基于“智慧政务”中的文本挖掘应用

摘要

现在，随着人们的生活水平不断提高，网络不断发达。网络政务也进入了人们的生活。网络政务平台就此诞生。对于网络问政平台来说，随着百姓的生活水平提高，百姓的需求也不断增加，同时也产生很多问题，因此网上问政平台会有大量的群众的意见、建议或者问题留言，每一条问题留言都需要相关部门去处理。在之前每一条留言都是由人工处理，工作人员先将这些留言进行分类，然后交给相关部门对具体问题进行处理然后做出相应霍夫，这不仅花费大量的人力物力，而且效率并不高。随着大数据、云计算、人工智能等技术的发展，希望能借助自然语言处理技术来建立一个模型将这些留言进行分类，减轻工作人员的负担。

本队伍用 python 软件对网络留言进行分析处理即自然语言处理，使用文本处理包括分词去重去停用词、文本分类、文本聚类、二维表的处理等知识对所提的问题进行处理。

关键词：文本分类、文本聚类、机器学习

目录

1、群众留言分类

1.1 问题描述

在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系（参考附件 1 提供的内容分类三级标签体系）对留言进行分类，以便后续将群众留言分派至相应的职能部门处理。目前，大部分电子政务系统还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。请根据附件 2 给出的数据，建立关于留言内容的一级标签分类模型。通常使用 F-Score 对分类方法进行评价。

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i + R_i}$$

其中P_i为第 i 类的查准率，R_i为第 i 类的查全率。

1.2 解题思路

该问题属于自然语言处理中的文本分类问题，我们可以利用 python 对文本进行分类，参照文本分类的建模方法的一般步骤如下：

1）分析问题，了解数据。

2）抽取数据，将需要分类的文本数据导入 python。

3）数据清洗，对导入的数据做预处理，包括两个步骤：

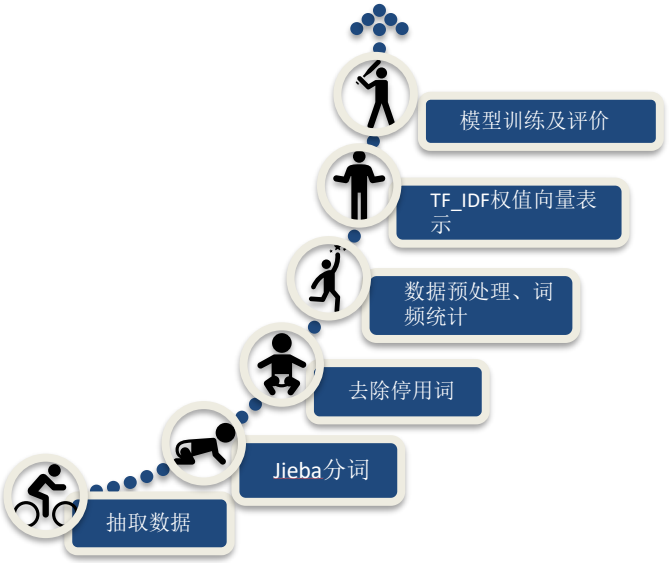
1、jieba 分词，对清洗后的数据进行分词，也就是将这些一长串的文本分割成一个一个的词。2、去除停用词，将一些无意义的字词和符号去掉，如：“的”、“在”、“什么”以及语气助词、各种符号等。

4）数据预处理，以及词频统计。

5）文本向量表示，将文本转化成向量的形式。获取训练集和测试集的 tf-idf 权值向量。

6）将模型训练以及评价模型。

在 python 的第三方库 sklearn 中,有前人已经制作好的分类模型，我们只需使用即可，而该模型实际是一个类似于机器学习模型，其通过模型训练不断学习记录，使之分类的判断越来越准确。因此，训练数据越多，模型准确率就越高。同时在数据清洗阶段、数据预处理阶段做的好不好，也会影响分类的准确率。



解题流程图

1.3 具体步骤

2 数据抽取

先对附件 2 进行分析，其图片如下，每一行都是一条留言，最后有该留言的标签，因考虑到主题的内容过少，特征会少会引起歧义，从而影响最后的判断。所以我们取“留言详情”和“一级标签”的内容进行模型训练。

	留言编号	留言用户	留言主题	留言时间	留言详情	一级标签
2	24	A00074011	市西湖建筑集团占道施工有安全隐	2020/1/6 12:09:38	目施工围墙内。每天尤其上下班期	城乡建设
3	37	U0008473	水一方大厦人为烂尾多年，安全隐	2020/1/4 11:17:46	护栏围着，不但占用人行道路，而	城乡建设
4	83	A00063999	投诉A市A1区苑物业违规收停车费	2019/12/30 17:06:1	业主已多次向物业和社区提出诉求	城乡建设
5	303	U0007137	区蔡锷南路A2区华庭楼顶水箱长年不	2019/12/6 14:40:14	的用品，霉是一种强致癌物，我们	城乡建设
6	319	U0007137	A1区A2区华庭自来水好大一股霉味	2019/12/5 11:17:22	的用品，霉是一种强致癌物，我们	城乡建设

在 python 中 csv 格式比 xlsx 格式导入效率更好一些，因此我们先将 xlsx 转成 csv 格式。转化代码如下：

```
#-*-coding:utf-8-*-
import pandas as pd
data = pd.read_excel('附件4.xlsx','Sheet1',index_col=0)
data.to_csv('data4.csv',encoding='gb18030')
```

然后将附件 2.csv 导入 python 并取出“留言详情”和“一级标签”两列数据，并更改其列名称为 message 和 label，为了数据的公平性，在七个一级标签：城乡建设、卫生计生、环境保护、教育文体、交通运输、商贸旅游和劳动和社会保障中取出相同数量的数据进行模型训练和测试，由于各标签数量不一致，所以选用时以最少数量的标签为标准，附件 2 中最少的是交通运输 613 条，所以这里我们每个标签取 600 条来做模型训练和测试。导入和选取代码如下：

```
data = pd.read_csv('附件2.csv',encoding='gb18030') #encoding为设置数据编码
data = data.loc[:,['留言详情','一级标签']]         #导入数据只取留言详情和一级标签的内容放在data中
data.columns = ['message', 'label']               #将标签进行更换
n = 600                                           #设置各标签选取数量
a = data[data['label'] == '城乡建设'].sample(n)
b = data[data['label'] == '商贸旅游'].sample(n)
c = data[data['label'] == '卫生计生'].sample(n)
d = data[data['label'] == '交通运输'].sample(n)
e = data[data['label'] == '劳动和社会保障'].sample(n)
f = data[data['label'] == '教育文体'].sample(n)
g = data[data['label'] == '环境保护'].sample(n)    #选取7*600条数据
data_new = pd.concat([a,b,c,d,e,f,g],axis=0)      #将取出的数据进行拼接整合
data_dup = data_new['message'].drop_duplicates()  #对去除重复数据
```

整合后结果：

In [10]:
```
data_new
```

Out[10]:

	message	label
1798	\n\t\t\t\t\t\n\t\t\t\t\t I市碧桂园小区美岸三街102栋业主无视城...	城乡建设
2000	\n\t\t\t\t\t\n\t\t\t\t\t\t长株潭城铁于4月29日更改路线，车次已增加，...	城乡建设
538	\n\t\t\t\t\t\n\t\t\t\t\t\t尊敬的市长：您好，我叫冯生新，是一名退休职工...	城乡建设
182	\n\t\t\t\t\t\n\t\t\t\t\t 1、上次我提出了BoBo水上乐园的建议，...	城乡建设
710	\n\t\t\t\t\t\n\t\t\t\t\t\tF6县三医院扩建住院部，在没有保证周边居民房...	城乡建设
...	...	...
2603	\n\t\t\t\t\t\n\t\t\t\t\t 在文溪路白鹭驿附近有一家专门给宾馆布草消...	环境保护
2220	\n\t\t\t\t\t\n\t\t\t\t\t\t该沙石料厂，对周边老百姓生活构成了严重污染，...	环境保护
2696	\n\t\t\t\t\t\n\t\t\t\t\t A市A4区大明工业园永利纸制品加工厂排放...	环境保护
2853	\n\t\t\t\t\t\n\t\t\t\t\t\t蒋厅长： 你好，我是西地省B...	环境保护
2536	\n\t\t\t\t\t\n\t\t\t\t\t 我是来自于L10县县飞山乡高桥村公塘组的...	环境保护

4200 rows × 2 columns

去重复后结果：

In [7]:
```
data_dup
```

Out[7]:
```
1464    \n\t\t\t\t\t\t\n\t\t\t\t\t\t\tM2县征拆中心（原住建局管的）的县城中河棚户...
1245    \n\t\t\t\t\t\t\n\t\t\t\t\t\t\t    广西贺州、湖北、河南、河北陕西等地的市州...
890     \n\t\t\t\t\t\t\n\t\t\t\t\t\t\t近二年，J10 县城建基础设施停滞不前，八一...
926     \n\t\t\t\t\t\t\n\t\t\t\t\t\t\t    清泉镇的廉租房已经建设多年，为什么老百姓...
1938    \n\t\t\t\t\t\t\n\t\t\t\t\t\t\t    我是新安镇一居民，我们镇上的自来水太不卫...
                                  ...
2392    \n\t\t\t\t\t\t\n\t\t\t\t\t\t\t楚粤路仔健烧烤店后厨位置，从早到晚浓烟滚滚，...
2166    \n\t\t\t\t\t\t\n\t\t\t\t\t\t\t尊敬的政府领导您好，我坐落在友爱村，这里建一...
2795    \n\t\t\t\t\t\t\n\t\t\t\t\t\t\t    位于K市K9县新圩镇龙家坊村与上清涵村交...
2217    \n\t\t\t\t\t\t\n\t\t\t\t\t\t\tE8县金桥建材砖厂自投资建厂以来一直存在着这...
2230    \n\t\t\t\t\t\t\n\t\t\t\t\t\t\t    石茂盛向市环保局举报E4县界岭加油站油库...
Name: message, Length: 4154, dtype: object
```

剩下 4154 条，则去除了 46 条重复留言信息。

2 1　　分词

Jieba 是一个 python 第三方库，它是常用的文本处理工具。其里面包含很多常用的词汇，同时也包含了一些数据处理函数。而我们 jieba 分词时，就是利用 jieba 库中已有词典，然后

去一个一个识别我们的文本，当识别到词汇时将其用空格分割出来。例如：

文本：我是中国人，我爱我的祖国。

处理方法：首先识别“我”接着再识别“我是”，很明显“我是”不是一个词，所以将“我”打断，第一步的结果为：我 是中国人，我爱我的祖国。

接着识别“是”，再识别“是中”、“是中国”这些都不是词，所以把“是”打断，结果为：我 是 中国人，我爱我的祖国。

识别“中国”的时候，发现是一个词，便将其打断。根据该方法处理结果为：我 是 中 国 人 ， 我 爱 我 的 祖国 。

Jieba 分词有三种模式，全模式、精确模式和搜索引擎模式，分词效果不同，其分出来的结果可能稍有不同。同时对于里面没有的新词汇，我们也可以往里添加。我们需要分词的代码如下：

```
data_cut = data_dup.apply(lambda x: jieba.lcut(x))
```

data_dup 是经过上一步去重后的结果，lcut(x)这里是将文本进行分词然后以列表形式输出分词结果：

```
In  [6]: data_cut

Out[6]: 1464    [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        1245    [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        890     [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        926     [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        1938    [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
                                    ...
        2392    [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        2166    [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        2795    [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        2217    [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        2230    [\n, \t, \t, \t, \t, \t, \n, \t, \t, \t, \t, \...
        Name: message, Length: 4154, dtype: object
```

2 2 去除停用词

为了提高分类精度，我们需要去除停用词。所谓“停用词”就是没有用的词，比如介词、人称代词等，还有就是该词在该领域或该模型中没有用，就需要去除。例如：在该模型中“领导”、“市长”、“县长”这样的词汇哪一类都有可能出现，他不特定的代表哪一类，因此它是需要去掉的。停用词表可以存放在 txt 文件中，每一行只能有一个词，我们可以写代码时调用即可，同时我们也可以在写代码时直接添加。各领域有各领域的停用词表，网络上都可以找的到。这里我们使用通用的停用词表。并添加一些自定义词汇，例如：政府 解决 日 月 时间 县 书记 领导 省 区 尊敬 单位 相关 真的 想 希望 一座 一个 一些 一样 举报 市政府 近日 近来 谁 提 反映 内容 有关 部门等。

详细见附件 stopword.txt。

处理代码：

```
stoplist = pd.read_csv('stopword.txt',encoding='gbk',sep='hahaha',header=None)
stoplist = [' ','\n\r','\n','\t','\u3000']+list(stoplist.iloc[:,0])
data_after_stop = data_cut.apply(lambda x: [i for i in x if i not in stoplist])
```

首先读取 stopword.txt 文件内容，并以 hahaha 进行连接，这里最好不要用“，”、“。”和空格连接，因为这样会对处理识别时造成歧义，从而造成去除不准确。观察文本信息，我

们可以看到它有\t\n\u 等等编码字符。这些也需要去除，所以我们可以另外用代码添加。第三行为去除处理，对 data_cut（分词后文本）中的词如果不在 stoplist（停用词表）中，则将其取出放在 data_after_stop 中。

```
labels = data_new.loc[data_after_stop.index, 'label']
adata = data_after_stop.apply(lambda x: ' '.join(x))
```

将每一条处理后的留言详情文本对应的标签存入 labels 中备用。同时处理后文本用空格连接起来也存入 adata 进行备用。

```
In [4]: data_after_stop
Out[4]: 1464    [M2, 县征, 拆, 中心, 原住, 建局, 县城, 河, 棚户, 改造, 项目, 工程,...
        1245    [广西, 贺州, 湖北, 河南, 河北, 陕西, 市州, 提取, 公积金, 装修, 新房, ...
        890     [二年, J10, 县城, 建, 基础设施, 停滞不前, 八一, 东路, 延伸段, 剩余, ...
        926                [清泉, 镇, 廉租房, 建设, 多年, 老百姓, 租, 医院, 学校, 公平]
        1938    [新安镇, 居民, 镇上, 自来水, 太, 卫生, 一到, 下雨天, 泥, 水, 浑水, 水...
                                        ...
        2392    [楚粤路, 仔健, 烧烤店, 后厨, 位置, 浓烟滚滚, 御景华宸, 小区, 住户, 深受其...
        2166    [坐落, 友爱, 村, 建, 混凝土, 场地, 污水, 乱流, 污染环境, 砖厂, 污水, ...
        2795    [位于, K, K9, 新圩镇, 龙家坊, 村, 清涵, 村, 交界, 地方, 几个, 非法...
        2217    [E8, 县金桥, 建材, 砖厂, 投资, 建厂, 规范, 先建后, 报批, 强行, 征地,...
        2230    [石, 茂盛, 向市, 环保局, E4, 界岭, 加油站, 油库, 漏油, 污染, 水源, ...
        Name: message, Length: 4154, dtype: object

In [5]: adata
Out[5]: 1464    M2 县征 拆 中心 原住 建局 县城 河 棚户 改造 项目 工程 潜心 运作 招标 阶段 ...
        1245    广西 贺州 湖北 河南 河北 陕西 市州 提取 公积金 装修 新房 K 住房 公积金 放开 ...
        890     二年 J10 县城 建 基础设施 停滞不前 八一 东路 延伸段 剩余 几百米 打通 七一 东...
        926                        清泉 镇 廉租房 建设 多年 老百姓 租 医院 学校 公平
        1938    新安镇 居民 镇上 自来水 太 卫生 一到 下雨天 泥 水 浑水 水人 长期 吃 行 百忙之...
                                        ...
        2392    楚粤路 仔健 烧烤店 后厨 位置 浓烟滚滚 御景华宸 小区 住户 深受其害 开窗 小区 出口...
        2166    坐落 友爱 村 建 混凝土 场地 污水 乱流 污染环境 砖厂 污水 乱流 收 红包 事 混凝...
        2795    位于 K K9 新圩镇 龙家坊 村 清涵 村 交界 地方 几个 非法 洗 矿厂 多年 没人管...
        2217    E8 县金桥 建材 砖厂 投资 建厂 规范 先建后 报批 强行 征地 环境污染 噪音 扰民 ...
        2230    石 茂盛 向市 环保局 E4 界岭 加油站 油库 漏油 污染 水源 一事 请问 邓 局长 事...
        Name: message, Length: 4154, dtype: object
```

我们可以观察这些结果，如果发现有无用词，我们可以在将其加入停用词表，重复刚才步骤去除停用词。这样使我们的分类更准确。

2 3 文本向量化表示：

我们知道机器没有思想，他是读不懂我们的文字的，所以我们要将文字转成机器能够读懂的形式。因此我们需要将文本向量化，使用独热编码方式将文本转化为词向量矩阵，如果出现该词那么该元素就是 1，否则为 0，如：

文本 1：我 是 中国 人 我 爱 我 的 祖国

文本 2：我 来自 北京

处理方法：先将全部词进行统计形成一个词向量，确保每一个词只出现一次。从上面我们可以统计为：[我 是 中国 人 爱 的 祖国 来自 北京]，对应成数字：[1 1 1 1 1 1 1 1 1]

根据该向量文本 1 可以写成：[1 1 1 1 1 1 1 0 0]

文本 2 为：[1 0 0 0 0 0 0 1 1]

这样的话会忽略词频信息，如文本 1“我”是出现了 3 次，这样也会对模型分类造成误差。所以我们需要将词频信息放进去，从而引进 TF-IDF 权重策略。

TF：关键词频，一篇文档中出现关键词的频率。

$$TF = \frac{N}{M}$$

N：单词在某文档出现的频次，M：该文档的单词数

IDF：逆向文本频率，用于衡量关键词权重的指数。

$$IDF = \log\left(\frac{D}{D_w}\right)$$

$$TF - IDF = TF \times IDF$$

D：总文档数，D_w：出现该单词的文档数。

我们先将文本划分为训练集和测试集，这里我们使用 0.2 的比例选取测试样本集。

```
data_tr, data_te, labels_tr, labels_te = train_test_split(adata, labels, test_size=0.2)#划分训练集和测试集
```

data_tr 为训练集样本，data_te 为测试集样本，labels 分别为对应的标签。

将测试集和训练集分别转化成相应的 TF-IDF 权值向量。并分别存放在 X_tr 和 X_te 里面。我们称该步骤为数据预处理。

```
countVectorizer = CountVectorizer()
data_tr = countVectorizer.fit_transform(data_tr)    #文本形式的训练样本转化成向量
X_tr = TfidfTransformer().fit_transform(data_tr.toarray()).toarray()  #训练集转化成TF-IDF权值向量

data_te = CountVectorizer(vocabulary=countVectorizer.vocabulary_).fit_transform(data_te) #测试集转成向量形式
X_te = TfidfTransformer().fit_transform(data_te.toarray()).toarray() #测试集转成TF-IDF权值向量
```

2 4 模型训练、测试以及评价

我们使用朴素贝叶斯算法模型（GaussianNB()）进行分类。这些算法都在 sklearn 中，只需导入使用即可。

首先将预处理后的样本数据导入到模型中，然后放入测试样本进行测试，该阶段也就是学习阶段。

将测试样本放入模型中进行数据预测，最后模型可以通过预测结果对比等操作输出其准确率以及模型评价的 F-Score 值。相关代码如下：

```
model = GaussianNB()  #导入朴素贝叶斯分类模型
model.fit(X_tr, labels_tr)  #模型训练
y = model.predict(X_te) #模型测试
print("准确率：",model.score(X_te, labels_te))  #输出准确率
print("其他指标:",classification_report(labels_te, y, target_names=None))  #输出各标签分类的指标
```

```
准确率： 0.7027677496991577
其他指标:               precision    recall  f1-score   support

        交通运输       0.74      0.62      0.68       113
     劳动和社会保障       0.71      0.65      0.68       119
        卫生计生       0.80      0.78      0.79       116
        商贸旅游       0.58      0.63      0.60       118
        城乡建设       0.61      0.57      0.59       115
        教育文体       0.73      0.82      0.77       124
        环境保护       0.74      0.83      0.79       126

    accuracy                           0.70       831
   macro avg       0.70      0.70      0.70       831
weighted avg       0.70      0.70      0.70       831
```

从上图我们得出结果：

准确率：0.70	
一级标签	F-Score
交通运输	0.68
劳动和社会保障	0.68
卫生计生	0.79

商贸旅游	0.60
城乡建设	0.59
教育文体	0.77
环境保护	0.79
平均值	0.70

从结果图中来看，y 为测试集预测结果，为了直观查看我们的预测结果，我们可以打印 y 和 labels_te 的内容出来（labels_te 为测试集实际数据标签），通过查看对比可看出是哪一个分类出错，然后通过前面的序号到文档中查找对应的留言内容。

```
In  [51]: labels_te

Out[51]: 6228    劳动和社会保障
         8508       卫生计生
         2452       环境保护
         8358       卫生计生
         6118    劳动和社会保障
                  ...
         2065       环境保护
         8647       卫生计生
         3202       交通运输
         2028       环境保护
         6424    劳动和社会保障
         Name: label, Length: 831, dtype: object
```

```
In  [50]: y

Out[50]: array(['卫生计生', '环境保护', '环境保护', '劳动和社会保障', '劳动和社会保障', '交通运输', '卫生计生',
                '城乡建设', '教育文体', '交通运输', '劳动和社会保障', '城乡建设', '教育文体', '商贸旅游', '环境保护',
                '卫生计生', '环境保护', '教育文体', '商贸旅游', '卫生计生', '卫生计生', '商贸旅游', '环境保护',
                '商贸旅游', '商贸旅游', '交通运输', '交通运输', '教育文体', '商贸旅游', '卫生计生', '商贸旅游',
                '卫生计生', '商贸旅游', '商贸旅游', '环境保护', '教育文体', '劳动和社会保障', '城乡建设', '卫生计生',
                '商贸旅游', '劳动和社会保障', '劳动和社会保障', '城乡建设', '环境保护', '卫生计生', '劳动和社会保障',
                '教育文体', '教育文体', '商贸旅游', '交通运输', '环境保护', '卫生计生', '环境保护', '教育文体',
                '教育文体', '城乡建设', '卫生计生', '商贸旅游', '劳动和社会保障', '城乡建设', '教育文体', '环境保护',
                '劳动和社会保障', '劳动和社会保障', '教育文体', '劳动和社会保障', '商贸旅游', '教育文体', '教育文体',
                '环境保护', '教育文体', '劳动和社会保障', '交通运输', '商贸旅游', '卫生计生', '卫生计生', '环境保护',
                '教育文体', '教育文体', '交通运输', '交通运输', '商贸旅游', '环境保护', '交通运输', '城乡建设',
```

1.4 模型优缺点

优点：该模型建模过程简单，易实现。也容易应用至其他数据分类。

缺点：该模型需要大量的训练数据才能使预测更准确。

至此，第一个分类问题结束。需要用到的库以及完整代码见附件。

2、热点问题挖掘

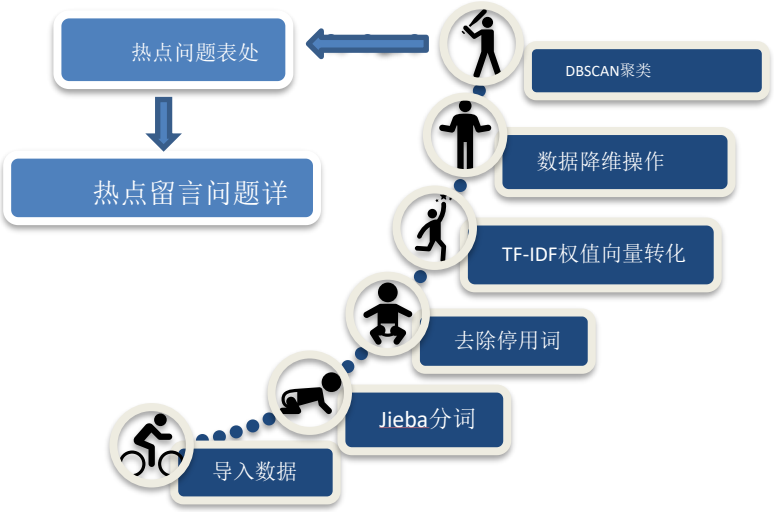
2.1 问题描述

某一时段内群众集中反映的某一问题可称为热点问题，如“XXX 小区多位业主多次反映入夏以来小区楼下烧烤店深夜经营导致噪音和油烟扰民”。及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。请根据附 3 将某一时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，并给出评价结果，按表 1 的格式给出排名前 5 的热点问题，并保存为文件“热点问题表.xls”。按表 2 的格式给出相应热点问题 对应的留言信息，并保存为“热点问题留言明细表.xls”。

2.2 解题思路

通过读问题要求和文档内容，知该问题是将收集到的留言信息统计整理，将同一个问题进行归类排序，有利于去找到当前或某一时间段的热点信息。所以我们可以认为这就是一个文本聚类的问题。用文本聚类的方法可以解决该问题。出于聚类的准确性，我们用“留言详情”的内容进行聚类操作。对于大致思路为：

1）导入数据。
2）去除停用词。
3）jieba 分词。
4）TF-IDF 权值向量转换。
5）数据降维操作。
6）使用 DBSCAN 的方法进行聚类。
7）热点问题表处理
8）热点留言问题明细表处理



热点问题挖掘流程图

2.3 具体步骤

1 2　导入文件

导入文件然后去除重复数据，代码如下：

```
data = pd.read_excel('附件3.xlsx','Sheet1')
data = data.drop_duplicates('留言详情')
```

```
data
```

	留言编号	留言用户	留言主题	留言时间	留言详情	反对数	点赞数
0	188006	A000102948	A3区一米阳光婚纱艺术摄影是否合法纳税了?	2019/2/28 11:25:05	\n\t\t\t\t\t\n\t\t\t\t\t座落在A市A3区联丰路米兰春天G2栋320，...	0	0
1	188007	A00074795	咨询A6区道路命名规划初步成果公示和城乡门牌问题	2019/2/14 20:00:00	\n\t\t\t\t\t\n\t\t\t\t\tA市A6区道路命名规划已经初步成果公示文件，...	0	1
2	188031	A00040066	反映A7县春华镇金鼎村水泥路、自来水到户的问题	2019/7/19 18:19:54	\n\t\t\t\t\t\n\t\t\t\t\t本人系春华镇金鼎村七里组村民，不知是否有相关...	0	1
3	188039	A00081379	A2区黄兴路步行街大古道巷住户卫生间粪便外排	2019/8/19 11:48:23	\n\t\t\t\t\t\n\t\t\t\t\t靠近黄兴路步行街，城南路街道、大古道巷、一步...	0	1
4	188059	A00028571	A市A3区中海国际社区三期与四期中间空地夜间施工噪音扰民	2019/11/22 16:54:42	\n\t\t\t\t\t\n\t\t\t\t\tA市A3区中海国际社区三期四期中间，即蓝天璞...	0	0

1 2 1 数据清洗

与问题 1 中的步骤一样，先导入停用词表，以备去停用词操作。

jieba 分词，将文本进行 jieba 分词，为了更好的对比，我们直接将 data 中的“留言详情”数据分词，然后建立新的一列存储分词结果，列名称为“详情切词”。

```
In [185]: stwlist = [line.strip() for line in open('tiyongci.txt','r',encoding='utf-8').readlines()]
          stwlist = ['', ' ', ',', '\t', '\n', '\u3000','领导','政府','国家'] + stwlist
          data['详情切词'] = data['留言详情'].apply(lambda i:jieba.lcut(i))
          data['详情切词'] = data['详情切词'].apply(lambda x: [i for i in x if i not in stwlist])
          data['详情切词'] = [' '.join(i) for i in data['详情切词']]
```

去除停用词，将通用停用词表文件（tiyongci.txt）导入，当然也将一些停用词表中没有的无用词添加进去。然后进行去除。去除之后将其以空格进行拼接。结果前 10 条如下：

```
In [195]: data['详情切词'][:10]
```

```
Out[195]: 0    座落在 市 A3 区联 丰路 米兰 春天 G2 栋 320 一家 名叫 一米阳光 婚纱 艺术...
          1    市 A6 区 道路 命名 规划 初步 成果 公示 文件 转化 正式 成果 希望 加快 路名 ...
          2    系 春华 镇金鼎村 七里 组 村民 不知 相关 水泥路 到户 政策 自来水 到户 政策 主导...
          3    靠近 黄兴路 步行街 城 南路 街道 古道 巷 一步 两 搭桥 小区 停车场 东面 围墙 外...
          4    市 A3 区 中海 国际 社区 三期 四期 蓝天 璞 洲 幼儿园 旁边 块 空地 处于 三不...
          5    麓 泉 社区 麓 谷 明珠 小区 栋 居民 近期 感觉 震惊 伤心 购房 签合同 办 产权 ...
          6    “ 二高 一部 发出 非法 集资 打击 通知 中是 金融 犯罪 通知 中 点 特征 中 欺骗...
          7    一名 市 地铁站 上班 安检员 中介 公司 介绍 上班 安检员 岗位 分 两个 班次 一个班...
          8    12 月 21 日 下午 17 时 52 分许 路 公交车 司机 座位 旁边 汽车 编号 3...
          9    保利 麓 谷林语 桐梓 坡路 与麓 松路 交汇处 地铁 凌晨 点 施工 噪音 扰民 家里 老...
          Name: 详情切词, dtype: object
```

TF-IDF 权值向量转化，并提取稀疏文本特征

```
vectorizer = TfidfVectorizer(max_df=0.5,max_features=40000,
                             min_df=5,stop_words=stwlist,ngram_range=(1,2),
                             use_idf=True)
x = vectorizer.fit_transform(data['详情切词'])
```

1 2 2 数据降维

使其适用于 DBSCAN 算法

由于 LSA/SVD 结果并未标准化，我们必须重做标准化。在进行降维操作，代码如下：

```
[216]: svd = TruncatedSVD(15)
       normalizer = Normalizer(copy=False)
       lsa = make_pipeline(svd,normalizer)
       x = lsa.fit_transform(x)
```

1 2 3　　B　M聚类处理

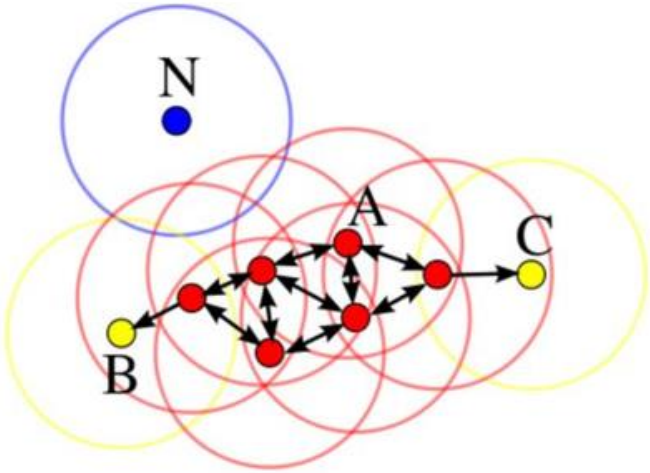
DBSCAN 是基于密度的空间聚类算法，还算法是将具有足够密度区域划分为簇。如下图，空间中有很多点，这些点分布不均匀，有稀疏有迷离。以其中一个点为中心点（核心对象），若规定在该点半径为 5 内如果发现临近有 3 个点（边界点）以上，则这些点聚为一个簇，反之不到 3 个点就不聚为簇。在刚刚聚成的簇中有更换中心点，又继续搜索，当满足条件时又把这些点收进该簇中，又继续更换中心点，重复刚才操作，直到找不到新的点加入。

基本概念：

A：核心对象

B,C：边界点

N：离群点



从该过程不难看出密度越大越容易聚在一起，反之不容易聚在一起。所以我们可以利用这个方法来解决我们的文本聚类问题。首先一条文本是可以转化成向量，也就是可以认为一条文本是一个空间点，在留言中，反映同一个问题的两条文本，它们的字词大部分是相同，且有些字词也会出现多次。当他们转成向量是其内容是相差不大的，这时我们就可以使用 DBSCAN 聚类算法将其聚为一类。从而解决我们的问题。需要注意的是，一些日常用语介词语气词等在任何文本中都会出现，这会降低我们聚类精确度，所以聚类之前一定要将这些词去掉。

降维操作结束后，得到需要聚类的数据 x，现将 x 放入 DBSCAN 算法模型当中，调整参数 eps（两个样本之间的最大距离，即扫描半径）和 min_samples（核心点的邻域内最大样本数）来找到最好的聚类效果，每一条聚类结果以数字形式输出。

```
db = DBSCAN(eps=0.2,min_samples=1).fit(x) #数据聚类
core_samples_mask = np.zeros_like(db.labels_,dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
```

这里我们先将 eps=0.2，min_samples=1，并将聚类标签以及噪点保存。

1 2 3 输出聚类结果

将切词后的二维表重新创建一列来保存聚类结果，列名为“cluster”，其结果数据和之前的数据相对应。它的内容为数值，相同数值为同一类，处理如下：

```
In [231]: labels = db.labels_      #统计聚类标签
          clusterTitles = db.labels_
          dbscandf = data
          dbscandf['cluster'] = clusterTitles  #将聚类的结果存入列名为“cluster”中。
```

我们可以直接输出它对应“留言主题”来查看。也可以统计聚类数和噪点数，同时我们还可以直接以图像来观察聚类情况。其代码和结果如下：

聚类统计

```
In [236]: n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
          n_noise_ = list(labels).count(-1)
          print('聚类数：',n_clusters_) #统计聚类数
          print('噪点数：',n_noise_)  #统计噪点数
```

```
聚类数： 3035
噪点数： 0
```

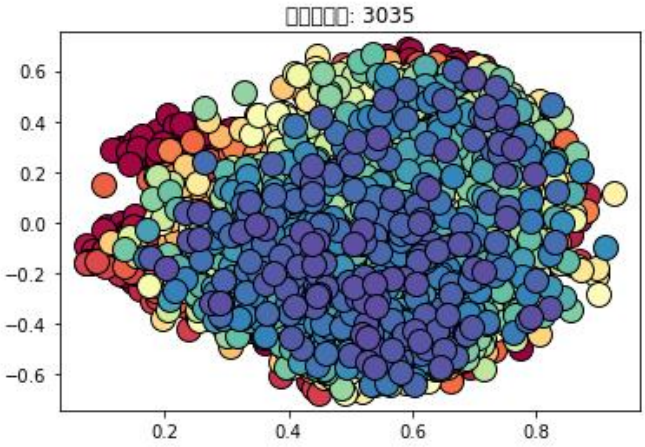
主题查看

```
In [235]: dbscandf[dbscandf['cluster'] ==0]['留言主题'].head(20)#查看聚类值为0的留言主题
```

```
Out[235]: 0    A3区一米阳光婚纱艺术摄影是否合法纳税了？
          Name: 留言主题, dtype: object
```

结果图形查看

```
import matplotlib.pyplot as plt
%matplotlib inline

unique_labels = set(labels)
colors = [plt.cm.Spectral(each)
          for each in np.linspace(0, 1, len(unique_labels))]


for k, col in zip(unique_labels, colors):
    if k == -1:
        col = [0, 0, 0, 1]

    class_member_mask = (labels == k)

    xy = x[class_member_mask & core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
             markeredgecolor='k', markersize=14)

    xy = x[class_member_mask & ~core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
             markeredgecolor='k', markersize=6)

plt.title('大致聚类数：%d' % n_clusters_)
```



结果输出表格

```
[239]: dbscandf
```

:[239]:

	留言编号	留言用户	留言主题	留言时间	留言详情	反对数	点赞数	详情切词	cluster
0	188006	A000102948	A3区一米阳光婚纱艺术摄影是否合法纳税了?	2019/2/28 11:25:05	\n\t\t\t\t\n\t\t\t\t座落在A市A3区联丰路米兰春天G2栋320，...	0	0	座落在 市 A3 区联 丰路 米兰 春天 G2 栋 320 一家 名叫 一米阳光 婚纱 艺术...	0
1	188007	A00074795	咨询A6区道路命名规划初步成果公示和城乡门牌问题	2019/2/14 20:00:00	\n\t\t\t\t\n\t\t\t\tA市A6区道路命名规划已经初步成果公示文件，...	0	1	市 A6 区 道路 命名 规划 初步 成果 公示 文件 转化 正式 成果 希望 加快 路名 ...	1
2	188031	A00040066	反映A7县春华镇金鼎村水泥路、自来水到户的问题	2019/7/19 18:19:54	\n\t\t\t\t\n\t\t\t\t本人系春华镇金鼎村七里组村民，不知是否有相关...	0	1	系 春华 镇金鼎村 七里 组 村民 不知 相关 水泥路 到户 政策 自来水 到户 政策 主导...	2
3	188039	A00081379	A2区黄兴路步行街大古道巷住户卫生间粪便外排	2019/8/19 11:48:23	\n\t\t\t\t\n\t\t\t\t靠近黄兴路步行街，城南路街道、大古道巷、一步...	0	1	靠近 黄兴路 步行街 城 南路 街道 古道 巷 一步 两 搭桥 小区 停车场 东面 围墙 外...	3
4	188059	A00028571	A市A3区中海国际社区三期与四期中间空地夜间施工噪音扰民	2019/11/22 16:54:42	\n\t\t\t\t\n\t\t\t\tA市A3区中海国际社区三期四期中间，即蓝天璞...	0	0	市 A3 区 中海 国际 社区 三期 四期 蓝天 璞 洲 幼儿园 旁边 块 空地 处于 三不...	4

1 2 4 热点问题表填写

时间范围处理

取出相同聚类数值对应的留言时间，然后再取出最大时间值和最小时间值，组合成时间范围。在具体操作中有时分秒的数据，我们只需用打断字符串函数，即 split()函数将其打断，再取前半部分的年月日即可，

问题详情处理

由于留言详情文本过多，难以提取主干部分，这里我们直接复制留言主题内容，大部分的主题内容就是留言详情的主题。

热度指数处理

热度指数可以通过该问题的留言数目、点赞数和反对数进行分析，再评定中留言数目占的比重肯定是要大些，居民遇到的问题可分为小问题和大问题，有时候小问题只是暂时的、只是出现在小部分人或者可以自行解决，这些问题很少去留言，而大问题是很急切，妨碍到大家的生活，这些问题往往难以自行解决，不管是大问题还是小问题，解决不了第一时间想到的还是去留言。留言越多说明这个问题妨碍到的人越多，说明该问题热点度较高。而点赞数表示该问题被赞成程度越高，反对数则表现出不赞成该问题。因此我们定义热度指数为：

$$热度指数 = 留言数目 \times 10 + 点赞数 - 反对数$$

热度指数越高，该该问题反映人数越多。

问题 ID 处理

通过热度指数处理后，我们可以将该内容从大到小进行排序，然后再问题 ID 列内从 1 开始编号。

热度排名处理

在前面我们已经将热度指数排序好了，现在我们只需要用该列进行排名即可。

地点/人群处理

在提取关键词的方法中可以利用词性标记进行提取，但是处理步骤繁多。也可以通过正则表达式提取关键词，该方法虽然简单，但是准确度不高。这里我们使用正则表达式，根据留言详情的内容，我们用正则表达式，提取出市、县、区、小区的前面部分的地址内容，通过观察数据，市县区的名字大都只有两个字，而小区的话字长都不等，我们尽量取长一点，这样在工作人员处理问题时可以通过里面超过或是缺失的地址内容判断出来。小区一般小于等于四字词语居多。所以我们取小区前的四个字。对于留言详情中没有出现地点的情况，我们默认他们为“市民”

处理过程

1）首先新建一个热点问题表格，用来存储全部的热点问题。

```
#提取cluster的数据用于标记
data_clu = data['cluster']
data_n = []
for i in data_clu:
    if i not in data_n:
        data_n = data_n + [i]
```

```
#建立热点问题表
n_data = pd.DataFrame(columns=('热度排名','问题ID','热度指数','时间范围','地点/人群','问题描述'))
```

2）将 data 的留言时间转成时间戳格式存入新列（int_time）中，用于后续的时间大小比较。

```
#时间的datatime格式转成时间戳

ct = []
data['留言时间']=data['留言时间'].apply(pd.to_datetime,format='%Y-%m-%d %H:%M:%S')
for i in dbscandf['留言时间']:
    timeStamp = i.replace(tzinfo=datetime.timezone.utc).timestamp()
    ct.append(timeStamp)
data['int_time'] = ct
```

3）根据聚类后的 data 表的 cluster 中的数据，其数字相同就是同一类。我们根据这个特征，依次将相同数字的一行提取出来。将该类第一条留言主题直接复制到热点问题表格。

4）将该类的时间戳（int_time）进行比较。再把最大和最小时间戳对应的留言时间提取出来，用空格打断只取前面的日期，最小日期和最大日期中间用“至”连接起来。然后存入热点问题表的时间范围。

```
#热点问题表数据填写

b=0
for i in data_n:
    index = data['cluster'] == i
    dataa = data[:][index]

#     问题描述输出
    daque=dataa['留言主题'].head(1)
    daque=str(daque).split()[1]

# 时间范围输出
    datime=dataa['int_time']
    for j in datime:
        if (j>b):
            b=j
    d=b
    for k in datime:
        if (k<d):
            d=k
```

```
    c = data['int_time'] == b
    c1 = data['留言时间'][c]
    d1 = data['int_time'] == d
    d2 = data['留言时间'][d1]
    max_date = str(c1).split()[1]
    min_date = str(d2).split()[1]
    if max_date!=min_date:
        shijian = min_date +'至'+max_date
        print('时间'+shijian)
    if max_date==min_date:
        shijian = min_date
    b=0    #初始化b
```

5）统计该类条数，将该类所有的点赞数相加，所有的反对数也相加，通过公式将热度指数计算出来，并把结果存入热点问题表。

```
# 热度指数输出
    hang = len(dataa)
    c_agree = 0
    c_disagree = 0
    agree = dataa['点赞数']
    disagree = dataa['反对数']
    for i in agree:
        c_agree = c_agree + i
    for j in disagree:
        c_disagree = c_disagree + j
    HI = hang * 10 + c_agree - disagree
    HI = str(HI).split()[1]
    HI = int(HI)
```

6）用正则表达式通配“英文+市”或“英文+数字+市”提取市名称。用该方法继续通配区名，小区名。最后通配不到，则以“市民”替代。并存入热点问题表的“地点/人群”。

```
# 地点人群输出
    xx = ''
    xiang = dataa['留言详情']
    for i in xiang:
        xx=xx+i
    zz=''
    a = re.findall('\w市|\w\d市',xx)
    if a!=[]:
        zz=a[0]
    l = re.findall('\w\d区',xx)
    if l!=[]:
        zz=zz+l[0]
    h = re.findall('\w\w\w\w小区',xx)
    if h!=[]:
        zz=zz+h[0]
    if h==[]:
        zz = zz + '市民'
```

```
#输出到热点问题表
    n_data = n_data.append(pd.DataFrame({'时间范围':[shijian],'问题描述':[daque],'热度指数':[HI],'地点/人群':[zz]}))
```

7）再将热点问题表中热度指数排序后填入从 1 开始对“问题 ID”编码。对热度指数排名并将结果填入“热度排名”，最后将前 5 条数据导出到“热点问题表。 x l s ”，详细代码见附件。

```
n_data=n_data.sort_values(by='热度指数',ascending=False)
n_data=n_data.reset_index(drop=True)
```

```
# 将热度指数正数化
rmin = n_data['热度指数'].min()
if rmin<=0:
    n_data['热度指数'] = n_data['热度指数'] - rmin
```

```
# 问题ID编码
num = 1
for i in range(len(n_data)):
    n_data.at[i,'问题ID'] = num
    num = num+1
```

```
#热度排名
v = n_data['热度指数']
v = v.rank(method='min',ascending=False)
n_data['热度排名']=v
```

```
# 取前5条数据
qian5 = n_data.head(5)
```

```
# 输出到xls文件
qian5.to_excel('热点问题表.xls',index=None)
```

1 2 5 热点留言问题明细表填写

我们的热点问题表中的热度指数是由 data 表中计算而来的，所以我们可以通过热点问题表中的热度指数使两个表格联系起来。这样我们就可以将热点问题表中的问题 ID 复制到 data 表格中，再将 data 表格，按照已给表格样例导出到热点留言明细表中。步骤及代码如下

1）在 data 中新建一列“热度指数”，利用刚才计算输出到热点问题表的方法，将热度指数计算出来并存入新建列中。

```
# 在data中添加热度指数
# 先将数据提取出来处理好后才存入data中

ple = data[['cluster','点赞数','反对数']]
ple['热度指数']=None
for i in data_n:
    index = ple['cluster'] == i
    dataa1 = ple[:][index]
    hang1 = len(dataa1)
    c_agree1 = 0
    c_disagree1 = 0
    agree1 = dataa1['点赞数']
    disagree1 = dataa1['反对数']
    for i in agree1:
        c_agree1 = c_agree1 + i
    for j in disagree1:
        c_disagree1 = c_disagree1 + j
    HI1 = hang1 * 10 + c_agree1 - disagree1
    HI1 = str(HI1).split()[1]
    HI1 = int(HI1)
    ple['热度指数'][index]=HI1
data['热度指数']=ple['热度指数']
```

2）在 data 表中再新建一列“问题 ID”，取出 n_data（热点问题表）的“问题 ID”和“热度指数”列。相应的与取出 data 表中的热度指数，从 n_data 中一个一个的取出热度指数中的数据和 data 中的热度指数相比较，两者相等则将 n _data 中的问题 ID 存入 data 的问题 ID 中。

```
data = data.reindex(columns=['问题ID','留言编号','留言用户','留言主题','留言时间','留言详情','反对数','点赞数','热度指数'])
mle = n_data[['问题ID','热度指数']]
rmin = data['热度指数'].min()
if rmin<=0:
    data['热度指数'] = data['热度指数'] - rmin
for i in mle['热度指数']:
    fle = n_data['问题ID'][i]
    for j in data['热度指数']:
        if j==i:
            data['问题ID'][j]=fle
```

3）将 data 表按“问题 ID”从小到大排序。

```
data=data.sort_values(by='问题ID',ascending=True)
```

4）输出到文件“热点留言问题明细表”。

```
# 输出到xls文件
data = data.reindex(columns=['问题ID','留言编号','留言用户','留言主题','留言时间','留言详情','反对数','点赞数'])
data.to_excel('热点留言问题明细表.xls',index=None)
```

3、答复意见的评价

3.1 问题描述

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。

3.2 解题思路

附件 4 数据格式如下：

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
2549	A00045581	区景蓉华苑物业管理有问题	2019/4/25 9:32:09	物业公司却以交20万保证金，不	意收取停车管理费，在业主大会结束后业委会	2019/5/10 14:56:53
2554	A00023583	潇楚南路洋湖段怎么还没修	2019/4/24 16:03:40	店面的生意带来很大影响，里面，	需整体换墙，且换墙后还有三通雨污水管道	2019/5/9 9:49:10
2555	A00031618	快提高A市民营幼儿园老师的	2019/4/24 15:40:04	的同时更是加大了教师的工作压	民办幼儿园聘任教职工要依法签订劳动合同，依	2019/5/9 9:49:14
2557	A000110735	公寓能享受人才新政购房补	2019/4/24 15:07:30	落户A市，想买套公寓，请问购	年龄35周岁以下（含），首次购房后，可分别	2019/5/9 9:49:42
2574	A0009233	于A市公交站点名称变更的建	2019/4/23 17:03:19	“马坡岭小学”，原“马坡岭	保留“马坡岭”的问题。公交站点的设置需要	2019/5/9 9:51:30
2759	A00077538	A3区含浦镇马路卫生很差	2019/4/8 8:37	再把泥巴冲到右边，越是上下班	于您问题中没有说明卫生较差的具体路段，也	2019/5/9 10:02:08
2849	A000100804	区教师村小区盼望早日安装电	2019/3/29 11:53:23	台为老社区惠民装电梯的规范性	A市A3区人民政府办公室下发了《关于A市A3区	2019/5/9 10:18:58
3681	UU00812	区东澜湾社区居民的集体民	2018/12/31 22:21:59	跑好远，天寒地冻的跑好远，眼	装修前期准备及设施设备采购等工作。下一步	2019/1/29 10:53:00
3683	UU008792	襄阳光住宅楼无故停工以及	2018/12/31 9:55:00	也没得到相关准确开工信息。住	住单位落实分户检查后，西地省楚江新区建设	2019/1/16 15:29:43
3684	UU008687	和顺路洋湖壹号小区路段公	2018/12/31 9:45:59	立交桥等地方做立体绿化，取	余部分也按规划要求完成了建设，其中西边绿化	2019/1/16 15:31:05
3685	UU0082204	A2区大托街道大托新村违建	2018/12/30 22:30:30	乡规划局审批通过《温室养殖大	公司支付一笔耕地征收补偿款给原大托村，但	2019/3/11 16:06:33
3692	UU008829	鄱阳村D区安置房人防工程的	2018/12/29 23:27:51	D区安置房地下室近两万平方米续，	按长人防发[2014]7号文件要求，鄱阳村	2019/1/29 10:52:01

1）从图中我们可以看出，如果我们需要处理该条问题，也就是填写答复意见，我们就需要从留言详情中去看问题，才能针对问题做答复。所作答复的内容肯定要与留言详情相关，也就是两者会出现大部分相同内容，我们通过判断两者文本单词相似度，当达到一定值时便认为两者具有相关性，也就是做到了所问所答，这是评价答复质量的一方面。

相关性指数可以这么算，将留言详情的文本，先做

2）同时我们还可以看到每条留言是有留言时间，每次答复也是有答复时间。我们可以通过查看两者相隔的时间长短来得出答复的及时性，这也可以作为评价答复质量的另一方面。

3）对于评价答复质量，相关性和及时性两者重要程度肯定是不相等的，相关性肯定是占的比重大些，及时性占的小一些，所以我们可以分别计算两者的指数值，再乘以一个权重值，两者相加得出答复质量。以下是我们定义的一个公式。

$$答复质量 = 相关性指数 \times 0.7 + 及时性指数 \times 0.3$$

3.3 实现方案

2 2 相关性指数

1）相关性主要是比较留言与答复的文本相似度，所以我们还是要在文本中去分析。我

们可以这么做，将留言详情先进行文本清洗，即 jieba 分词，去除停用词，然后进行文本向量转化，TF-IDF 权值策略。到这一步其实我们就可以通过权重得到那些词是出现文本中的次数是比较多的，这时我们可以取出权重值排在前面的单词，同时可以认为这些单词就是该留言的主要内容，为了指数的正确性，我们应该尽量多的取权重排前的单词。

2）相应的在答复意见中我们也对其做同样的处理，也取出权重值较高的单词，注意该单词数量应与留言中取得单词数量一致。

3）统计留言单词数量作为单词总数，利用比较算法，将答复意见中单词一个一个的和留言中的单词比较，记录相同单词个数，将相同数量除以单词总数可以得到相关性指数。

$$相关性指数 = \frac{相同单词数量}{留言单词总数} \times 100\%$$

获取 TF-IDF 权值较高的单词代码：

```
import jieba
import pandas as pd
import jieba.analyse
data = pd.read_csv('data3.csv',encoding='gb18030')
data = data.loc[:,['留言编号','留言详情']]
data.columns = ['label', 'message']
data = data['message'].sample(30)

stopWords = pd.read_csv('tiyongci.txt', encoding='utf-8', sep='hahaha', header=None)
stopWords = list(stopWords.iloc[:, 0])

def word_cut(fenci_text):
    final = ""
    for i in fenci_text:
        if i not in stopWords:
            if (i != '。' and i != '，'):
                final = final + ' ' + i
    return final
jieba.load_userdict('newdic2.txt')
f = []
d = []
for j in range(len(data)):
    text = data[j]
    fenci_text = jieba.cut(text)
    b = word_cut(fenci_text)
    f.append(b)
for k in range(len(f)):
    a = jieba.analyse.extract_tags(f[k], topK=5, allowPOS= ())
    d.append(a)
print(d)
```

2.2.1 及时性指数

1）将答复时间减去留言时间，我们可以得到一个时间段，这段时间我们可以认为是处理时长，我们可以规定处理时长为多少那么他的及时性指数为多少。以下是我们定义的一个及时性表格。

处理时长	及时性指数
1 天内	100%
一星期内	80%
半个月内	70%
1 个月内	60%
2 个月内	50%
3 个月内	40%
3 个月以上	20%

2.2.2 答复质量

利用前面答复质量公式，我们可以算出答复质量指数，指数最高为 1，指数越高，说明该答复越好反之则越差。

参考文献

https://www.kesci.com/home/project/5c19f99de17d84002c658466
https://edu.tipdm.org/classroom/97/introduction
https://blog.csdn.net/linchuhai/article/details/88526476
https://blog.csdn.net/huacha__/article/details/81094891

附件

1、群众留言分类详细代码：

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.metrics import classification_report
import pandas as pd
import jieba

data = pd.read_csv('附件 2.csv',encoding='gb1803a0') #encoding 为设置数据编码
data = data.loc[:,['留言详情','一级标签']]              #导入数据只取留言详情和一级标签的内容放在 data 中
data.columns = ['message', 'label']                    #将标签进行更换
n = 600                                                #设置各标签选取数量
a = data[data['label'] == '城乡建设'].sample(n)
b = data[data['label'] == '商贸旅游'].sample(n)
c = data[data['label'] == '卫生计生'].sample(n)
d = data[data['label'] == '交通运输'].sample(n)
e = data[data['label'] == '劳动和社会保障'].sample(n)
f = data[data['label'] == '教育文体'].sample(n)
g = data[data['label'] == '环境保护''].sample(n)       #选取 7*600 条数据
data_new = pd.concat([a,b,c,d,e,f,g],axis=0)           #将取出的数据进行拼接整合
data_dup = data_new['message'].drop_duplicates()     #对去除重复数据
data_cut = data_dup.apply(lambda x: jieba.lcut(x))
stoplist = pd.read_csv('stopword.txt',encoding='gbk',sep='hahaha',header=None)
stoplist = [' ','\n\r','\n','\t','\u3000']+list(stoplist.iloc[:,0])
data_after_stop = data_cut.apply(lambda x: [i for i in x if i not in stoplist])
labels = data_new.loc[data_after_stop.index, 'label']
adata = data_after_stop.apply(lambda x: ' '.join(x))

data_tr, data_te, labels_tr, labels_te = train_test_split(adata, labels, test_size=0.2)#划分训练集和测试集
countVectorizer = CountVectorizer()
data_tr = countVectorizer.fit_transform(data_tr)       #文本形式的训练样本转化成向量
X_tr = TfidfTransformer().fit_transform(data_tr.toarray()).toarray()    #训练集转化成 TF-IDF 权值向量
data_te = CountVectorizer(vocabulary=countVectorizer.vocabulary_).fit_transform(data_te) #测试集转成向量形式
X_te = TfidfTransformer().fit_transform(data_te.toarray()).toarray() #测试集转成 TF-IDF 权值向量

model = GaussianNB()    #导入朴素贝叶斯分类模型
model.fit(X_tr, labels_tr)    #模型训练
y = model.predict(X_te) #模型测试
print("准确率：",model.score(X_te, labels_te))    #输出准确率
print("其他指标:",classification_report(labels_te, y, target_names=None))    #输出各标签分类的指标
```

2、热点问题挖掘详细代码：

```
from sklearn.decomposition import PCA
from sklearn.decomposition import TruncatedSVD
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import Normalizer
from sklearn import metrics
from sklearn.pipeline import make_pipeline
from pprint import pprint
import logging
import datetime
import time
from time import time
import numpy as np
import os
import re
from sklearn.cluster import DBSCAN
import pandas as pd
import jieba
data = pd.read_excel('附件 3.xlsx','Sheet1')
stwlist = [line.strip() for line in open('tiyongci.txt','r',encoding='utf-8').readlines()]
stwlist = ['', ' ', ',', '\t', '\n', '\u3000','领导','政府','国家'] + stwlist
data['详情切词'] = data['留言详情'].apply(lambda i:jieba.lcut(i))
data['详情切词'] = data['详情切词'].apply(lambda x: [i for i in x if i not in stwlist])
data['详情切词'] = [' '.join(i) for i in data['详情切词']]
vectorizer = TfidfVectorizer(max_df=0.5,max_features=40000,
                             min_df=5,stop_words=stwlist,ngram_range=(1,2),
                             use_idf=True)
x = vectorizer.fit_transform(data['详情切词'])
svd = TruncatedSVD(15)
normalizer = Normalizer(copy=False)
lsa = make_pipeline(svd,normalizer)
x = lsa.fit_transform(x)
db = DBSCAN(eps=0.2,min_samples=1).fit(x) #数据聚类
core_samples_mask = np.zeros_like(db.labels_,dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_         #统计聚类标签
clusterTitles = db.labels_
dbscandf = data
dbscandf['cluster'] = clusterTitles    #将聚类的结果存入列名为“cluster”中。
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)
print('聚类数：',n_clusters_) #统计聚类数
print('噪点数：',n_noise_)    #统计噪点数
dbscandf[dbscandf['cluster'] ==0]['留言主题'].head(20)#查看聚类值为 0 的留言主题
```

```
import matplotlib.pyplot as plt
%matplotlib inline

unique_labels = set(labels)
colors = [plt.cm.Spectral(each)
          for each in np.linspace(0, 1, len(unique_labels))]


for k, col in zip(unique_labels, colors):
    if k == -1:
        col = [0, 0, 0, 1]

    class_member_mask = (labels == k)

    xy = x[class_member_mask & core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
             markeredgecolor='k', markersize=14)

    xy = x[class_member_mask & ~core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
             markeredgecolor='k', markersize=6)

plt.title('大致聚类数: %d' % n_clusters_)
#提取 cluster 的数据用于标记
data_clu = data['cluster']
data_n = []
for i in data_clu:
    if i not in data_n:
        data_n = data_n + [i]
#时间的 datatime 格式转成时间戳

ct = []
data['留言时间']=data['留言时间'].apply(pd.to_datetime,format='%Y-%m-%d %H:%M:%S')
for i in dbscandf['留言时间']:
    timeStamp = i.replace(tzinfo=datetime.timezone.utc).timestamp()
    ct.append(timeStamp)
data['int_time'] = ct
#建立热点问题表
n_data = pd.DataFrame(columns=('热度排名','问题 ID','热度指数','时间范围','地点/人群','问题描述'))
#热点问题表数据填写

b=0
for i in data_n:
```

```
        index = data['cluster'] == i
        dataa = data[:][index]

#         问题描述输出
        daque=dataa['留言主题'].head(1)
        daque=str(daque).split()[1]

# 时间范围输出
        datime=dataa['int_time']
        for j in datime:
            if (j>b):
                b=j
        d=b
        for k in datime:
            if (k<d):
                d=k
        c = data['int_time'] == b
        c1 = data['留言时间'][c]
        d1 = data['int_time'] == d
        d2 = data['留言时间'][d1]
        max_date = str(c1).split()[1]
        min_date = str(d2).split()[1]
        if max_date!=min_date:
            shijian = min_date +'至'+max_date
        if max_date==min_date:
            shijian = min_date
        b=0       #初始化 b

# 热度指数输出
        hang = len(dataa)
        c_agree = 0
        c_disagree = 0
        agree = dataa['点赞数']
        disagree = dataa['反对数']
        for i in agree:
            c_agree = c_agree + i
        for j in disagree:
            c_disagree = c_disagree + j
        HI = hang * 10 + c_agree - disagree
        HI = str(HI).split()[1]
        HI = int(HI)

# 地点人群输出
        xx = ''
```

```
        xiang = dataa['留言详情']
        for i in xiang:
            xx=xx+i
        zz=''
        a = re.findall('\w 市|\w\d 市',xx)
        if a!=[]:
            zz=a[0]
        l = re.findall('\w\d 区',xx)
        if l!=[]:
            zz=zz+l[0]
        h = re.findall('\w\w\w\w 小区',xx)
        if h!=[]:
            zz=zz+h[0]
        if h==[]:
            zz = zz + '市民'

#输出到热点问题表
        n_data  =  n_data.append(pd.DataFrame({'时间范围':[shijian],'问题描述':[daque],'热度指数
':[HI],'地点/人群':[zz]}))
#  热度指数排序
n_data=n_data.sort_values(by='热度指数',ascending=False)
n_data=n_data.reset_index(drop=True)
#  将热度指数正数化
rmin = n_data['热度指数'].min()
if rmin<=0:
        n_data['热度指数'] = n_data['热度指数'] – rmin
#  问题 ID 编码
num = 1
for i in range(len(n_data)):
        n_data.at[i,'问题 ID'] = num
        num = num+1
#热度排名
v = n_data['热度指数']
v = v.rank(method='min',ascending=False)
n_data['热度排名']=v
#  取前 5 条数据
qian5 = n_data.head(5)
#  输出到 xls 文件
qian5.to_excel('热点问题表.xls',index=None)
ple=data[['cluster','点赞数','反对数',]]
ple['热度指数']=None
for i in data_n:
        index=ple['cluster']==i
        dataa1=plr[:][index]
```

```
        hang1 = len(dataa1)
            c_agree1 = 0
            c_disagree1 = 0
            agree1 = dataa1['点赞数']
            disagree1 = dataa1['反对数']
            for i in agree1:
                c_agree1 = c_agree1 + i
            for j in disagree1:
                c_disagree1 = c_disagree1 + j
            HI1 = hang1 * 10 + c_agree1 - disagree1
            HI1 = str(HI1).split()[1]
            HI1 = int(HI1)
            ple['热度指数'][index]=HI1
data['热度指数']=ple['热度指数']
data  =  data.reindex(columns=['问题 ID','留言编号','留言用户','留言主题','留言时间','留言详情
','反对数','点赞数','热度指数'])
mle = n_data[['问题 ID','热度指数']]
rmin = data['热度指数'].min()
if rmin<=0:
    data['热度指数'] = data['热度指数'] - rmin
for i in mle['热度指数']:
    fle = n_data['问题 ID'][i]
    for j in data['热度指数']:
        if j==i:
            data['问题 ID'][j]=fle
data=data.sort_values(by='问题 ID',ascending=True)
#  输出到 xls 文件
data.to_excel('热点留言问题明细表.xls',index=None)
```