# “智慧政务”中的文本挖掘应用

## 摘要

随着互联网平台的高速发展和 5G 时代的加速到来，越来越多群众通过网上政府平台发表自己的意见和建议，各类社情民意相关的文本数据量不断攀升，这根以往主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了极大挑战。同时，随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，面对复杂庞大的数据，政府部门通过互联网相关工具处理数据，解决回应民生问题是很有必要的。这样不仅能提高政府部门工作效率，还能提高及时解决各类民生问题，更好的服务群众。

针对任务一，通过朴素贝叶斯预测留言的分类。首先对留言进行去符号化，再利用 jieba 中文分词工具对具体留言进行分词，通过 TF-IDF 算法构建词频矩阵并计算好 TF-IDF 向量，最后用训练好的模型对测试集进行分类（朴素贝叶斯文本分类）。

针对任务二，通过 jieba 分词将留言进行去停顿词，再将分词后的词语转化为 TF-IDF 权重向量，再用余弦相似度依次进行比较归类，在将归类好的结果按照问题出现的次数、点赞、反对等进行排名得出最终结果。

针对任务三，首先根据题目中所要求的相关性、完整性、可解释性三个方面构造一套可用的评分公式并对文本留言进行初步评分。第二步则是通过 jieba 分词构建词组并转化为 TF-IDF 词向量，构建高斯朴素贝叶斯模型。最终通过模型预测的结果和基本评分结合得出最终判断。

**关键词：**自然语言处理技术、机器学习、TF-IDF、朴素贝叶斯模型

# Abstract

With the rapid development of Internet platforms and the acceleration of the 5G era, more and more people are expressing their opinions and suggestions through online government platforms, and the volume of text data related to various social and public opinion is constantly rising, which has brought great challenges to the work of the relevant departments, which used to rely mainly on the manual division of messages and hotspots collation. At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of intelligent government systems based on natural language processing technology is already a new trend in the development of social governance innovation, in the face of complex and huge data, government departments through the Internet related tools to process data, solve the response to people's livelihood problems is very necessary. This will not only improve the efficiency of government departments, but also improve the timely resolution of various livelihood problems and better serve the public.

For Task 1, the classification of the message was predicted by the Plain Bayesian First, the message is de-symbolized, then the specific message is subdivided using the JIEBA Chinese lexical tool, the word frequency matrix is constructed by the TF-IDF algorithm and the TF-IDF vector is calculated, and finally the test set is classified using a trained model (Naive Bayesian Model).

For task two, through the jieba participle will leave a message to de-stop the word, and then the words after the participle into TF-IDF weight vector, and then the cosine similarity in order to compare and categorize, in the categorization of good results according to the number of times the problem, the number of likes, dislikes, etc. to

rank the final result.

For task three, a set of usable scoring formulas was first constructed and a preliminary scoring of the text message was given based on the three aspects of relevance, completeness and interpretability required in the title. The second step is to construct a Gaussian Naive Bayesian Model by constructing lexical phrases from jieba participles and transforming them into TF-IDF word vectors. The final judgement was eventually arrived at through a combination of the results predicted by the model and the basic score.

**Keywords:** Natural Language Processing, Machine Learning, TF-IDF, Naive Bayesian Model

# 目录

一、问题分析 5

二、数据准备 5

2.1 中文分词 5

2.2 去除停用词及其余干扰因素 6

2.3 人工筛选分类 6

2.4 建立基本评分公式 6

三、算法简介 7

3.1 TF-IDF 7

3.2 TextRank 8

3.3 JIEBA 分词 8

3.4 MultinomialNB 9

3.5 GaussianNB 9

四、程序设计 10

4.1 群众留言分类 10

4.2 热点问题挖掘 14

4.3 答复意见的评价 15

五、参考文献 18

## 一、问题分析

问题共给出 4 个附件数据，附件 1 提供了内容分类三级标签体系，附件 2 提供了群众留言相对应分类情况，附件 3 提供了群众留言的赞成反对数据，附件 4 提供了政府的回复情况。

问题一要求对留言进行分类，根需据附件 2 给出的数据，建立关于留言内容的一级标签分类模型。留言的分类需要根据具体的留言内容而分析出是属于什么类别的，所以可考虑用留言中的关键词的出现频率来确定留言的标签。

问题二要求根据附件 3 将某一时间段内反映特定地点或特定人群问题的留言进行归类，定义合理的热度评价指标，并将评价结果按相应要求保存。需要定义热度评价标志用于数据排序，得到热度排名前五的数据 需要考虑到相识问题数量，相识问题的总点赞数以及相识问题的总反对数，用三个条件来判断数据的热度；将地点和人群进行提取出来，地点或人群主要分布在留言主题里面，可对留言主题进行分析得到相应的地点或人群

问题三要求针对附件 4 相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等角度对答复意见的质量给出一套评价方案，并尝试实现。其中相关性就需要考虑答复意见的内容是否与问题相关，完整性指的是答复是否满足某种规范，可解释性则要看答复意见中的相关解释的完整是否充足以及是否具有针对性。因此问题三需要从多方面考虑才能得出较为符合规范的评价。

## 二、数据准备

### 2.1 中文分词

在机器学习中，计算机是无法对一整串的字符串进行分析处理，故通常需要将各种留言分割成诸多词汇，再将词汇转换成相应的数据让计算机进行识别，因此再解决三个问题之前，都需要将所需留言进行中文分词，再进行进一步的数据处理。

## 2.2 去除停用词及其余干扰因素

对于所要分析数据大体上都是不完整的，不一致的脏数据，无法直接进行数据挖掘。或挖掘效果差强人意。为了提高数据挖掘的质量，就在挖掘分析之前加入数据预处理技术。三个问题中的留言可能会受到无关紧要的数据影响，因此就要除去了各种符号，制表符，空格，回车等不必要停用词的干扰因素。还需要根据要处理的数据对停用词表进行相应的补充，此举用于去除杂糅及无用的语义，为后期模型的建立提高效率及准确程度

## 2.3 人工筛选分类

针对问题 3 要对回复构建一套完整评价方案的解决之前，先要将所提供的的附件 1 里的回复进行一遍人工筛选，通过阅读比较问题和答复之后，将完整规范的答复标记为 1，含糊一般的答复标记为 0，此举是为了后期模型构建所进行的准备。

## 2.4 建立基本评分公式

问题 3 需从相关性、完整性、可解释性三点来构建评价方案，为此我从相关性、完整性、可解释性三点中各取两项要素构建图 1 所示公式。再由公式所得出的评分结合模型预测的结果得出最终的评价。

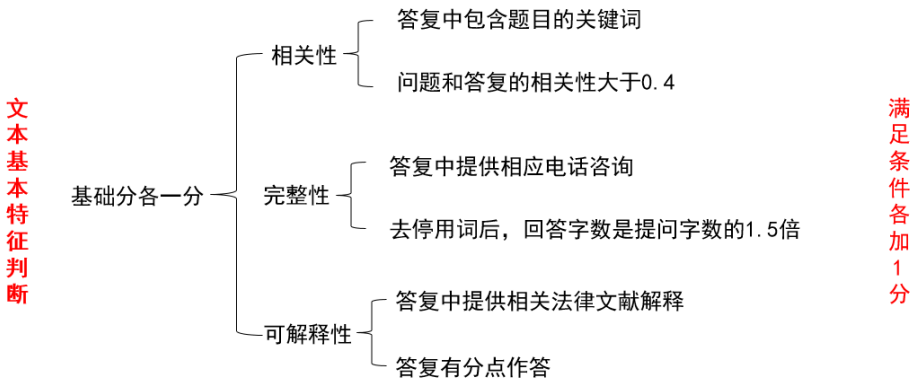



*图 1 问题 3 基本评分公式*

## 三、算法简介

### 3.1 TF-IDF

TF-IDF 模型是一类应用广泛的加权技术，经常被用来进行信息检索和数据挖掘。TF(Term Frequency)是词频的简称，可理解为文本内词汇出现的频率，逆文本频率的缩写为 IDF，即一个词语普遍关键性的度量。

TF-IDF 的核心思想为：若某短语(或词)于一篇文章内多次出现，即 TF 较高，同时甚少出现于其他文章内，那么判定该短语(或词)具备良好类别区分性能。其各自的计算公式如下：

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

图 2 TF 的计算公式

其中符号 ni,j 代表 ti 此词在 dj 此文件内出现的次数，那么上式分母表示 dj 内全部字词出现次数的合计值

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

图 3 IDF 的计算公式

|D|代表语料库内文件总量；|{j:ti∈dj}|代表包含 ti 此词语的文件数量(也可理解为 ni,不等于零的文件数量)。若此词语 ti 未在语料库内，就会出现除数等于零的结果，所以，通常使用 1+|{j:ti∈dj}|。

TF-IDF 的值则是这两个值的乘积值：

$$tfidf_{i,j} = tf_{i,j} \times idf_i$$

图 4 TF-IDF 的计算公式

### 3.2 TextRank

TextRank 是一种基于图的无监督关键词抽取方法。该方法将文档构建为图 G= (V, E) ，其中 V 为节点集，E 为采用共现关系构造任意两点之间的边。当两个节点在同一句子中共现，则两个节点之间存在边。根据下图公式，迭代传播各节点的权重，直至收敛。接着对节点权重进行倒序排序，选择排序靠前的词作为文本关键词。节点 WS (Vi) 的值计算方法为

$$WS(V_i)=(1-\lambda)+\lambda\times\sum_{V_j\in In(V_i)}\frac{w_{ji}}{\sum_{V_k\in Out(V_j)}w_{jk}}WS(V_j)$$

图 5 WS（Vi）的计算公式

其中，对于任意一个节点 Vi，为得到节点权重 WS (Vi) ，需对其进行迭代传播计算；In (Vi) 表示 Vi 的入度，即指向点 Vi 的节点集；Vj 表示指向 Vi 节点集中的某一节点，wji 表示节点 Vi 和 Vj 间的边权重。对于点 Vj, Out (Vj) 表示 Vj 的出度,即 Vj 指向的节点集合。Vk 表示 Vj 指向的节点集合的某一节点；wjk 表示节点 Vj 和 Vk 间的边权重。WS (Vj) 为节点 Vj 的节点权重。 λ 为阻尼系数，取值范围为 0 到 1。

### 3.3 JIEBA 分词

Jieba 分词采用基于前缀词典实现的高效词图扫描，生成句中汉字所有可能成词情况所构成的有向无环图（DAG），同时采用动态规划查找最大概率路径，找出基于词频的最大切分组合，对未登录词，采用基于汉字成词能力的 HMM 模型，使得能更好的实现中文分词效果。

### 3.4 MultinomialNB

朴素贝叶斯分类器是一种有监督学习，有多项式、伯努利、高斯三种常见模型，其中多项式朴素贝叶斯比较适用于离散值模型，如文本分类。对于文本分类模型不仅需要看词语是否出现在文本中，同时还需要看出现频次。

$$P(C=c)=\frac{\text{属于类}c\text{的文档数}}{\text{训练集文档总数}}$$

图 6 先验概率如公式

$$P(w_i|c)=\frac{\text{词}w_i\text{在属于类}c\text{的所有文档中出现次数}}{\text{属于类}c\text{的所有文档中的词语总数}}$$

图 7 条件概率如公式：（P(wi|c)表示词 wi 在类别 c 的文档中的权重）

先验概率和条件概率的计算均使用了最大似然估计，计算出的是相对频率值，使训练数据出现的概率最大。

$$P(w_i|c)=\frac{\text{词}w_i\text{在属于类}c\text{的所有文档中出现次数}+1}{\text{属于类}c\text{的所有文档中的词语总数}}$$

图 8 拉普拉斯平滑如公式

$$\arg\max_{c\in C}[\log P(c)+\sum_{i=1}^{n}\log P(w_i|c)]$$

图 9 预测结果

### 3.5 GaussianNB

高斯贝叶斯适合使用在各个属性在各个类别中均服从高斯分布，便使用极大似然估计得到高斯分布的参数——均值和方差，之后使用概率密度来得到某样本属于各个类别的“概率”

中心极限定理是高斯分布相关的一个重要定理，该定理认为当分布样本足够大时，其渐进分布是高斯分布．高斯分布的密度函数 f(x）如下图所示。

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

*图 10 高斯分布函数*

在文本分类中，上述公式的数学期望值 μ 等于样本数据集中特征值的均值，σ 为特征值在各个样本等概率条件下的方差．当高斯分布的概率密度函数参数确定后，根据文本的特征值计算出属于某种类别的概率，然后将该样本分到概率最高的类别中去

任意 d 文本，假设每个特征之间相互独立且各类有标记的训练样本个数相同，预测它属于文本类 C∈{c1,c2,...,cd}中的某一类 ck，计算方法如下图所示。

$$P(c_k \mid d)=\frac{p(c_k)p(d \mid c_k)}{p(d)}$$

*图 11 概率计算公式*

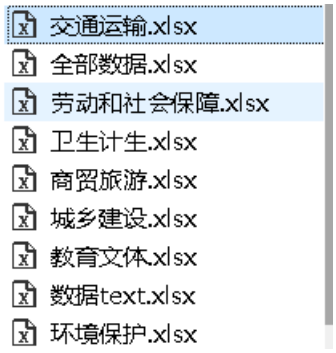
在分类中通常把上公式等价于 $p(dc_k)=\prod^{n}_{i=1}f_i(x_i)$，求取属于所有类中 $p(dc_k)$ 的概率并把该 d 文本预测为概率最大的类别

## 四、程序设计

### 4.1 群众留言分类

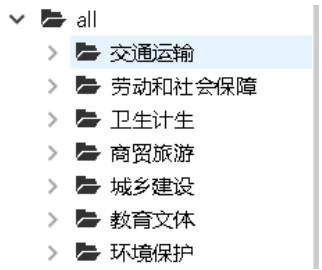
首先对文本进行处理：按全部数据中的 1 类标签把表格拆分为 8 个表格

```python
#第一步：调用pandas包
import pandas as pd
#第二步：读入文件
iris = pd.read_excel('./全部数据.xlsx')
# 第三步：获取class列表并去重
class_list = list(iris['一级标签'].drop_duplicates())
#第四步：按照类别分文件存放数据
for i in class_list:
    iris1 = iris[iris['一级标签']==i]
    iris1.to_excel('./%s.xlsx'%(i))
```

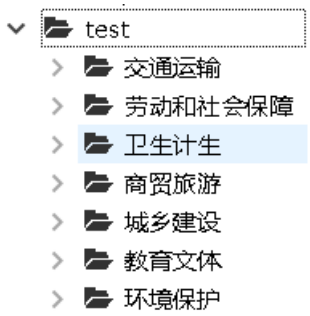



再把每个表格里留言转换成 txt 并存入相应的文件夹里

```python
import xlrd

import pandas as pd

def txt_create(name,msg): #txt的名，内容
    desktop_path = "./all/环境保护/"
    full_path = desktop_path + name + '.txt'
    file = open(full_path, 'w')
    file.write(str(msg).encode('GBK','ignore').decode('GBK'))


data = xlrd.open_workbook('环境保护.xlsx')
st = data.sheets()[0]
rows = st.nrows

for i in range(rows):
    string = str(i)
    txt_create(str(i),st.row_values(i))
```




这里使用第一批的数据充当测试集，用全部数据充当训练集, 文本预处理同上




对全部数据进行分词并导出分词结果

```
def segText(inputPath, resultPath):
    fatherLists = os.listdir(inputPath)  # 主目录
    for eachDir in fatherLists:  # 遍历主目录中各个文件夹
        eachPath = inputPath + eachDir + "/"  # 保存主目录中每个文件夹目录，便于遍历二级文件
        each_resultPath = resultPath + eachDir + "/"  # 分词结果文件存入的目录
        if not os.path.exists(each_resultPath):
            os.makedirs(each_resultPath)
        childLists = os.listdir(eachPath)  # 获取每个文件夹中的各个文件
        for eachFile in childLists:  # 遍历每个文件夹中的子文件
            eachPathFile = eachPath + eachFile  # 获得每个文件路径
            content = readFile(eachPathFile)  # 调用上面函数读取内容
            result = (str(content)).replace("\r\n", "").strip()  # 删除多余空行与空格

            cutResult = jieba.cut(result)  # 默认方式分词，分词结果用空格隔开
            saveFile(each_resultPath + eachFile, " ".join(cutResult))  # 保存分词好的文件
```

对分词后的数据进行向量化

```
def bunchSave(inputFile, outputFile):
    # 分类的类别列表就是文件夹的名字
    catelist = os.listdir(inputFile)
    # 创建一个bunch实例
    # 类别名，类别名，文件的完整路径，文件内容
    bunch = Bunch(target_name=[], label=[], filenames=[], contents=[])
    # 向bunchd的target_name添加数据，extand是python list的一个添加多个元素的方法
    bunch.target_name.extend(catelist)
    #获取每个目录下的全部文件
    for eachDir in catelist:
        eachPath = inputFile + eachDir + "/" # 拼出分类子目录的路径
        fileList = os.listdir(eachPath) # 获取每个类别下的全部txt文件
        for eachFile in fileList:
            # 完整的每个文件路径
            fullName = eachPath + eachFile  # 二级目录子文件全路径
            bunch.label.append(eachDir)  # 当前分类标签
            bunch.filenames.append(fullName)  # 保存当前文件的路径
            bunch.contents.append(readFile(fullName).strip())  # 保存文件词向量
    with open(outputFile, 'wb') as file_obj:  # 持久化必须用二进制访问模式打开
        pickle.dump(bunch, file_obj)
```

获取停词表

```
def getStopWord(inputFile):
    stopWordList = readFile(inputFile).splitlines()
    return stopWordList
```

计算 TF-IDF

```
def getTFIDFMat(inputPath,
                stopWordList,
                outputPath,
                tftfidfspace_path,
                tfidfspace_arr_path,
                tfidfspace_vocabulary_path):  # 构建TF-IDF向量
    bunch = readBunch(inputPath)
    tfidfspace = Bunch(target_name=bunch.target_name, label=bunch.label, filenames=bunch.filenames, tdm=[],
                       vocabulary={})
    '''读取tfidfspace'''
    tfidfspace_out = str(tfidfspace)
    saveFile(tftfidfspace_path, tfidfspace_out)

    vectorizer = TfidfVectorizer(stop_words=stopWordList, sublinear_tf=True, max_df=0.5)

    tfidfspace.tdm = vectorizer.fit_transform(bunch.contents)
    tfidfspace_arr = str(vectorizer.fit_transform(bunch.contents))
    saveFile(tfidfspace_arr_path, tfidfspace_arr)
    tfidfspace.vocabulary = vectorizer.vocabulary_  # 获取词汇
    tfidfspace_vocabulary = str(vectorizer.vocabulary_)
    saveFile(tfidfspace_vocabulary_path, tfidfspace_vocabulary)
    '''over'''
    writeBunch(outputPath, tfidfspace)
```

对测试集进行同上的操作，然后构建出测试集的 TF-IDF 向量

```
def getTFIDFMat(inputPath,
                stopWordList,
                outputPath,
                tftfidfspace_path,
                tfidfspace_arr_path,
                tfidfspace_vocabulary_path):  # 构建TF-IDF向量
    bunch = readBunch(inputPath)
    tfidfspace = Bunch(target_name=bunch.target_name, label=bunch.label, filenames=bunch.filenames, tdm=[],
                       vocabulary={})
    '''读取tfidfspace'''
    tfidfspace_out = str(tfidfspace)
    saveFile(tftfidfspace_path, tfidfspace_out)

    vectorizer = TfidfVectorizer(stop_words=stopWordList, sublinear_tf=True, max_df=0.5)

    tfidfspace.tdm = vectorizer.fit_transform(bunch.contents)
    tfidfspace_arr = str(vectorizer.fit_transform(bunch.contents))
    saveFile(tfidfspace_arr_path, tfidfspace_arr)
    tfidfspace.vocabulary = vectorizer.vocabulary_  # 获取词汇
    tfidfspace_vocabulary = str(vectorizer.vocabulary_)
    saveFile(tfidfspace_vocabulary_path, tfidfspace_vocabulary)
    '''over'''
    writeBunch(outputPath, tfidfspace)
```

进行分类

```
'''处理结束'''
predicted = clf.predict(testSet.tdm)
total = len(predicted)
rate = 0
for flabel, fileName, expct_cate in zip(testSet.label, testSet.filenames, predicted):
    if flabel == expct_cate:
        print(fileName, ":实际类别：", flabel, "-->预测类别：", expct_cate, "预测正确\n")
    else:
        rate += 1
        print(fileName, ":实际类别：", flabel, "-->预测类别：", expct_cate, "预测错误\n")
print("准确率：",(1-float(rate)/ float(total)))
```

处理结果如下：

```
./split/test_split/环境保护/123.txt :实际类别： 环境保护 -->预测类别： 环境保护 预测正确
./split/test_split/环境保护/124.txt :实际类别： 环境保护 -->预测类别： 环境保护 预测正确
./split/test_split/环境保护/125.txt :实际类别： 环境保护 -->预测类别： 环境保护 预测正确
./split/test_split/环境保护/126.txt :实际类别： 环境保护 -->预测类别： 环境保护 预测正确
./split/test_split/环境保护/127.txt :实际类别： 环境保护 -->预测类别： 环境保护 预测正确
./split/test_split/环境保护/128.txt :实际类别： 环境保护 -->预测类别： 环境保护 预测正确
./split/test_split/环境保护/129.txt :实际类别： 环境保护 -->预测类别： 环境保护 预测正确
./split/test_split/环境保护/130.txt :实际类别： 环境保护 -->预测类别： 环境保护 预测正确
准确率： 0.9953051643192489
```

## 4.2 热点问题挖掘

首先要对原始数据进行处理。通过 jieba 分词将附件 3 里的留言详情进行分词和去停顿词处理。

其次将分词处理后的数据用 TF-IDF 进行向量化。将向量化后的数据进行余弦相似度处理，将相识度大于 0.4 的分为同一类别，为了防止一条信息被分为多类，将已经分类的置零处理，判断时去除全部为零的行，将得到的数据按要存储下来可得到“热点问题留言明细表”，如下图所示：

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | 问题ID | 留言编号 | 留言用户 | 留言时间 | 留言详情 | 点赞数 | 反对数 |
| 2 | 1 | 188006 | A0001029 | 2019/02/2 | 座落在A市A3区联丰路米兰春天G2栋320，一家 | 0 | 0 |
| 3 | 2 | 188007 | A0007479 | 2019/02/1 | A市A6区道路命名规划已经初步成果公示文件， | 1 | 0 |
| 4 | 3 | 188031 | A0004006 | 2019/07/1 | 本人系春华镇金鼎村七里组村民，不知是否有柜 | 1 | 0 |
| 5 | 4 | 188039 | A0008137 | 2019/08/1 | 靠近黄兴路步行街，城南路街道、大古道巷、一 | 1 | 0 |
| 6 | 5 | 188059 | A0002857 | 2019/11/2 | A市A3区中海国际社区三期四期中间，即蓝天璞 | 0 | 0 |
| 7 | 5 | 205478 | A909124 | 2019/12/2 | 作为一个上班族，每天白天很忙，甚至于晚上还 | 0 | 0 |
| 8 | 5 | 233538 | A0008222 | 2019/04/2 | 尊敬的领导，我住A4区天健一平方英里小区。2 | 0 | 0 |
| 9 | 5 | 288837 | A0009791 | 2019/11/1 | 我们是A市A2区新开铺街道福满新城二期工程周 | 0 | 0 |
| 10 | 6 | 188073 | A909164 | 2019/03/1 | 作为麓泉社区麓谷明珠小区6栋居民，我们近期 | 0 | 0 |
| 11 | 7 | 188074 | A909092 | 2019/01/3 | `二高一部”发出关于针对非法集资的打击的通 | 0 | 0 |
| 12 | 7 | 196463 | A0004314 | 2019/02/1 | 西地省三供一业移交政策大限已到，西地省开关 | 0 | 0 |
| 13 | 7 | 206296 | A0008952 | 2019/02/2 | 一直以来，富绿新村业主投诉不断，相关部门麻 | 0 | 0 |
| 14 | 8 | 188119 | A0003502 | 2019/05/2 | 我是一名在A市某地铁站上班的安检员，我是由 | 0 | 0 |
| 15 | 9 | 188170 | A8801132 | 2019/12/2 | 12月21日下午17时52分许，6路公交车（司机座 | 0 | 0 |
| 16 | 10 | 188249 | A0008408 | 2019/09/1 | 保利麓谷林语桐梓坡路与麓松路交汇处地铁凌晨 | 0 | 0 |
| 17 | 10 | 211202 | A0009976 | 2019/04/0 | 门口修地铁凌晨2、3点还在修，声音巨大！！家 | 0 | 0 |
| 18 | 10 | 261741 | A0006245 | 2019/09/1 | 地铁六号线至施工以来，一直施工到凌晨三点多 | 0 | 0 |
| 19 | 11 | 188251 | A0001309 | 2019/10/1 | 近来，下午晚高峰五点半左右，经过特立路与东 | 0 | 0 |
| 20 | 11 | 257881 | A0001309 | 2019/04/1 | 近期经常上下班开车从望仙路、特立路等经过， | 0 | 0 |
| 21 | 12 | 188260 | A0005348 | 2019/05/3 | 还我宁静我要复习迎考，大半年底商空调/冰柜外 | 0 | 0 |
| 22 | 13 | 188396 | A0004758 | 2019/04/1 | 桐梓坡589号白鹤咀停车场，由聚美龙楚新能源 | 1 | 2 |
| 23 | 13 | 198874 | A0004758 | 2019/04/1 | 桐梓坡589号白鹤咀停车场，由聚美龙楚新能源 | 0 | 0 |
| 24 | 13 | 204710 | A0004758 | 2019/10/0 | A3区一小的一面紧临地铁6号线项目施工，仅仅 | 0 | 0 |
| 25 | 13 | 273805 | A0004758 | 2019/04/1 | 桐梓坡589号白鹤咀停车场，由聚美龙楚新能源 | 1 | 0 |
| 26 | 13 | 283371 | A0004758 | 2019/10/1 | 尊敬的领导：向您报告A3区第一小学门口违规 | 0 | 0 |
| 27 | 14 | 188399 | A0009793 | 2019/07/0 | 您好，我想举报 A市利保壹号公馆项目 夜间噪 | 0 | 0 |
| 28 | 14 | 214888 | A0009583 | 2019/07/1 | 位于A3区学士路卓越千山里工地日夜施工扰民 | 0 | 0 |

*图 12 热点明细表部分截图*

在第二步的基础上进行热点问题的排名，热点问题指标是按照：相识热点数+点赞总数-反对总数得到，时间范围将相识类型的最大时间和最小时间得到，地点或人群在留言主中提取出来，将得到的数据按要存储下来可得到“热点问题表”，如下图所示：

| 热度排名 | 问题ID | 热度指数 | 时间范围 | 地点/人群 | 问题描述 |
|---|---|---|---|---|---|
| 1 | 1 | 2098 | 2019/08/19 | A市A5区K | A市A5区汇金路五矿万境K9县存在一系列问题 |
| 2 | 2 | 1759 | 2019/04/11 | A市 | 反映A市金毛湾配套入学的问题 |
| 3 | 3 | 1634 | 2019/01/08 | A4区 | 请问A4区公安派出所对58车货一案办案的进度如何了 |
| 4 | 4 | 739 | 2019/03/01 | A市 | 承办A市58车贷案警官应跟进关注留言 |
| 5 | 5 | 709 | 2019/04/18 | A4区小区 | A4区绿地海外滩小区距渝长厦高铁太近了 |

*图 13 热点问题表部分截图*

### 4.3 答复意见的评价

从附件 4 的留言中，可以了解到优秀的答复一般包含如下要素：切合问题的解答、有针对性的分点解释、引用法律文献或者公约条款进行说明、提供相关的电话咨询。针对这几点便可构建一套基本的评分公式。其中完整性、相关性和可解释性的基础分各一分，得分标准具体分为六点，第一点判断在去除停用词后，回答的字数是否大于等于问题字数的一点五倍，若超过，则完整性加一分。第二点判断回答中是否包含电话咨询，若包含，则完整性加一分。第三点判断回答中是否包含标题的关键词，此处需要将标题筛选出并应用 TextRank 算法提取关键词，若回答全部包含了问题的关键词，则相关性加一分。第四点则需要用 TF-IDF 与余弦相似性来计算回答与问题的相似度，倘若相似度大于 0.4，则相关性加一分。第五点判断回答中是否包含文献的解释，第六点判断是否分点回答，若满足条件，则可解释性加一分。公式构建完毕后，将数据进行先一步的评分，得到评分后的结果进行下一步的操作。

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 留言编号 | 留言用户 | 留言主题 | 留言时间 | 留言详情 | 答复意见 | 答复时间 | 相关性 | 完整性 | 可解释性 |
| 0 | 0 | 2549 | A0004558 | A2区景蓉 | 2019/4/2 | 2019年4月 | 现将网友 | 2019/5/1 | 3 | 1 | 1 |
| 1 | 1 | 2554 | A0002358 | A3区潇楚 | 2019/4/2 | 潇楚南路 | 网友“A0 | 2019/5/9 | 2 | 2 | 1 |
| 2 | 2 | 2555 | A0003161 | 请加快提 | 2019/4/2 | 地处省会 | 市民同志 | 2019/5/9 | 2 | 2 | 3 |
| 3 | 3 | 2557 | A0001107 | 在A市买公 | 2019/4/2 | 尊敬的书 | 网友“A0 | 2019/5/9 | 1 | 3 | 1 |
| 4 | 4 | 2574 | A0009233 | 关于A市公 | 2019/4/2 | 建议将“ | 网友“A0 | 2019/5/9 | 2 | 2 | 1 |
| 5 | 5 | 2759 | A0007753 | A3区含浦 | ######## | 欢迎领导 | 网友“A0 | 2019/5/9 | 2 | 3 | 1 |
| 6 | 6 | 2849 | A0001008 | A3区教师 | 2019/3/2 | 尊敬的胡 | 网友“A0 | 2019/5/9 | 2 | 2 | 2 |
| 7 | 7 | 3681 | UU00812 | 反映A5区 | 2018/12/ | 我做为一 | 网友“UU | 2019/1/2 | 2 | 2 | 3 |
| 8 | 8 | 3683 | UU008792 | 反映A市美 | 2018/12/ | 我是美麓 | 网友“UU | 2019/1/1 | 2 | 3 | 1 |
| 9 | 9 | 3684 | UU008687 | 反映A市洋 | 2018/12/ | 胡书记好 | 网友“UU | 2019/1/1 | 2 | 1 | 1 |
| 10 | 10 | 3685 | UU008220 | 反映A2区 | 2018/12/ | 我家住在 | 网友“UU | 2019/3/1 | 1 | 2 | 1 |
| 11 | 11 | 3692 | UU008829 | A5区鄱阳 | 2018/12/ | 胡书记： | 网友“UU | 2019/1/2 | 2 | 2 | 1 |
| 12 | 12 | 3700 | UU00877 | A4区万国 | 2018/12/ | 尊敬的书 | 网友“UU | 2019/1/1 | 1 | 1 | 1 |
| 13 | 13 | 3704 | UU008148 | 举报A市芦 | 2018/12/ | 尊敬的领 | 网友“UU | 2019/1/3 | 1 | 1 | 1 |
| 14 | 14 | 3713 | UU008122 | 建议增开 | 2018/12/ | 建议增开 | 网友“UU | 2019/1/1 | 2 | 2 | 1 |
| 15 | 15 | 3720 | UU008444 | 关于A市新 | 2018/12/ | 2016年下 | 网友“UU | 2019/3/6 | 2 | 2 | 1 |
| 16 | 16 | 3727 | UU008119 | 投诉A3区 | 2018/12/ | 12月16日 | 网友“UU | 2019/1/3 | 1 | 2 | 1 |
| 17 | 17 | 3733 | UU008706 | 建议在A市 | 2018/12/ | 梅溪湖至 | 网友“UU | 2019/1/1 | 2 | 2 | 1 |
| 18 | 18 | 3747 | UU008201 | 希望相关 | 2018/12/ | 希望相关 | 网友“UU | 2019/1/8 | 1 | 2 | 1 |
| 19 | 19 | 3755 | UU008168 | 希望A市社 | 2018/12/ | 看病需要 | 网友“UU | 2019/1/4 | 1 | 1 | 1 |
| 20 | 20 | 3756 | UU008168 | 希望A市潇 | 2018/12/ | 希望潇楚 | 网友“UU | 2019/1/4 | 1 | 3 | 1 |
| 21 | 21 | 3760 | UU008150 | 反映A9市 | 2018/12/ | A9市北盛 | 网友“UU | 2019/1/8 | 2 | 2 | 1 |
| 22 | 22 | 3762 | UU008105 | 呼吁A5区 | 2018/12/ | 尊敬的市 | 网友“UU | 2019/1/1 | 2 | 1 | 2 |
| 23 | 23 | 3777 | UU008162 | 关于A市地 | 2018/12/ | A市委市政 | 网友“UU | 2019/1/2 | 1 | 1 | 1 |
| 24 | 24 | 3788 | UU008160 | 咨询A市商 | 2018/12/ | 深圳市缴 | 网友“UU | 2019/1/3 | 2 | 2 | 1 |

图 13 留言评分截图

在对数据进行基本的处理和分类后，便要创建模型进行机器学习。首先将训练集样本转化为词频向量，再将词频向量转化为 TF-IDF 权重向量。构建高斯朴素贝叶斯模型，将 TF-IDF 权重向量与对应标签进行模型的训练。训练完毕后，便可将所需数据进行预测，得要结果存入表中，进行最后的统合。

在对所有的数据评分完毕及完成模型预测之后，便结合两者结果对答复进行一个模糊综合评价。在经过多次实验和评价标准调整后，整理出如下图所示判断标准。

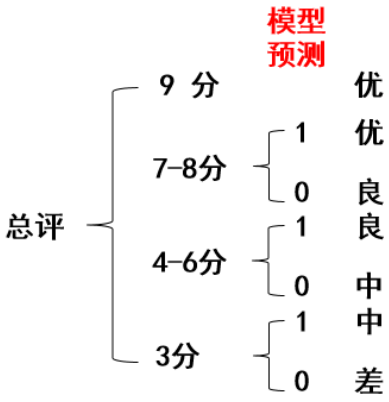



图 14 判断标准

在进行测试的 2815 条数据中，优良中差的比率约为 2:9:12:5，评价的结果大部分符合人为判断。

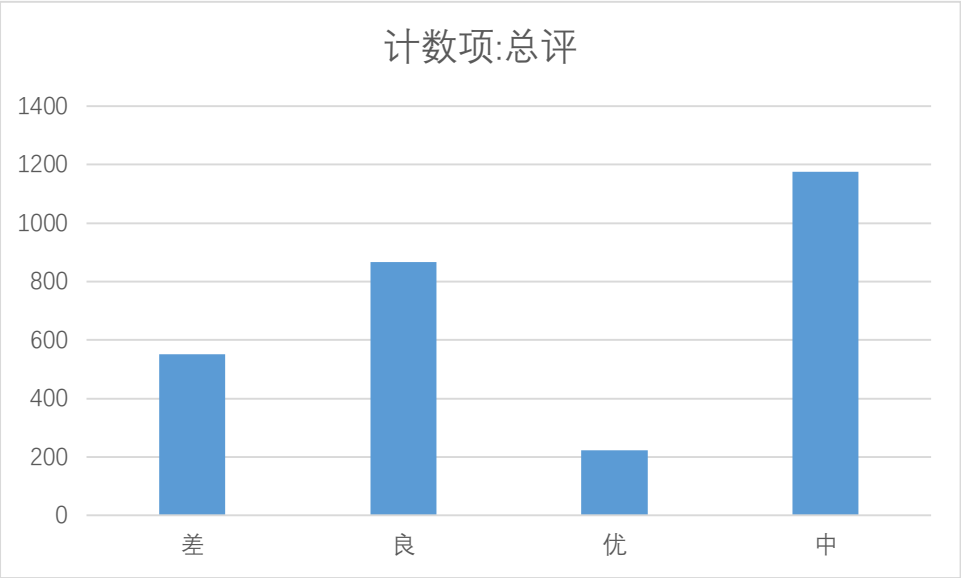



*图 15 测试数据评价结果柱状图*

## 五、参考文献

[1]王习涛 马雁疆 刘新新. 基于余弦相识度的聚类算法在统计调查对象分类中的应用研[J]. 市场研究 2019,(05),15-20

[2]黄春梅 王松磊. 基于词袋模型和 TF-IDF 的短文本分类研究[J]. 软件工程 2020,23(03),1-3

[3]石凤贵.基于 TF-IDF 中文文本分类实现.马鞍山师范高等专科学校[J].现代计算机 2020,(06),51-54+75

[4]苏慧婧 群诺 贾宏云 基于 GaussianNB 模型的藏文文本分类研究与实现[J].青海师范大学学报(自然科学版) 2019,04

[5]张颖怡 章成志 陈果 基于关键词的学术文本聚类集成研究[J]. 情报学报 2019,08