最后还是倾向了积极情绪，我的训练模型由“哈哈”关键词带动预测结果为积极，我人为判断预测值比所给实际值更贴切。

不过在其他关键词重复率不高的“苦乐夹杂”类文本中，我的训练模型也没有较好的预测值。比如//@休闲璐:失去了太多伟大的普通人//@回忆专用小马甲:好难受啊，送英雄//@苍南派:心痛//@迷影心生://@M大王叫我来巡山:损失太大了这么多过劳…真的太辛苦了。致敬 测试集所给的实际值是积极，我的预测值是消极，显然是过于偏重了“失去”等消极高频词。

（5）分词不理想

本次实验分词使用jieba分词，在实际操作时有一些美中不足。影响最大的是分词不够细，比如与“聊”有关的就有“畅聊”“想聊”“聊过”“地聊”“聊上”“聊一聊”“想聊”“聊微信”“聊越”“聊开”等多个相关分词结果，这导致了相同词义本分散在了不同词语，导致特征冗余或特征不明显。我尝试过的解决办法是在jieba源码文件中，修改搜索范围，设定搜索上线为临近的三个字，以此缩小分词的分散性，该方法有一定效果。

由以上总结，本次实验属于NLP典型问题：文本情感分类。在实验过程中，我学会了熟练使用python的字符串操作对数据进行清洗，学会了熟练使用pandas的基本功能来处理数据；了解到了TF-IDF、word2Vec等经典的处理手段，在分析实际应用场景后选择了使用自己设计的特征提取方案；调用sklearn，体会不同模型在同一个数据集上的不同训练质量。

但最终我的实验效果并不好，经过多处的修改调整也没有得到改善，实验过程中也意识到了自己解决方案的局限性。NLP问题一方面要处理和利用好非结构化数据，另一方面要贴合和捕捉非完全客观的训练标签，所以想要在NLP问题上获得较理想的预测结果，我还需要学习更多的实证有效的解决办法，而不是囿于自己有限的临时设计中。

# 参考书籍与参考网址

[1]书籍：

机器学习python实践 by魏贞元

[2]网页：

|  |
| --- |
| 1 情感分析的新方法，使用word2vec对微博文本进行情感分析和分类 |
| 2 短文本情感分析 |
| 3 情感分析简述 |
| 4 使用python+机器学习方法进行情感分析(详细步骤) |
| 5 Python 正则表达式 |
| 6 Python set() 函数 |
| 7 Pandas.DataFrame 的 iterrows()方法详解 |
| 8 Python的递推式构造列表（List comprehension） |
| 9 Jieba分词词性标注以及词性说明 |
| 10 Python中的groupby分组 |
| 11 python 数据合并函数merge( ) |
| 12 numpy.ones用法 |
| 13 Python大数据：jieba分词，词频统计 |
| 14 python如何给字典排序 |
| 15 Python之如何删除pandas DataFrame的某一/几列 |
| 16 python sklearn分类模型算法的学习 |
| 17【计算语言学实验】基于 Skip-Gram with Negative Sampling (SGNS) 的汉语词向量学习和评估 |
| 18 自然语言处理NLP中的N-gram模型 |
| 19 二分类器解决多分类问题 |
| 20 Python+sklearn训练结果保存与加载（以垃圾邮件分类为例） |
| 21 jieba分词最细粒度分词的代码修改 |
| 22自然语言情感分析之jieba分词 |
| 23 逐步视频讲解--用Tensorflow进行中文自然语言处理--情感分析 |
| 24使用Chinese-Word-Vectors作为pytorch中的预训练向量 |
| 25在pandas中遍历DataFrame行 |
| 26 python列表和字典之间的相互转换 |
| 27将sklearn回归分析的预测值输出到CSV文件中 |
| 28 pandas.DataFrame.sample 随机选取若干行 |
| 29对dataframe的行和列进行遍历和修改 |
| 30深度学习自学记录（3）——两种多分类混淆矩阵的Python实现（含代码） |
| 31 sklearn-朴素贝叶斯 |
| 32 python实现中文字符繁体和简体中文转换 |
| 1 <https://blog.csdn.net/meyh0x5vDTk48P2/article/details/78993600?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522159270301619725222413545%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=159270301619725222413545&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-5-78993600.first_rank_ecpm_v3_pc_rank_v2&utm_term=%E5%BE%AE%E5%8D%9A%E6%83%85%E6%84%9F%E5%88%86%E6%9E%90> |
| 2 <https://blog.csdn.net/weixin_30929195/article/details/96507164?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase> |
| 3 <https://blog.csdn.net/weixin_30764137/article/details/98023791?ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%83%85%E6%84%9F%E5%88%86%E6%9E%90%E7%9B%91%E7%9D%A3%E5%AD%A6%E4%B9%A0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-98023791> |
| 4  https://blog.csdn.net/Yan456jie/article/details/52242790?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522159273264919725247630898%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request\_id=159273264919725247630898&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_ecpm\_v3~pc\_rank\_v2-2-52242790.first\_rank\_ecpm\_v3\_pc\_rank\_v2&utm\_term=%E6%83%85%E6%84%9F%E5%88%86%E6%9E%90+nlp+%E6%9C%89%E7%9B%91%E7%9D%A3%E7%9A%84%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0 |
| 5 <https://www.runoob.com/python/python-reg-expressions.html> |
| 6 <https://www.runoob.com/python/python-func-set.html> |
| 7 <https://www.jianshu.com/p/14c054225f03> |
| 8 <https://www.cnblogs.com/reanote/p/python_list_comprehension.html> |
| 9 <https://blog.csdn.net/enter89/article/details/80619805> |
| 10 <https://blog.csdn.net/qq_32618817/article/details/80587228> |
| 11 <https://blog.csdn.net/qq_21840201/article/details/80727504> |
| 12 <https://blog.csdn.net/BBxxf90/article/details/84927995> |
| 13 <https://www.jianshu.com/p/f516292d8b9c> |
| 14 <https://www.py.cn/faq/python/12154.html> |
| 15 <https://www.cnblogs.com/rrttp/p/8116805.html> |
| 16 <https://blog.csdn.net/huoyingchong64/article/details/89918362> |
| 17 <https://www.cnblogs.com/yanqiang/p/12109901.html> |
| 18 <https://blog.csdn.net/songbinxu/article/details/80209197> |
| 19 <https://blog.csdn.net/u012370185/article/details/90205231?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase> |
| 20 <https://mp.weixin.qq.com/s?__biz=MzI4MzM2MDgyMQ==&mid=2247489737&idx=1&sn=f40e3e1eced43aef193b065c992c9be7&chksm=eb8ab393dcfd3a853f97ac0a01751f4baf198fd152f45adfd7ffb7204900a492a002c119cbbc&token=1074581413&lang=zh_CN#rd> |
| 21 <https://zhuanlan.zhihu.com/p/41213689> |
| 22 <https://blog.csdn.net/ziyonghong/article/details/83926800> |
| 23 <https://blog.csdn.net/weixin_41947081/article/details/82156183> |
| 24 <https://www.jianshu.com/p/0385b1c6b94b> |
| 25 <https://blog.csdn.net/ls13552912394/article/details/79349809> |
| 26 <https://blog.csdn.net/loner_fang/article/details/80940600> |
| 27 <https://blog.csdn.net/weixin_43919570/article/details/104252947> |
| 28 <https://blog.csdn.net/zhengxu25689/article/details/87347700?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase> |
| 29 <https://blog.csdn.net/starmoth/article/details/84779622> |
| 30 <https://blog.csdn.net/qq_40784418/article/details/105611821> |
| 31 <https://blog.csdn.net/Forlogen/article/details/85126210> |
| 32 <https://blog.csdn.net/weixin_43336281/article/details/106162631> |

# 数据修改清单

|  |  |  |
| --- | --- | --- |
| 原 | 现 |  |
| 0 | 1 | 记一下目前为止看过的视频课《系统解剖学》中国医大霍琨我大一上的偶像，但是老师退休了吧从来没在学校见过本人，超想见本人！跟着老师手绘在图上画，很有助于理清思路，尤其是神经系统那一块，救我狗命。老师讲病例也会画图，小脑共济失调，核上瘫核下瘫什么的，印象深刻。《生物化学与分子生物学?展开全文c |
| 0 | 1 | 中国建筑使命必达！工人辛苦夜以继日# #2郑州·郑州航空港区人民医院L77YUJ的微博视频? |
| 0 | 1 | 平安健康# #那些奋战在一线的医务人员们，请你们一定要照顾好自己，我们都在！ |
| -1 | 0 | 传播途径还是说的不全面，粪便如果可以传播一定要做好消毒。这个很隐藏的！ |
| 0 | -1 | 70多天前，柳叶刀的论文就发布了……你想吐吗？这样的时间线佩服了！? |
| 0 | -1 | 说出来不怕兄弟们笑话我一天整这么多活发这么多视频其实一方面想逗你们笑另一方面是想把自己注意力给尽量移开一些今天一整天我都很不舒服我没有视频里面看上去这么开心这么多骚话视频都是在上班空闲时间录的其他时间我除了靠着小太阳取暖和做了点工作一点别的想法都没有了整个人已经没啥动?展开全文c |
| 0 | 1 | 这期蛮好哭的。。去接送医护人员上下班的志愿者小伙子给医护人员做饭的饭店姐姐云南运香蕉的村民火神山雷神山昼夜不停的建筑工人们他们都没有计报酬只是想要尽可能的发光发热做到来自普通人的一点帮助//@上流有个大胖子:真的感动于好多好多朴实的人带着善意去奉献自己能做的最大努力 |
| 0 | 1 | 【#手绘武汉日记#：等我们好了，一起去看樱花】“武大的樱花马上就开了，等我们好了，一起去看樱花，吃热干面！”来自福建中医药大学附属人民医院的护士长陈水凤，目前支援金银潭医院，病人对她说的这句话被她记录在日记本里。她说，等你们病好了，我们就回家了。（记者孟昭丽?展开全文c |
| 1 | 0 | 分享：返工潮上海以监督检查督促织密基层防控网————要闻——中央纪委国家监委网站O网页链接 |
| -1 | 1 | 太心疼！边防巡逻路上战士突然晕倒…每次巡逻都是生死考验，他们却无怨无悔地守卫祖国！致敬边防军人！L再也不孬了的微博视频? |
| -1 | 1 | 莫名好笑哈哈哈哈哈哈? |
| 0 | 1 | 改成2月7号播出华晨宇歌迷温馨提示：请注意勤洗手通风，请出门戴好口罩，有发烧感冒咳嗽和其他不适症状请立即就医，让我们携起手来，共渡难关！在此和华晨宇一起祝大家新年快乐，万事如意，平安健康！#歌手华晨宇# |
| 0 | 1 | 和@UNINE\_姚明明一起为武汉加油，为中国加油，大家都照顾好自己，照顾好家人！//@姚明明全球后援会:姚明明ymm#姚明明鼠年行大运#抗击疫情！?和@UNINE\_姚明明一起向武汉一线的医务人员致敬！等疫情过去，我们共同期待春暖花开 |
| 0 | 1 | 今天也差不多是这个点我爱我的志愿者们准备自费奖励她们！明天还有最后一天！嘎油！? |
| -1 | 0 | 0点【拍2件19.8】创尔美多效修复面膜2片拍2件19.8，多效面膜和祛痘面膜2款可选，4.9的高好评，主打祛痘淡印/补水保湿/收缩毛孔/胶原修护，这款不含酒精，敏感肌肤也适用，膜布比较厚，和别的医美面膜比起来还要大一圈，适合我们大脸星人? |
| -1 | 1 | 病毒变异如此之快，微观世界造反，想要吃掉宏观？可惜找错了地方，这里是中国！休想！//【重庆出现四代感染病例传染链条延长】O重庆出现四代感染病例传染链条延长? |
| 1 | 0 | 【￥19.9】JICE净安酒精免洗多功能除菌消毒喷雾100ml，含Q12除菌因子，99.99%除菌消毒，植萃除菌，温和洁净，设计轻便，方便随身携带！二合一：O网页链接? |
| 0 | -1 | 而且非要让我们写这种奇怪的东西，我根本不想了解李文亮啊… |
| 0 | 1 | 保佑 |
| 0 | 1 | #重庆快新闻#【在重症感染区连续工作四五个小时，重庆首批援鄂医疗队这样战疫情！】1月27日早晨8时许，刚从病区出来的新桥医院张诚副教授通过微信语音讲述自己值守第一个夜班的感受：“以前我们在不同科室岗位，这次临时抽组搭档，很快就磨合配合默契，在首个夜班顺利完成了患者收治，现在我只想回去好?展开全文c |
| 0 | -1 | 拜托了不要逼我口吐芬芳了！明知道自己有生病还瞒报祸害别人玩呢啊？ |
| 0 | -1 | 《半个喜剧》几乎全场都在掉眼泪，刚开始就把自己代入角色，我以为我已经没有眼泪了，跟前男友分手，他远在澳洲读书，分手三个月听说他有了新的女朋友，听说他换了新的房子，听说他过得很好，我在想他有没有想过我，这段时间我是怎么熬过来的，分手第三天，家里人来电话，家人重病进重症监护室，我甚至?展开全文c |
| -1 | 1 | 跳20米高悬崖、高原藏区深夜突袭！这位“不要命”的缉毒警让毒贩闻风丧胆　从四川大凉山到境外“金三角”，从羊肠小道到深山密林，他的名字让毒贩闻风丧胆。他曾带队深入“金三角”腹地，跨境抓捕大毒枭；他曾纵身跳下20米高悬崖，抓捕嫌疑人；他曾多次在公路上上演生死时速，在海拔4200米的高原藏区?展开全文c |
| 0 | -1 | 1月20号之前的错暂且不论，1月20号之后武汉市的应对措施仍然糟糕至极，一点都没有我们的政府这些年来应对重大事件的沉稳高效，唯一拿得出手的不过是火神雷神，但那是最容易做的决定，因为有小汤山做样板，不过是砸钱而已。荒唐至此，唯一的解释是F4已经没心情救灾了，只想安全下庄? |
| 0 | -1 | 德约科维奇出来解释一下为什么澳大利亚的空气那么清新，运动员却要退赛？？ |
| 0 | 1 | 养生经济《中国工程院院士张伯礼：我们期盼的疫情拐点将要出现》出院的病例超过了死亡的病例，这一点显示了非常好的前景，也让我们看到了希望。O中国工程院院士张伯礼：我们期盼的疫情拐点将要出现#养生经济##疫情拐点将出现##新型冠状病毒可通过接触传播##疫情仍处于扩散阶段##疫情结束见面谁也别笑谁胖# |
| 0 | -1 | ?/??????s???因为最开始说的不会人传人我们根本没把冠状病毒放在心上可是这几天感染人数蹭蹭的往上升看着报道看着实时刷新的数字心都慌SARS的时候我还小我没有什么印象也不知道害怕可这次不一样这次我已经有了自己的认知能力我知道害怕我希望家人都能平平安安我不想出门也不?展开全文c |
| -1 | 0 | 【#武汉全民体温监测#：上门排查与自查自报相结合】武汉市即日起将开展全民体温监测工作。每日每户每人监测一次并电话、微信报告，体温异常者由工作人员上门检查。发热或相关症状人员，由社区组织其及时到社区卫生服务中心诊断筛查，根据诊断情况分类集中收治或隔离医学观察。如遇不配合排查工作的市民?展开全文c |
| -1 | 0 | 无症状感染者有没有传染性和病毒变异官方目前未发现#最新疫情地图##漏报信息致患者自缢武汉多人被处理#L我爱徐滨的微博视频? |
| 0 | 1 | “武汉是能过关的，武汉本来就是一座很英雄的城市。”他最后哽咽那里我也差点没忍住??发热依旧是最主要症状！早发现早隔离！9分钟不长！看完我真的放心好多 |
| 0 | 1 | #羽生结弦#请回答1988里的阿泽党快快来入坑，羽生结弦就是现实生活中的阿泽2西安? |
| -1 | 0 | #全国确诊新型肺炎病例#截至1月30日24时，全国累计报告确诊病例9692例，现有重症病例1527例，累计死亡病例213例，累计治愈出院病例171例，共有疑似病例15238例。02YlQgC06。? |
| 0 | 1 | 过了这么多年，我终于发现了那些让我喜欢的人的特质，他们总是生机勃勃的，有自己喜欢的事情，不过多的干涉别人的生活，懂得人际交往的边界，不四处炫耀自己所拥有的一切，言辞温和，与人为善，适度的野心。容得下生命的不完美，也经得起世事的颠簸，将人生的一切都根植于生活。 |
| 0 | 1 | #共同战疫#【一起听！#致敬疫情前线的军人#】“没有不愈的伤，没有过不去的荒凉，穿过黑夜看到明天的阳光！”海陆空军医除夕夜驰援武汉，退役军人参建火神山医院，军医穿尿不湿进病房……军装穿在心，疫区是他们逆行的方向！@央视军事推出原创歌曲《逆行的方向》，为战友加油！请平安！盼凯旋！?展开全文c |
| 0 | 1 | #全国疫情总体呈下降趋势#威力还要再打2350个馕共计4000捐赠给华阳街道#成都爆料。L魔画精灵的微博视频? |
| 0 | 1 | @人民日报:【人民微评：保护好医务人员的身心健康】#武汉协和14名感染医护出院#，多么温暖而又振奋人心的好消息！防控疫情，医护人员冲锋在第一线，付出多，压力大。但他们依然坚守岗位，用专业知识展现着爱与责任。敬佑生命、救死扶伤、甘于奉献、大爱无疆，这就是最可敬的人！ |
| 0 | 1 | #疫情期间让你明白了什么#保护野生动物，维护生态安全保护野生动物，实现人与自然和谐共处保护野生动物，停止杀戮保护野生动物，从不吃野味做起保护野生动物，保护我们自己#78%人类新发传染病与野生动物有关#? |
| 1 | 0 | #延迟复工工资发放新政策##劳动法##劳动法问答##网络民工讨薪##劳动最光荣##劳动节##工资#2月12日，19:30-20:30，公益普法《疫情防控期间企业用工管理十大高频、热点、难点问题解析》，免费预约直播→O网页链接讲师：郝云峰律师，劳动法分会常务理事，北京东合?展开全文c |
| -1 | 0 | "脱髓鞘病变的病因与护理　　1、脱髓鞘病变的病因的症状　　多青壮年发病，病前两周内有上感呼吸感染症状,或疫苗接种史。有受凉、过劳、外伤等发病诱因。首发症状为下肢麻木、无力，病变相应部位背痛和束带感，尿潴留和大便失禁。　　2、脱髓鞘病变的护理　　避免诱因：　　外伤，劳累，激动?展开全文c" |
| 0 | 1 | 把相册从头到尾翻了一遍。从17年到20年。两千多张照片。记录着每天的生活。从武汉到温县。在武汉的时候想回来家。回家倒也想念在武汉的生活。每天充实高兴。工作之余都是吃吃喝喝。躺在被窝里聊通宵。只是觉得安心。没有其他的烦恼。那时候总觉得自己是最幸运的。现在自己是一个大人了。没有谁会义务的?展开全文c |
| 0 | 1 | ????亲爱的宝贝夫君张涤????好啊，你又玩失踪啦??你再不出现，我可要打110找你了??你可知，没有你的日子，再美好的时光都觉得孤单乏味。爱你爱到心碎，想你想到沉沦??我喜欢你折腾我，，我可舒服了??，，??小寒日初长，风吹腊梅香??年味渐浓，人心愈暖??????????????展开全文c |
| 0 | -1 | #最新疫情地图#男子出门不带口罩，还称自己在政府上班！门卫说：政府让戴口罩！男子说：政府让你吃屎你吃屎？L吐槽曝料的微博视频??? |
| 0 | -1 | @林耘:拘了送他吃“免费饭” |
| 1 | 0 | 【16.90】敬修堂佰花方免洗抗菌喷雾液酒精75度喷雾杀菌洗手液消毒液2小时疯抢1469件不是你需要的？自助查找心仪商品：O网页链接75%的免洗抗菌液，消字号卫生许可，特殊时期的刚需物品，线上线下都被抢疯了，杀菌率超高！有效预防病毒，防止感染，消毒杀菌，为?展开全文c |
| -1 | 0 | 【#上海确诊1例新型冠状病毒感染肺炎#自武汉来沪】1月20日晚，国家卫生健康委确认上海市首例输入性新型冠状病毒感染的肺炎确诊病例。患者为56岁女性，湖北省武汉市户籍。1月12日自武汉来沪后，因发热、乏力等症状，于1月15日在本市一发热门诊就诊后即被收治入院隔离治疗。经上海市疾控部门检测，并经?展开全文c |
| 0 | 1 | 让爱留守『大山小爱』年味十足，一群志愿者像个小孩子一样童心未泯，怀揣着最初的梦想，砥砺奋进，只争朝夕，不负韶华！漫天的烟火诉说不尽志愿者们的赤子之心。大家都努力，为明天的联欢会积极准备着。#大别山春节联欢会##大山小爱##骆小胖##青春生院##青年生院说##与梦齐飞#?展开全文c |
| 0 | -1 | 连续两周周日上班了，我感觉自己快要爆炸了。现在心里只想着一件事，那就是回家过年? |
| -1 | 0 | 这也就能解释，为什么此前《财新》报道中称，黄冈市蕲春县县长在1月20日的全县病毒性肺炎防控大会上讲话时指出，当时黄冈市病毒性肺炎病例已达109例，但当时黄冈还未出现一例确诊患者——因为当时黄冈还没有试剂盒，就算有，黄冈当地也无权限确诊。『试剂盒供不应求，武汉新型冠状病毒肺炎确诊之难』?展开全文c |
| 1 | -1 | 头疼了一整天好怕自己得病了不停测体温晚上吃晚米酒鸡蛋好了? |
| -1 | 0 | 【防疫战疫：普通大众心理健康指南】面对新冠病毒肺炎，社会公众开始出现各式各样的心理反应。从疾病流行初期的不够重视，心存侥幸，到产生同恐惧、过度紧张。那么，我们到底应该怎么做呢？1、及时了解新型冠状病毒肺炎的性质，掌握流行情况，做一以心中有数；2、采取积极有效的预防措施；3、有症状及?展开全文c |
| 0 | -1 | 怕找遍吴家都无比有几个人知道这门子亲戚。　　可就是这么一个人，竟然敢在外#若松问凝#面打着吴家的旗号招摇撞骗？　　这事要让吴老爷子知道了那还了得？最轻也要申饬吴凌柏老#金银潭医院首批中医药参与治疗患者出院#? |
| 0 | 1 | 世界那么大，每个人都有自己的生活，但我还是喜欢我自己的。经历过的都是我的，谁的生活，我都不换。一件小事，不算故事。? |
| 1 | 0 | 【两部委联合印发《关于建立跨省流域上下游突发水污染事件联防联控机制的指导意见》】 |
| 0 | -1 | //@-----yuki-----://@无敌惠惠子://@火野映司厨\_:太自私了//@尛柒\_Capricorn://@苏东东仔:医务人员罢工，民众反对设隔离区，好不容易给的口罩拿来示威游行……香港这些年都教出一群什么玩意儿？ |
| 0 | 1 | 感恩奋战在一线的医务人员??周老师你也要照顾好自己??新年快乐??我宇宙无敌爆炸爱你??//@R1SE-周震南:谢谢你们?? |
| 1 | 0 | 新年换新衣，内衣也一样哦，要换新。如果是无钢圈胸罩的话，你是否会认为胸部会很下垂?错错错华歌尔的「GOCOCi」虽然是无钢圈，但是因为用了花生型内垫设计，可以保持美胸哦。无论是穿着的感觉还是手感，特别特别棒！运动、居家时，请一定要试试哦!?2日本·东京? |
| 0 | -1 | @小D1980水灵灵:#武汉排查出1499名重症患者全部入院#看看吧，这才是人间，新闻上都是啥？ |
| 1 | 0 | 春节即将来临，为切实做好2020年春节节前安全工作，强化施工现场的安全管理，消除辖区重点部位安全隐患，2020年1月8日上午，由街道裴副主任带队，街道安监站联合民航路社区工作人员对辖区环城巷瑞思贝幼儿园装修施工工地进行安全检查。检查中，裴副主任要求建设手续完备后方可开工，并指导学校负责人?展开全文c |
| 0 | -1 | 本想着放假了就能多见见你，能够打破我们之间的那种尴尬，让原本就不熟悉的我们更加熟悉些，结果，啊哈哈哈哈哈哈哈哈，因为一个嘴馋的人儿，感觉要大结局了，或许疫情好了我们都该回去上班了，也没几次见面的机会，希望下次不再有一些列的事儿，让我失去多接触你了解你的机会，? |
| 1 | 0 | @Astrobots星际雄兵很好玩的一款玩具，居家旅行必备玩物。? |
| 0 | 1 | wokxswl哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈//@钪氪:笑死我了哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈笑到咳嗽哈哈哈哈哈哈哈山东tai～产 |
| 1 | 0 | 像石正丽的研究确实很高端的，她研究这种东西可以轻易发在自然和科学这种高级别的科学期刊。为什么呢？因为这种高感染性的病毒，西方实验室不太给你研究的，怕感染。来自学术科学界的奖励让她有动力做这种美国的科学伦理不鼓励或不允许做的实验。中国这块似乎没有太多的约束。 |
| 0 | 1 | #最新疫情地图#希望越来越好啊? |
| -1 | 0 | #海淀知事#【北京一#四世同堂家庭9口人患病#】2月14日下午，北京市新型冠状病毒感染的肺炎疫情防控工作举行第二十场新闻发布会。北京市疾病预防控制中心副主任庞星火通报了一起四世同堂家庭的聚集性疫情。庞星火介绍，该家庭共有7口人，1月中旬，3个亲戚从外地途经武汉来京探亲，后导致这家人得病。?展开全文c |
| 0 | -1 | 老实说日本政府这操作真的气到不知道骂什么好，现在这游轮就是个病毒培养皿！//@进击的安利:能不能要求至少把船上的中国人接下来单独隔离？厚劳省太不可靠了，不能让他们制造人祸啊！ |
| 0 | 1 | 县医院心血管科全体医护人员，以全身心付出，让每位病人有了家的感觉。我在心血管科住院诊诒，前天我出院心血管科有位不知姓名亲自用轮椅推我到医院大门口，方便我乘车。我建议心血管内科主要领导应通知该院长点名表彰奖励！欢匝搏友们点评与转发，这多好时题材我怎么就设想到！！！！！ |
| -1 | 0 | 易烊千玺让我成为更好的人第23天毛笔字√今天处理的事情很多，网上办公真的觉得心理很累，但是还是一边做一边劝自己。洗漱完做了会儿运动，向拥有平坦小肚子迈进的第一天。 |
| -1 | 0 | 【女子从武汉返乡19天至今无症状父母姑姑等5名亲戚确诊新型肺炎】这个案例比较特殊，19天她自己没有症状，但传染给了5名家人。对此，当地疾控中心表示，该女子虽无症状但她是病毒携带者，有接触者均已隔离。之前我看报道说，潜伏期不会超过14天，请专家解读一下，这是怎么回事？我觉得，人类还没有?展开全文c |
| 0 | -1 | //@小虫甲://@-沉川溪行-://@蓝桥春雪OuO://@作家陈岚:锦麟因红斑狼疮引起肺部感染死亡。//@罗蕾莱lorelei:居然没人转，居然没人转。看完这个新闻我手都是抖的。以前认识很多红斑狼疮的小朋友，不及时治疗会发热、溃烂、浑身上下破败不堪、身体四处都是感染的创口，连喊疼的力气都没有了！而这个慈 |
| 0 | -1 | 霓虹的心有多大啊//@当我统治世界我就去种花:无语//@G\_O-C:日本居然认为无症状的就不传播，竟然不计算密切接触者……请问是中日没通网么？中国这边好几个毒倒全家才排查出来的毒王了解一下？ |
| -1 | 0 | 我刚使用会员特权送给了@西藏昌都人韩红5朵鲜花，希望陪你从开始到未来的每一个高光时刻，我们一起走花路吧！西藏昌都人韩红已收到12339朵花? |
| 0 | 1 | 共抗疫情，华邦在行动！[拥抱]（驰援注射用甲泼尼龙琥珀酸钠，正是治疗肺炎重症病患急需的药品！）O重庆邮政运2吨多医药驰援武汉来自上市公司华邦健康? |
| -1 | 0 | #你平安回来我承包一年家务#【疫情期间如何解决焦虑情绪？】人类的悲喜不相通，焦虑却出奇一致...害怕安静、逃避独处，陷入恶性循环的情绪怪圈？英国哲学家AlanWatts教你解决之道：停止强迫性思考和内耗情绪！#叉酱铲酱##疫情拐点将出现#L英国普罗派乐卫视的微博视频???O网页链接? |
| 0 | -1 | 皮肤过敏也就算了我还上火了一直咳嗽觉得自己……emmmm? |
| 1 | 0 | 新型冠状病毒可以通过接触传播。无论是返程途中，还是日常工作生活中，勤洗手、勤消毒等防范措施很重要。那么，一般来讲，日常生活物品中哪些是适合病毒存活的“温床”？病毒能在这些物体表面存活多久？下面这些相关研究或许能提供参考。 |
| -1 | 1 | 保护穿山甲宝宝//@angelababy:为了维护大自然的多元生态，也为人类自己的健康，保护穿山甲，拒绝野味！ |
| 1 | 0 | 实力 |
| 0 | 1 | #武汉将再建一个小汤山医院#因为未来是你自己的，只有你自己能给自己最大的安全感。别忘了答应自己要做的事，别忘记自己想去的地方，不管爱情那有多难，有多远。? |
| 0 | 1 | 【助力战“疫”，国泰“疫”不容辞】#上海战疫##上海加油#“生命重于泰山，疫情就是命令，防控就是责任。”救死扶伤的医务人员义无反顾地“逆行”深入疫区；建筑工人连夜奋战，誓要再现“小汤山”的奇迹；解放军子弟兵心系人民，千里驰援要打赢这场没有硝烟的战争。O助力战“疫”，国泰“疫”不容辞? |
| 0 | 1 | #任嘉伦时尚芭莎明星电子刊#武汉加油//@任嘉伦全国后援会:任嘉伦#任嘉伦时尚芭莎明星电子刊#感谢@时尚芭莎，感谢所有参与的嘉人，跟@任嘉伦Allen一起驰援湖北，奉献爱心?????? |
| 0 | 1 | //@西藏昌都人韩红:#韩红爱心驰援武汉##抗击新型肺炎我们在行动#官宣。 |
| 0 | 1 | #点亮心愿岁岁平安##大连医护人员驰援武汉#此刻大连，四架包机，全市500名医护人员将于21:30起航，连夜飞赴湖北！L德州云中心的微博视频? |
| 0 | 1 | //@湘南幼专团委:【新闻之家】#湖南102人医疗队驰援湖北#湖南省向湖北省派出的第八批医疗队于今晚驰援湖北省武汉市，目前#湖南累计援鄂医疗队员1059人#祝平安凯旋！武汉加油，中国加油！#湘南幼专##湘南幼专团委# |
| 0 | 1 | //@斯年之殇yeah:#正能量偶像孟美岐#武汉加油！孟美岐你一直是个行胜于言的人，微博发了好多条提醒我们保护好自己，私下又默默为武汉捐款。误解你的人从来就不懂你。追星要的不是攀比，要的是那一刻的美好，希望大家都能不忘初心//@孟美岐LoveMQ应援团:#韩红爱心驰援武汉#抗击肺炎，驰援捐助！ |
| 0 | 1 | #上海新冠肺炎病因诊断专家组奔赴武汉#《祖国不会忘记》武汉别怕，我们来了！除夕之夜，他们驰援武汉。不辱使命，坚决完成救援任务。山东各大医院均派出多批医疗队出征武汉烟台业达医院、烟台山医院、毓璜顶医院、烟台莱阳中心医院等。#全国重症医务人员近10%投入武汉#华东医院、华山医院、第十人民医?展开全文c |
| 0 | 1 | #防控疫情三亚在行动#【央视报道三亚驰援湖北温暖举动】连日来，三亚派出首批医疗队及援助蔬菜物资驰援湖北、指定两家酒店安置滞留湖北的游客……1月29日，央视《新闻联播》《晚间新闻》《东方时空》等栏目对三亚的温暖举动进行了报道。29日晚，央视《新闻联播》报道称，最近这段时间，由于防范疫?展开全文c |
| 0 | 1 | 【李兰娟院士除夕再进京！】今天，很多人朋友圈里被李兰娟院士这张出行的照片刷屏，配文说：今日浙江大学李兰娟院士率领浙江首批135名医务人员驰援武汉，医者仁心，向最美逆行者致敬。钱江晚报记者随后从李兰娟院士处了解到，李院士今天出行不是去武汉，而是赶往北京参加和此次疫情有关的会议。今天除?展开全文c |
| 0 | 1 | 【泪目妈妈给上前线军医儿子录制鼓励视频：“盼望你早日平安归来”】“儿子，当妈妈知道的时候，你已经踏上征途，面对疫情挺身而出责无旁贷。虽然有点担心，但是妈妈坚决支持你。”大年三十晚上，空军军医大学医生史庆辉接到驰援武汉的命令。而这天，他的母亲刚刚出院回家。白衣战士们，我们都在等?展开全文c |
| 0 | 1 | 【星夜兼程！#货车司机900公里驰援雷神山#】#为抗击疫情默默付出的人#李加权是江苏泗阳的一名货车司机，近日，他载着一车紧急物资，包括防护服、口罩、呼吸机和一辆救护车，从上海出发到武汉，全程900多公里。经过18个小时星夜兼程，物资成功运抵雷神山医院。这一路被他用手机记录了下来。?展开全文c |
| -1 | 0 | "我觉得重大时间节点是1月初，已经成为有医生得了相同的疾病住院了。柳叶刀最早的论文是12月1号，但因潜伏期长，有种传染性低的错觉。12月的时候，作为官方是有侥幸心理存在的。已经有人传染人的情况，但案例太少，可能是暴露在相同感染源的情况。可这些资料直接影响R0,潜伏期以及传播途径的判定" |
| 0 | 1 | “娜娜，咱俩娃都乖得很，家里也都好着呢，你放心。你可一定要做好防护啊，能休息了就抓紧时间休息......”陕西高速西汉机关党支部的惠双权正在与驰援武汉奋战一线的妻子对话。#抗击新型肺炎我们在行动#三秦都市报全媒体#唐朝报道#O风雨同袍携手战“疫”? |
| -1 | 0 | 我参与了@优酷花花万物发起的投票【「家庭一角」投票】，我投给了“骨灰级球鞋发烧友的鞋柜”这个选项，你也快来表态吧~@X玖少年团谷嘉诚BAE |
| 0 | 1 | 我参与了@财经网发起的投票【你们公司/单位支持网上办公吗？】，我投给了“支持，我们是良心企业！”这个选项，你也快来表态吧~ |
| 1 | 0 | 我免费围观了@孔庆东的回答，问题价值388.00元，快来一起围观~O微博问答? |
| 0 | 1 | 【你保护大家，我们守护你！#白衣天使加油#】#抗击疫情央企行动#1月31日，中国节能环保集团新时代健康产业（集团）有限公司向北京医院捐赠100台空气净化器，为无私奉献的一线医务人员提供进一步保护与帮助！L国资小新的微博视频??? |