“智慧政务”中的文本挖掘应用

摘要

随着社会和科技的不断发展，互联网已经成为我们生活中不可或缺的部分，群众可以通过一些平台获得自己想要的信息，政府机构也可以通过指定的平台进行网络问政，从而更好地了解民意，提高人民生活质量。但单靠人工对群众留言进行整理，不仅效率低，工作人员的负荷也会大增加，错误率增大。随着大数据、云计算、人工智能等技术的发展和运用，如何用这些技术快速、高效、精准处理群众留言便成了热点。因此，运用自然语言处理和文本挖掘的方法对客户留言信息的研究具有重大的意义，本文主要运用$python$进行模型的搭建。

针对问题1，通过提取附件2客户留言信息进行数据处理预处理，然后运用$jieba$中文分词工具对客户留言详情进行分词，并绘制出相应一级标签词云图，其次我们利用$TF-IDF$算法计算出留言信息的$TF-IDF$权值向量，然后建立关于分类的朴素贝叶斯模型和支持向量机模型，提取80%对模型进行训练，并用剩余的20%对模型进行检验，最后采用$F-score$算法对模型进行评价，得出朴素贝叶斯的$F1=0.828$，支持向量机的$F1=0.848$，即得出支持向量机分类比朴素贝叶斯较好。

针对问题2，根据附件3提取留言主题等重要信息，运用层次聚类算法、$TF-IDF$算法等思想建立对留言信息热点问题模型，通过模型得出关于留言信息的热点详情，然后提取排名前5的热点问题及详细信息，随后建立热点问题表和热点问题留言明细表。

针对问题3，提取附件4的客户留言详情和答复意见信息，运用余弦相似度算法，找出答复意见与留言详情之间的相似度，建立关于答复意见质量的评价体系。

关键词：$python$、$TF-IDF$算法、朴素贝叶斯、支持向量机、层次聚类算法、余弦相似度

目录

1 问题重述

1.1 背景介绍

近年来，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、 汇聚民智、凝聚民气的重要渠道，各类社情民意相关的文本数据量不断攀升，给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。

智慧政务就是通过“互联网+政务服务”构建智慧型政府，利用云计算、移动物联网、人工智能、数据挖掘、知识管理等技术，提高政府在办公、监管、服务、决策的智能水平，形成高效、敏捷、公开、便民的新型政府，实现由“电子政务”向“智慧政务”的转变。智慧政务有四大领域：智慧办公、智慧监管、智慧服务、智慧决策，主要包括城市服务、智慧公安、智慧税务、智慧交管、智慧医疗、智慧教育等诸多行业，覆盖各省、市、县各级行政单位，运用互联网、大数据等现代信息技术，加快推进部门间信息共享和业务协同，简化群众办事环节、提升政府行政效能、畅通政务服务渠道，解决群众“办证多、办事难“等问题，为群众提供多渠道、无差别、全业务、全过程的便捷服务。

1.2 需要解决的问题

(1) 基于群众留言内容详情建立关于群众留言内容的一级标签分类模型，并利用$F-score$对模型进行评价，找出最优的模型。

(2) 根据群众留言主题的内容，用层次聚类算法，聚类出相似度较高的问题，并利用自定义的热度评价指标挖掘出热点问题。

(3) 对留言与答复进行相似度计算，从而得到留言与答复的相关性、完整性、可解释性分析。

2 问题分析

2.1 问题一的分析

针对问题一，我们要建立关于客户留言内容的一级标签分类模型，由于客户留言内容信息包含数据量多且杂，无法进行正常的分析，因此，我们首先对附件2中客户留言详情运用$jieba$库进行中文分词并去除停用词，得到能够分类的词库，然后运用$TF-IDF$算法计算词的权值，并提取出每一类最为相关的词。权值计算之后，我们分别建立朴素贝叶斯分类模型和支持向量机分类模型，将样本数据进行训练和测试，得到了对客户留言信息的一级标签分类模型。最后我们运用$F-score$算法对两类分类模型进行检验，得出支持向量机分类模型精度优于朴素贝叶斯分类模型，因此选择支持向量机分类模型作为该分类模型。

2.2 问题二的分析

针对问题二，对于群众的留言，除了关注留言的分类标签外，相关部门还非常关心某段时间内群众集中反映的问题，即热点问题。问题二旨在挖掘出样本数据排名前五的热点问题，对其排序，并将对应问题的详细信息列出。

首先，本文将基于附件3的留言主题这一列数据，利用层次聚类($Hierarchical\ clustering$)算法建立模型，对留言主题进行相似度计算，再对相似度较高的留言主题进行归类与排序。其次，本文定义热度指数与归类后某一问题的留言条数及对应的点赞数、反对数有关，据此建立热度指数评价的公式。最后，根据上述步骤得到归类并排序后的热点问题留言明细表，以及热点问题表。

2.3 问题三的分析

针对问题三，答复意见的质量反映了政府对群众的关心程度，显得尤为重要，本文从相关性、及时性、完整性、可靠性、可解释性对答复意见的质量进行评价，建立一套评价体系，本文主要从答复意见与留言详情的相关性着手，分析了每行文本对应的相似度，将此作为主要依据形成评价体系。

3 模型建立与求解

3.1 问题一：构建基于留言内容的一级标签分类模型

3.1.1 基本流程

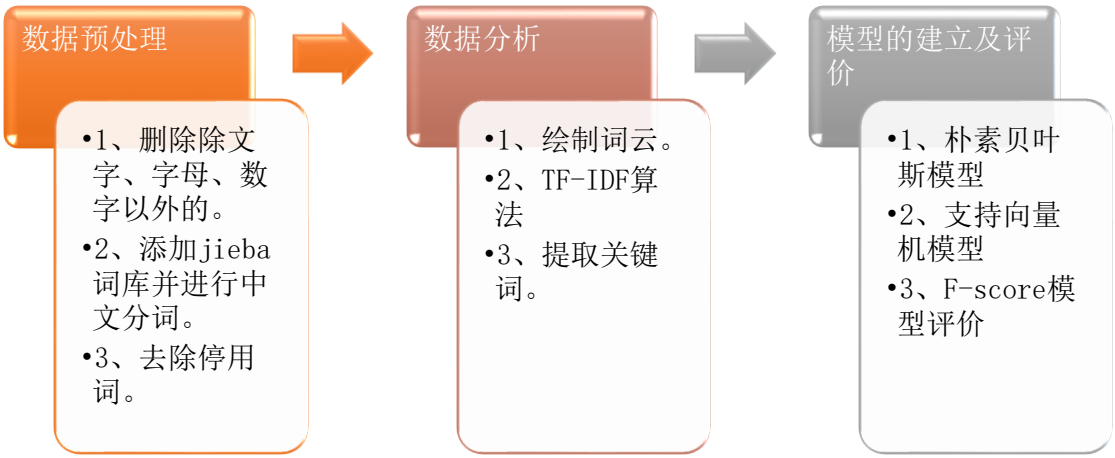

图1：基本流程图

3.1.2 数据预处理

(1) 数据去重、去空

首先我们提取附件2中客户“留言详情”一列，考虑数据中含有重复数据或者含有空数据，因此，我们对提取的数据删除其重复数据及空白数据，并对文本中的空格进行删除，方便后续操作。其次我们提取了用于分类的“一级标签”，并对其进行编号，其中七个编号分别是：城乡建设[0]；环境保护[1]；交通运输[2]；教育文体[3]；劳动和社会保障[4]，商贸旅游[5]，卫生计生[6]。编号的目的在于在模型建立中使计算机能够识别。具体代码如下图：

```
#标签向量化
data_new['label']=data_new['一级标签'].factorize()[0]
cat_id_df=data_new[['一级标签','label']].drop_duplicates().sort_values('label').reset_index(drop=True)
cat_to_id=dict(cat_id_df.values)
id_to_cat=dict(cat_id_df[['label','一级标签']].values)
data_dup = data_new['留言详情'].drop_duplicates()#去重


#删除除文字、字母、数字以外的
import re
def remove_punctuation(line):
    line = str(line)
    if line.strip()=='':
        return ''
    rule = re.compile(u"[^a-zA-Z0-9\u4E00-\u9FA5]")
    line = rule.sub('',line)
    return line
data_qumin = data_dup.apply(remove_punctuation)
```

图2

(2) 添加*jieba*词库、进行中文分词

由于数据中含有的地点名称等词库*jieba*库中没有，所以我们需要运用*jieba.load_userdict*添加相应的词库，词库文件见作品附件（*word.txt*），添加词库后，对处理好的留言详情运用*jieba*进行中文分词。*Jieba*分词采用了基于前缀词典现实的高效词图扫描，生成句子中所以可能成词的情况所构成的有向无环图（*DAG*），同时采用了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，对于未登陆词，采用了基于汉字成词能力的*HMM*模型，使得能更好的实现中文分词。具体代码如下图：

```
#添加jieba词库并分词
import jieba
jieba.load_userdict('./word.txt')
data_cut = data_qumin.apply(lambda x: jieba.lcut(x))
```

图3

(3) 去停用词

对文本数据进行分词后，数据中含有大量的与分类不相关的词，因此，我们需要对分的词去停用词，停用词表是网上常见停用词表见作品附件（*stopword.txt*），具体代码如下：

```
#提取停用词库并去停用词
stopWords = pd.read_csv('./stopword.txt', encoding='GB18030', sep='haha', header=None)
stopWords = [' ','不', '?', '≠', '不','会', '月', '日', '-','男','女','岁','斤',',','.', '。'] + list(stopWords.iloc[:, 0])
stopWords = list(set(stopWords))
data_after_stop = data_cut.apply(lambda x: [i for i in x if i not in stopWords])
print('data_after_stop',data_after_stop)
```

图4

3.1.3 数据分析

(1) 绘制词云

对“留言详情”进行分词、去停用词等操作后，得到了能够进行分类的词库，为了能够直观的看出每一类中词出现频率情况，我们绘制出了每一类的词云图，“城乡建设”的词云图如下，具体词云图见附件文件（词云图）

图5：城乡建设词云

(2) $TF-IDF$算法

原理： $TF-IDF$是用于估计一个词对于一个文件集或一个语料库中的其中一份文件的重要程度，其中，词频（TF）指某一个给定的词语在该文件中出现的次数。即

$$TF=\frac{\text{在某一类中词条}w\text{出现的次数}}{\text{该类中所以的词条数}}$$

逆向文件词频（IDF）的主要思想是如果包含词条的文档越少，则IDF越大，则说明词条具有很好的类别区分能力。某一词的IDF，可以由总文件数除以包含该词的文件数目，再取对数得到，即

$$IDF = \log(\frac{\text{语料库中的文档数}}{\text{包含词条的文档数} + 1})$$

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生高权重的$TF - IDF$。

$$TF - IDF = TF * IDF$$

生成$TF - IDF$向量：首先我们对上述所分的词索引存入$adata$，对所得的编号进行索引存入$labels$，将他们分为训练集和测试集两部分，分别求出其$TF - IDF$值，具体代码如下图6：

```
#计算tfidf权值
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
data_tr, data_te, labels_tr, labels_te = train_test_split(adata, labels, test_size=0.2)
countVectorizer = CountVectorizer()
data_tr = countVectorizer.fit_transform(data_tr)
X_tr = TfidfTransformer().fit_transform(data_tr.toarray()).toarray()
data_te = CountVectorizer(vocabulary=countVectorizer.vocabulary_).fit_transform(data_te)
X_te = TfidfTransformer().fit_transform(data_te.toarray()).toarray()
```

图6

(3) 提取关键词

为了能够了解每一类中最相关的词，我们运用$python$整理出了与每一类最相关的3个关键词及3对词组如下表：

表1：关键词表

标签	关键词	词组
城乡建设	开发商	棚户区 改造
	小区	住房 公积金
	业主	公积金 贷款
环境保护	排放	生活 环境
	环保局	周边 居民
	污染	环保局 领导
交通运输	的士	出租车 公司
	快递	的士 司机

	出租车	出租车 司机
	教师	代课 教师
教育文体	学生	培训 机构
	学校	教育局 领导
	职工	退休 工资
劳动和社会保障	退休	劳动 关系
	社保	退休 人员
	景区	电梯 故障
商贸旅游	传销	小区 电梯
	电梯	传销 组织
	独生子女	独生子女 父母
卫生计生	医生	社会 抚养费
	医院	乡村 医生

3.3.4 模型建立与评价

(1) 朴素贝叶斯分类模型

朴素贝叶斯分类（NBC）是以贝叶斯定理为基础并且假设特定条件之间相互独立的方法，先通过已给定的训练集，以特征词之间独立作为前提假设，学习从输出到输入的联合概率分布，再基于学习到的模型，输入X求出使得后验概率最大的输出Y。

设有样本数据集$D=\{d_1,d_2,\cdots,d_n\}$，对应样本数据的特征属性集为$X=\{x_1,x_2,\cdots,x_d\}$，类变量为$Y=\{y_1,y_2,\cdots,y_m\}$，即D可以分为y_m类别。其中$x_1,x_2,\cdots,x_d$相互独立且随机，则Y的先验概率$P_{prior}=P(Y)$，Y 的后验概率$P_{post}=P(Y|X)$，由朴素贝叶斯算法可得，后验概率可以由先验概率$P_{prior}=P(Y)$、概率$P(X)$、类条件概率$(Y|X)$计算出：

$$P(Y|X)=\frac{P(Y)P(X|Y)}{P(X)}$$

朴素贝叶斯基于各种特征之间相互独立，在给定类别为 y 的情况下，上式可以进一步表示为：

$$P(X|Y=y)=\prod_{i=1}^{d}P(x_i|Y=y)$$

由以上两式可以计算出后验概率为：

$$P_{post}=P(Y|X)=\frac{P(Y)\prod_{i=1}^{d}P(x_i|Y)}{P(X)}$$

由于$P(X)$的大小是固定不变的，因此在比较后验概率时，只比较上式分子部分即可。因此可以得到一个样本数据属于类别y_i的朴素贝叶斯计算如下：

$$P(y_i|x_1,x_2,\cdots,x_d)=\frac{P(y_i)\prod_{j=1}^{d}P(x_j|y_i)}{\prod_{j=1}^{d}P(x_j)}$$

通过对分类样本建立朴素贝叶斯模型，运用上述训练样本通过朴素贝叶斯模型进行训练得到分类模型$model_sz1$，运用训练模型对测试样本进行检验，输出其分类结果，具体代码如下图7：

```
#朴素贝叶斯模型
from sklearn.naive_bayes import ComplementNB
model1 = ComplementNB()
model_sz1=model1.fit(X_tr, labels_tr)
pre1 = model_sz1.predict(data_te)
print(pre1)
print('------------------------------')
```

图7

(2) 支持向量机分类模型

支持向量机（SVM）是一组相关的监督学习的方法， 是一种常用的文本分类方法，最初是针对二分类问题提出的，但在实际生活中多分类问题用的更为普遍。支持向量机的多分类方法主要是两种：一种是将多分类问题分解成多个二分类问题，然后再将多个二分类器的输出组合在一起实现多分类；另一种就是将多个分类面的参数求解合并到一个最优化问题中，最终通过求解该最优化问题实现多分类，本文用的到正是这种。支持向量机分类包括以下三种方法：一是$1-a-r$方法，对于K类问题，需要构造K个将其中一个类别与其他所有类别区分开的两类支

持向量机分类器，分类时采用最大输出法将多个两类分类器的输出组合在一起实现多分类；二是树形支持向量机多分类方法，其基本思想是从根节点开始，采取某种方法将该节点所包含的类别划分为两个子类，然后对这两个子类进一步划分，循环，直到子类只包含一个类别，停止；三是决策树支持向量机多分类器，是指将支持向量机和二叉决策树结合起来构成多类别的分类器，称为SVM决策树方法。本文用到的是$LinearSVC$实现线性分类支持向量机，根据$liblinear$实现的，既可用来二分类，也可以用于多分类。$liblinear$的目标函数为：

$$\min\{\frac{1}{2}w^Tw + C\sum_{i=1}^{N}\xi(y_iw^Tx_i)\}$$

通过对分类样本建立朴素贝叶斯模型，运用上述训练样本通过朴素贝叶斯模型进行训练得到分类模型$model_sz2$，运用训练模型对测试样本进行检验，输出其分类结果，具体代码如下图8：

```
#支持向量机模型
from sklearn.svm import LinearSVC
model2 =LinearSVC()
model_sz2 = LinearSVC().fit(data_tr,labels_tr)
pre2=model_sz2 .predict(data_te)
print(pre2)
```

图8

(3) 模型的评价

$F-score$分类模型评估方法：$F-score$评估方法是综合考虑查全率（R）和查准率（P）的一个评价指标，其计算公共为：

$$F = \frac{(1+\beta^2)\cdot P\cdot R}{\beta^2\cdot(P+R)}$$

当$\beta = 1$时，查全率和查准率同样重要，即为：

$$F = \frac{2\cdot P\cdot R}{P+R}$$

模型评价：通过$python$计算出朴素贝叶斯模型和支持向量机模型的$F1$值如下：

表2：朴素贝叶斯$F1$值

Label	0	1	2	3	4	5	6	average

F1	0.83	0.90	0.68	0.89	0.88	0.76	0.86	0.828

表3：支持向量机$F1$值

Label	0	1	2	3	4	5	6	average
F1	0.82	0.89	0.77	0.90	0.90	0.78	0.88	0.848

由数据可以得出，朴素贝叶斯的$F1 = 0.828$小于支持向量机的$F1 = 0.848$，所以支持向量机的分类效果要优于朴素贝叶斯。

注：问题一具体代码见附件文件$C_1.py$。

3.2 问题二：基于群众留言的归类及热点问题的挖掘

3.2.1 层次聚类算法原理

文本的分类在广义上有两种含义：一种是有指导的学习过程，被称为分类；另一种是指无指导的学习过程，被称为聚类。本文问题一中就使用了常见的几种分类算法：朴素贝叶斯($naïve\ bayes$)，支持向量机(SVM)。而问题二是在不知道分类标签的情况下，对其聚类，也就是将文本集合分成多个类或簇，使得在同一个簇中的文本内容具有较高的相似度，而不同的簇中文本内容差别较大。

平面聚类虽然高效简单，但是它需要预先输入指定的簇数目，然而这个簇数目是未知的，$K-means$算法就属于这一类，$K-means$算法是给定簇的个数K，选定K个文本分别作为K个初始簇，将其他的文本加入最近的簇中，并更新簇的中心点，然后再根据新的中心点对文本重新进行划分，直到簇不再变化时，算法停止。$K-means$算法的容易实现，复杂度低，但对噪声文本敏感，并且在通常情况下无法确定K值。因此，有了如下的层次聚类方法，它属于分层聚类，输出的是层次结构，而且不需要预先指定聚类的簇数目，这正好符合问题二的要求。

层次聚类可以分为两种：一种是自下而上的凝聚层次聚类($Agglomerative$)，就是指一开始每个个体都是一个类，然后根据$linkage$寻找同类，最后形成一个类；另一种是自上而下的划分层次聚类($Divisive$)，与前面相反，一开始所有个体都属

于一个类，然后根据$linkage$排除异己，最后每一个个体都形成一个类。$linkage$判别类的方法有：最短距离法、最长距离法、中间距离法、类平均法，其中最常用的就是类平均法。

本文使用的是层次聚类中比较新的一种算法$BIRCH(Balanced\ Iterative\ Reducing\ and\ Clustering\ using\ Hierarchies$利用层次方法的平衡迭代规约和聚类)，此算法利用了一个树结构，类似于平衡$B+$树，称为聚类特征树($Clustering\ Feature\ Tree$，以下称$CF\ Tree$)，其每一个节点都是由若干个聚类特征($Clustering\ Feature$,以下称$CF$)组成。如下图9：

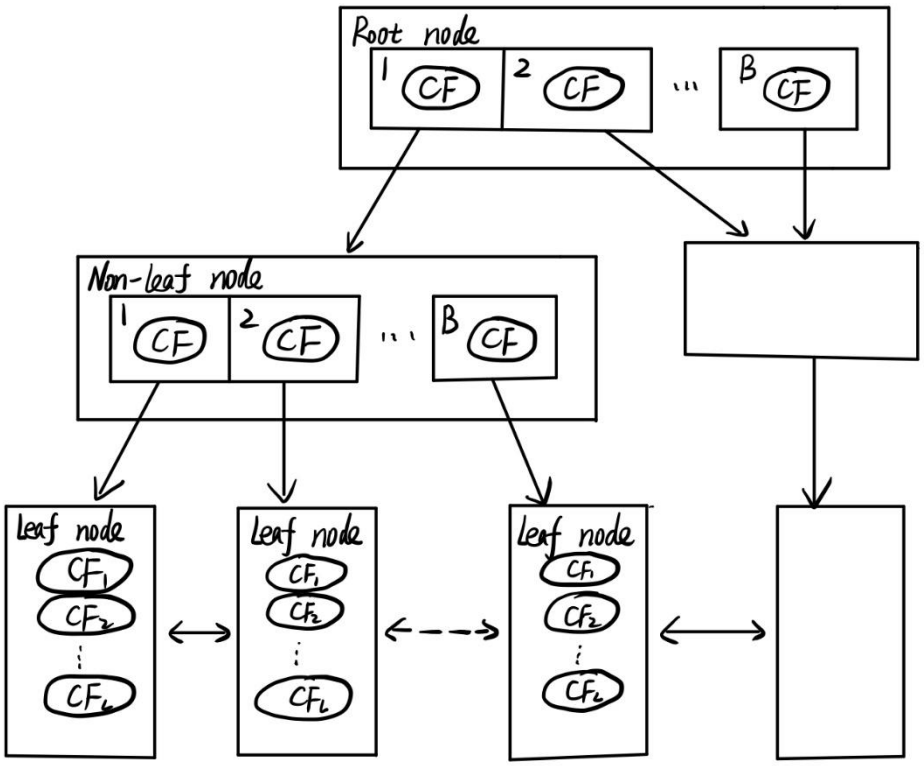

图9

在聚类特征树中，一个CF定义为一个三元组(N, LS, SS)，N代表这个CF中拥有的样本点的数量，LS代表这个CF中拥有的样本点各特征维度的和向量，SS代表这个CF中拥有的样本点各特征维度的平方和，并且CF满足线性关系。聚类特征数有

三个参数：内部节点的最大CF数B，叶子节点的最大CF数L，叶节点每个CF的最大样本半径阈值T。$BIRCH$算法的主要流程就是将所有的文本依次读入，在内存中建立一颗$CF\ Tree$，不用输入类别数K值，最后CF元组的组数即为最终的K。

3.2.2 模型建立

(1)导入数据及数据预处理

本文根据附件三中的留言主题一列对问题进行归类并找出热点问题，因此首先将附件三的留言主题导入$python$中，再对其分词操作，本文使用的是最常用来处理中文文本的$jieba$分词，代码如下图10：

```
def init_data(self):
    corpus = []
    self.title_dict = {}
    index = 0
    for line in f:
        title = line.strip()
        self.title_dict[index] = title
        seglist = jieba.cut(title, cut_all=False)
        output = ' '.join(['%s' % x for x in list(seglist)]).encode('utf-8')
        index += 1
        corpus.append(output.strip())
```

图10

(2)权值转换

本文使用的是$TF-IDF$,$TF(Term\ Frequency$,词频), $IDF(Inverse$

$Document\ Frequency$,逆向文件频率)，原理在问题 一的分析中已有阐述，其计算公式如下：

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

$n_{i,k}$是词条t_i在文件d_j中出现的次数，$\sum_k n_{k,j}$是文件d_j中所有词条出现的次数之和。

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

$|D|$是指文件总数，$|\{j:t_i\in d_j\}|$是包含词条t_i的文件数。

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

代码如下图11：

```
vectorizer = CountVectorizer()
transformer = TfidfTransformer()
tfidf = transformer.fit_transform(vectorizer.fit_transform(corpus))
word = vectorizer.get_feature_names()
self.weight = tfidf.toarray()
```

图11

(3)使用$BIRCH$算法聚类

$BIRCH$算法中对聚类效果影响最大的就是其中的参数$threshold, n_clusters$，由于并不知道类别数K，因此令$n_clusters = None$，最终经过不断的调参发现$threshold = 0.82$时聚类效果最佳，具体代码如下图12：

```
def birch_cluster(self):
    print('start cluster Birch --------------------')
    self.cluster = Birch(threshold=0.82, n_clusters=None)
    self.cluster.fit_predict(self.weight)
```

图12

3.2.3 模型求解

根据代码运行的结果，将相似度较高的归为一类，在聚类前共有4325条文本，聚类后共有3324条文本，考虑到点赞数和反对数可能存在误点的情况，本文直接将热点问题出现的频数作为热度指数，最终得到排名前五的热点问题如下图13：

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	48	2019/7/7至2019/12/731	A市伊景园滨河苑	投诉A市伊景园滨河苑捆绑车位销售
2	2	37	2019/11/15至2020/01/25	丽发新城小区附近	丽发新城小区附近的搅拌站噪音严重扰民
3	3	21	2018/11/15至2019/12/2	A市	咨询A市人才购房补贴政策
4	4	18	2019/01/04至2019/9/30	A市	请A市加快一圈二场三道建设力度
5	5	15	2019/01/17至2019/11/14	A市晟福兴汝金城二期	投诉A市晟福兴汝金城二期涉嫌虚假宣传

图13

然后再从附件三中提取出这五个热点问题的详细信息，部分结果截图如下图14（完整信息见作品附件热点问题留言明细表）：

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
1	190337	A00090519	关于伊景园滨河苑捆绑销售车位的维权投诉	2019-08-23 12:22:00	集团下发文件，强制要求职工再交	0	0
1	191001	A909171	A市伊景园滨河苑协商要求购房同时必须购买车位	2019-08-16 09:21:33	苦恼，作为一名退休职工，就攒了	1	12
1	195995	A909199	关于广铁集团铁路职工定向商品房伊景园滨河苑项目的问题	2019-08-10 18:15:16	项。也未给出首付款详细的说明，	0	0
1	196264	A00095080	投诉A市伊景园滨河苑捆绑车位销售	2019/8/7 19:52:14	国家三令五申禁止捆绑车位销售，	0	0
1	199190	A00095080	关于A市武广新城违法捆绑销售车位的投诉	2019/8/1 22:32:26	。但辛辛苦苦一辈子的老百姓房着	0	0
1	204960	A909192	家里本来就困难,还要捆绑买卖车位	2019-08-21 18:12:20	滨河苑房子又非要我们买根本用不	0	0
1	205277	A909234	伊景园滨河苑捆绑车位销售合法吗？！	2019-08-14 09:28:31	时捆绑购买12万一个的车位，不	0	1
1	205982	A909168	坚决反对伊景园滨河苑强制捆绑销售车位	2019-08-03 10:03:10	是明显的违法捆绑销售！通过购	0	2
1	207243	A909175	伊景园滨河苑强行捆绑车位销售给业主	2019-08-23 12:16:03	购买车位，不买车位就取消购买资	0	0
1	209506	A909179	A市武广新城坑客户购房金额并且捆绑销售车位	2019-08-02 16:36:23	不签合同的那种，工作人员又不说	0	0
1	209571	A909200	伊景园滨河苑项目绑定车位出售是否合法合规	2019-08-28 19:32:11	销售的车位。而单个车位高达12万	0	0
1	213584	A909172	投诉A市伊景园滨河苑定向限价商品房违规涨价	2019-07-28 13:09:08	12万/户，无视法律法规，无视民	0	0
1	214975	A909182	关于房伊景园滨河苑销售若干问题的投诉	2019-08-22 00:00:00	房子和车位一对一购买。据说车子	0	3
1	218709	A000106692	A市伊景园滨河苑捆绑销售车位	2019/8/1 22:42:21	签订正规购房合同，强制收取18	0	1
1	218739	A909184	A市伊景园·滨河苑欺诈消费者	2019-08-24 00:00:00	购房者交付车位定金，还不写入协	0	0
1	220534	A00075092	投诉武广新城伊景园滨河苑为广铁集团的定向商品房	2019-08-12 12:37:28	都买不起，要贷款，何来钱买车	0	0
1	222209	A00017171	A市伊景园滨河苑定向限价商品房项目违规捆绑销售车位	2019-08-28 10:06:03	消购房资格相要挟，逼迫职工交	0	0
1	223247	A00044759	投诉A市伊景园滨河苑捆绑销售车位	2019/7/23 17:06:03	取消车位捆绑。2.希望这个房子对	0	0
1	224767	A909176	伊景园滨河苑车位捆绑销售！广铁集团做个人吧！	2019-07-30 14:20:08	我合同，说什么预购不用！后面就	0	0
1	230554	A909174	投诉A市伊景园滨河苑捆绑车位销售	2019-08-19 10:22:44	是伤筋动骨的，购房者中有一部	0	0
1	234633	A909194	无视消费者权益的A市伊景园滨河苑车位捆绑销售行为	2019-08-20 12:34:20	职工购买房子的同时一对一购买	0	0
1	236301	A909197	和谐社会背景下的A市伊景园滨河苑车位捆绑销售	2019-08-30 16:32:12	这个还是和谐社会么？政府的	0	0
1	239032	A909169	请维护铁路职工权益取消伊景园滨河苑捆绑销售车位的要求	2019-09-01 10:03:10	同，现在还要求一户一车位，捆	0	1
1	244243	A909198	关于伊景园滨河苑捆绑销售车位的投诉	2019-08-24 18:23:12	业主共有，但属于城镇公共绿地	0	0
1	244342	A0008948	投诉A市伊景园滨河苑定向限价商品房违规涨价	2019/7/28 10:36:05	收取认购款18.5万，违法强行捆	0	0
1	244528	A909235	伊景园滨河苑开发商强买强卖!	2019-08-21 19:05:34	福利的项目，但现在却要求购房	0	2
1	246407	A00099597	举报广铁集团在伊景园滨河苑项目非法绑定车位出售	2019-09-01 14:20:22	团在项目中非法绑定高价车位出售	0	0

图14

注：热点问题表及热点问题留言明细表见作品附件文件，具体代码见附件文件$C_2.py$。

3.3 问题三：留言答复的评价方案

3.2.1 模型原理

本文主要从相关性角度建立评价体系，主要思想是提取留言详情与答复意见，对其进行分词等预处理操作后，提取关键词，转化为向量，使用余弦相似度计算群众留言详情与答复意见之间的夹角，这是使用非常广泛的计算两个向量相似度的公式，它可以去除文档长度的影响，其公式为：

$$sim(d_1,d_2)=\frac{\vec{V}(d_1)\cdot\vec{V}(d_2)}{\left|\vec{V}(d_1)\right|\left|\vec{V}(d_2)\right|}$$

其分子是两个向量的内积，分母是两个向量的欧几里得长度之积。

对于及时性，可以从答复时间与留言时间的差值上得出；可靠性可以从答复意见是否有具体依据，比如某条例或规章制度等方面得出结论；至于完整性及可解释性，本文未对其进行过多分析。

3.2.2 模型建立

(1)数据读取及分词

读入附件四(作品附件中的)，将其留言详情保存至$s1$字符串中，答复意见保存至$s2$字符串中，利用$jieba$分词及停用词表$stopword$(见作品附件$stopword.txt$)对其进行预处理，将分好的词保存至向量中，代码如下图15：

```
data = pd.read_excel('./附件4.xlsx')
res=[]
for j in range(2816):
    s1=''.join(data[data['label']==j]['留言详情'])
    s2=''.join(data[data['label']==j]['答复意见'])
    stopwords = []
    fstop = open('stopword.txt', 'r', encoding='gbk')
    for eachWord in fstop:
        eachWord = re.sub("\n", "", eachWord)
        stopwords.append(eachWord)
    fstop.close()
    stopwords = [' ', '<', '>', '≠', '≮', '会', '月', '日', '-', ',', '，'] + list(stopwords)
    s1_cut = [i for i in jieba.cut(s1, cut_all=True) if (i not in stopwords) and i != '']
    s2_cut = [i for i in jieba.cut(s2, cut_all=True) if (i not in stopwords) and i != '']
    word_set =set(s1_cut).union(set(s2_cut))
```

图15

(2)数据保存及形成向量

将分词后的所有词加上编号后保存至字典，并根据词袋模型统计每个词出现的次数，形成向量，代码如下图16：

```
word_dict = dict()
i = 0
for word in word_set:
    word_dict[word] = i
    i += 1
s1_cut_code = [0] * len(word_dict)
for word in s1_cut:
    s1_cut_code[word_dict[word]] += 1
s2_cut_code = [0] * len(word_dict)
for word in s2_cut:
    s2_cut_code[word_dict[word]] += 1
```

图16

(3)余弦相似度计算

代码如下图17：

```
sum = 0
sq1 = 0
sq2 = 0
for i in range(len(s1_cut_code)):
    sum += s1_cut_code[i] * s2_cut_code[i]
    sq1 += pow(s1_cut_code[i], 2)
    sq2 += pow(s2_cut_code[i], 2)
```

图17

3.2.3 模型求解

根据代码运行的结果，计算出每一行对应的答复意见与留言详情之间的相似度，发现其结果位于(0,1)，因此本文定义其评价指标为相似度大小，其相似度越接近于1，则表明答复意见与留言详情越相关；反之，越接近于0，则表明答复意见与留言详情相关性越低。部分相似度计算结果如下图18：

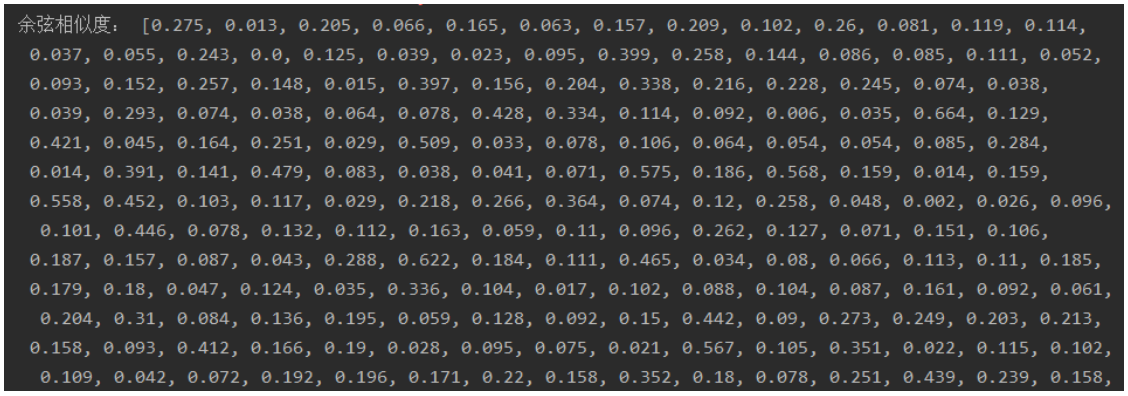

图18

此外，及时性通过答复时间与留言时间之间的间隔衡量，间隔时间越短，则越及时，说明办事效率越高；可靠性方面，本文从其答复意见是否有据可以来判断，若是，则认为其可靠性较高。至此，本文建立了一套融合相关性、及时性、可靠性的一套评价体系。

注：具体代码见作品附件$C_3.py$。

4 模型评价及推广

4.1 模型的优点

(1) 利用$python$软件对数据进行处理，并利用$jieba$词库进行分词，简便，快捷。

(2) 本文建立的模型与实际紧密联系，实用性强，可信度高。

(3) 本文建立的模型考虑充分，运用数据全面，从而减小模型误差。

4.2 模型的缺点

(1) 运用的$jieba$库，词库有限，未考虑地名、人名、情感词等情况，具有一定的误差。

(2) 模型直接采用多分类模型，使得模型分类有一定的误差。

4.3 模型的推广

分析所建模型，不难发现不仅可以运用在客户留言中，在其他方面也有许多运用，例如可以用于电商产品评价中，对评价信息进行分类，是好评、差评还是中评，还可以对商家回复信息的进行评价，是否相关，完整等。除此之外，该模型还在垃圾短信分类等其他方面也有许多的运用。

5 参考文献

[1] 温珍.基于$Python$语言的中文文本处理研究[J].南昌工程学院学报,2018,(第 3 期)：70-75.

[2] 韦文娟,韩家新,夏海洋.基于$Python$自然语言处理的文本分类研究[J].福建电脑,2016,(第 7 期)：4-5，8.

[3] 石凤贵.基于$TF-IDF$中文文本分类实现[J].现代计算机,2020,(第 6 期)： 51-

54，75.

[4] 叶雪梅,毛雪岷,夏锦春,王波.文本分类$TF-IDF$算法的改进研究[J].计算机工程与应用,2019,(第 2 期). 104-109，161 .

[5] 王阳,周云才.朴素贝叶斯分类算法的设计与分析[J].电脑知识与技术,2019,(第 11 期)：206-208.

[6] 章永来,周耀鉴.聚类算法综述[J].计算机应用,2019,(第 7 期): 1869-1882.

[7] 武永亮,赵书良,李长镜,魏娜娣,王子晏.基于$TF-IDF$和余弦相似度的文本分类方法[J].中文信息学报,2017,(第 5 期): 138-145.

[8] 郑捷. 机器学习算法原理与编程实践[M]. 北京：电子工业出版社, 2015.10.

[9] 弗朗索瓦·肖莱著；张亮（$hysic$）译. $Python$深度学习 人工智能 $deep\ learning\ with\ python$ 中文版[M]. 北京：人民邮电出版社, 2018.08.