2020年第八届

“泰迪杯”

数据挖掘挑战赛

选择题目：“智慧政务”中的文本挖掘应用（C）

“智慧政务”中的文本挖掘应用

摘　要

针对群众留言的分类，剔除无关数据，对文本进行去重、去特殊字符串、结巴分词、去停用词处理，得到每一留言所包含的实词，构建$TF-IDF$向量，用文档频次方法过进行特征选择，采用高斯朴素贝叶斯算法进行文本分类，用$F-Score$评价分类方法。

热点问题挖掘中，我们主要使用以余弦相似度作为距离的层次聚类算法，通过每个留言的留言主题分类，我们可以得到相似的留言内容，这样我们就能通过这些分类得到热点问题，并且我们将某类热度指标定义为该类留言的反对数+点赞数+该类所含留言的个数，并且得出结果以后，我们通过 Excel 对它进行人为的筛选操作，最终得到 2428 个分类，按题目要求给出‘热点问题表.xls’和‘热点问题留言明细表.xls’。

对于答复意见的评价，用 0 到 100 的分值来衡量答复的总体质量，答复的总体质量又分为相关性、完整性、可解释性和及时性四个指标，然后划分指标占总体的权重，运用了 LSI 算法计算相关性的评价值，根据答复到留言间隔的工作日天数构造分段函数计算及时性评价值，完整性和可解释性用匹配词语的方法定性地给出评价值，最后四个指标相加得到总体的答复质量评价值。

关键词：TF-IDF；朴素贝叶斯算法；PCA；余弦相似度；LSI；层次聚类

1.背景与挖掘目标

1.1 问题背景

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。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，对提升政府的管理水平和施政效率具有极大的推动作用。

1.2 挖掘目标

本次数据挖掘的目的是根据题目提供的留言分类标签、留言、留言及相应的答复等数据，深入分析数据，利用一系列的方法达到以下目标：

问题 1：群众留言分类

为了解决人工分类群众留言存在工作量大，效率低，且差错率高等问题，根据题目提供的数据，利用文本分类算法实现群众留言的自动分类，给每一条留言贴上一类标签，并且使用 F-Score 对分类方法进行评价

问题 2：热点问题挖掘

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。请根据附件 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。

问题 3：答复意见的评价

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。

2.挖掘过程概述

本文数据挖掘总流程图如下图 1 数据挖掘流程

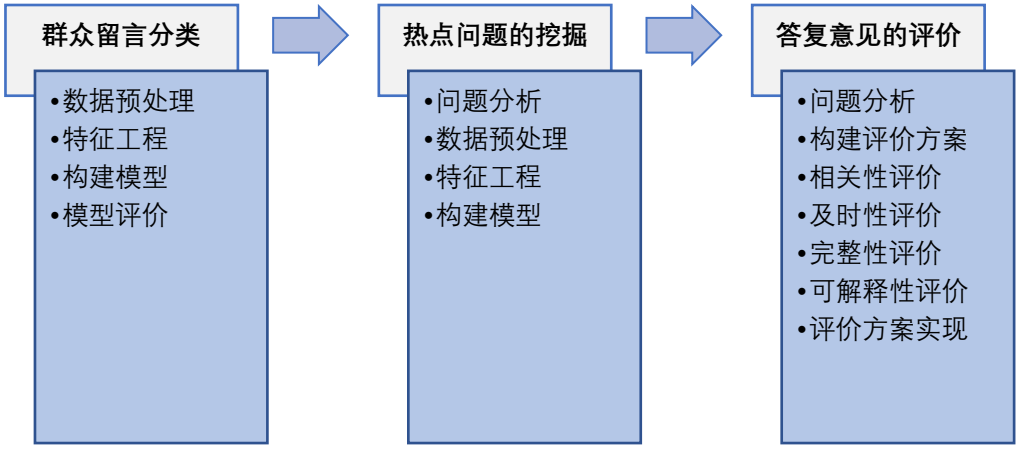

图 1 数据挖掘流程图

2.1 群众留言分类

对题目数据提供的附件 2，读取相关的留言数据及相应的标签，对留言数据做预处理，包括文本去重、去特殊字符串、结巴分词、去停用词，得到每条留言出现的实义词数据，根据这些数据构建$TF-IDF$权值向量矩阵，采用文档频次方法选择出较为重要的特征，用朴素贝叶斯分类器对留言分类。

2.2 热点问题挖掘

对附件 3 的数据先做数据预处理，其中包括数据清洗、增加属性、分词处理、去除停用词，再通过对“留言主题”的分词结果构建$TF-IDF$权值向量矩阵，并通过主成分分析法（PCA）对其进行降维处理，然后使用层次聚类算法对留言进行分类。某一类问题的热度指标我们定义为该类留言的所有“反对数”+“点赞数”+“该类所含个数”，最后根据热度指标的排序，得到排名前五的热点问题。

2.3 答复意见的评价

根据附件 4 给出的数据，对答复意见的评价分为相关性、及时性、完整性、可解释性四个角度，以 100 分为总分划分出这四部分的权重，再分别根据各自的情况来设计各部分的评价规则，得到数据中每条答复意见的四个角度的评分，最后把这四个角度的评分相加得到评价总分，从而看出每条答复意见质量的好坏。

3. 群众留言分类

3.1 数据预处理

数据预处理流程如下：

剔除无关数据：根据挖掘目标，剔除原始数据附件二中与挖掘主题无关的数据，删除本次实验没有用到的“留言编号”、“留言用户”、“留言时间”，保留三个有效属性：“留言主题”、“留言详情”、“一级标签”，也可以有选择性地读取要保留属性对应的列。

去除重复的留言：若数据中存在多条留言详情一模一样的留言，则视为一条留言，把重复的留言删除掉，使得所有留言的留言详情只出现一次。

去除空白字符：可以看到有的留言中带有大量的空白字符，即空格，制表符，换行符等，这对挖掘目标的实现毫无帮助，且留言开头和结尾大量的空白字符不便于我们处理时的阅读，还会因为编码格式的问题出现乱码，不利于后续的处理，可调用 python 相应的方法以及利用正则表达式去除空白字符。

去除特殊字符串：留言可能会存在银行账户，电话，qq，价格，日期，网址等字母数字的字符串，对文本的识别没有太大的帮助，我们通过正则表达式去除这些字符串。

分词处理：利用 Python 的 jieba 库将汉语文本字符串切分成合理的词语序列，再建立行业词典，使得一些完整的专业词汇不会被切分开，完善文本切分的效果。

去除停用词：建立停用词表，词表包括虚词、对文本的识别不大的词、标点等，过滤掉这些停用词，尽可能地使得剩下的词有识别度，能够反映该留言的类型，从而作为留言的特征。

留言主题和详情合并：数据中留言分为留言主题和留言详情两部分（下面如果不作特殊说明，则默认留言指的是合并后的留言主题和留言详情），为方便后续的处理，这两部分分别进行上述预处理后把留言主题的分词和留言详情的分词合并在一起。考虑到留言主题具有概括留言全文的作用，它相对于留言详情更能反映留言的类型，更具有代表性，我们通过提高留言主题分词的频数来提高主题分词在所有分词的权重，这里把留言主题里的分词复制三份然后和留言详情的分词合并在一起。

部分数据预处理结果如下表 3-1 数据预处理结果及对应标签，全部数据见“预

处理后.csv”，相关代码见“data_process.py”。

表 3-1 数据预处理结果及对应标签

一级标签	文本预处理后的词串
城乡建设	大道 西行 道 未管 路口 加油站 路段 ……
城乡建设	位于 书院 路 主干道 在水一方 大厦 ……
城乡建设	区苑 小区 位于 火炬 路 小区 物业 ……
……	……

3.2 特征工程

3.2.1 文本表示

在分词之后，通过统计每个词在该条留言出现的次数，就可以得到该留言基于词频的特征，即可用词频向量表示出文本，实现文本向量化。考虑到某个词在某留言中出现很多次但其实它在很多其他的留言中也出现了很多次，那么这个词并不能区分它所在的留言是什么类型，也就是说没有起到辨识的作用，而且各留言的长度篇幅不同，仅仅用词语出现的频数不能进行比较，要标准化处理，所以我们要找某一留言内单词频率高的词，且它在所有的留言里出现过的频率很少的词，通过计算词的$TF-IDF$值就能表示该词对留言的重要性。我们进一步把词频向量转化为基于$TF-IDF$值为特征的向量。可以通过 Python 的 sklearn 库的 CountVectorizer 类、TfidfTransformer 类来直接实现文本$TF-IDF$权值向量的表示。

$TF-IDF$算法步骤：

Step1：统计每个词在该留言的频数，构建每一留言的词频向量，向量的维数是语料中所有出现过的词的种数；

Step2：计算每一词语的词频TF，即对词语的频数做标准化处理，即：

$$TF = \frac{某个词在留言中的出现次数}{留言的总词数}.$$

Step3：计算每一词的逆文档频率IDF：

$$IDF = \log\left(\frac{留言总数}{包含该词的留言数目+1}\right).$$

Step4：计算$TF-IDF$：

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

因为文本向量的维数很大，大多数的词有的留言都没有出现，最后得以留言为行、特征为列的$TF-IDF$权值矩阵是稀疏矩阵。

3.2．2 文本特征选择

本文采取文档频数方法（DF）进行文本特征选择，对每一特征计算它的文档频次DF，即包含该词的留言数目除以留言的总数，设定一个最小阈值$DFmin$和一个最大阈值$DFmax$，若特征的文档频次DF小于最小阈值或大于最大阈值，则把该特征删除，因为若文档频次过小则说明该特征没有代表性，太少不足以对分类产生影响，若文档频次过大则说明该特征太多没有区分度，根据给定的阈值过滤无用的特征从而实现特征降维。

这里设定最小阈值$DFmin$为 0.004，大阈值$DFmax$为 0.1，降维后维度从 67586 变成 2624。

3.3 构建模型

文本分类模型选择高斯朴素贝叶斯，我们假设各特征之间互相独立，且各特征的值服从正态分布.

设留言是第j类标签的先验概率为：

$$P(Y_j).$$

第j类标签的留言有特征$X_i\ (i=1,2,\dots,n)$取值为x_i的概率为：

$$P(X_i=x_i|Y_j).$$

其中x_i为特征X_i可能的取值，n为特征的总数量；

第j类标签的留言若干特征$X_1,\dots,X_n$各取值为x_i的概率为：

$$P(X_1=x_1,\dots,X_n=x_n|Y_j).$$

留言若干特征$X_1,\dots,X_n$的各取值为x_i先验概率为：

$$P(X_1=x_1,\dots,X_n=x_n).$$

若干特征$X_1,\dots,X_n$各取值为x_i时是第j类标签的概率为：

$$P(Y_j|X_1=x_1,\dots,X_n=x_n).$$

根据贝叶斯定理有如下关系：

$$P(Y_j|X_1 = x_1, \dots, X_n = x_n) = \frac{P(Y_j)P(X_1 = x_1, \dots, X_n = x_n|Y_j)}{P(X_1=x_1,\dots,X_n=x_n)}.$$

因为我们的目标是给留言贴上标签，后验概率最大的那个标签就可以认为该留言的标签，而用上述式子计算具体的后验概率不可行也没必要，只需找出在若干特征$X_1,\dots,X_n$发生时哪类标签的后验概率最大即可，因为分母都一样，所以只需关注分子部分，再利用各特征之间互相独立的假设，原本对贝叶斯公式的求解转变为求朴素贝叶斯表达式的分子部分：

$$P(Y_j)\prod_i^n P(X_i = x_i|Y_j).$$

这部分越大说明对应的Y_j标签的后验概率越大，就越可能是该留言的标签。

朴素贝叶斯算法步骤：

Step1：计算训练集所有标签的先验概率$P(Y_j)$；

$$某类标签的先验概率 = \frac{该类标签的留言数}{留言总数}.$$

Step2：计算训练集所有特征在每一标签Y_j下的方差σ_{ij}^2和均值μ_{ij}；

Step3：计算测试集的每条被分类的留言所有标签对应的朴素贝叶斯表达式分子部分，因为特征是连续型变量,所以其中

$$P(X_i = x_i|Y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}}e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}.$$

代入上述所说的朴素贝叶斯表达式的分子部分，即：

$$P(Y_j)\prod_i^n \frac{1}{\sqrt{2\pi\sigma_{ij}^2}}e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}.$$

Step4:比较上一步所计算朴素贝叶斯表达式分子部分的大小，得出留言的标签。

使用高斯朴素贝叶斯分类器对留言分类可通过 Python 的 sklearn 库的 GaussianNB 实现。部分留言分类结果如下表 3-2 留言分类结果及对应实际标签，全部数据见“分类结果.csv”,相关代码见“model.py”。

表 3-2 留言分类结果及对应实际标签

预测标签	实际标签	留言主题	留言详情
城乡建设	商贸旅游	K8 县九嶷……	各位领导：……
城乡建设	城乡建设	A5 区红花……	肖局长：……
劳动和社会保障	劳动和社会保障	M 市医疗报……	厅长：……
……	……	……	……

3.4 模型评价

朴素贝叶斯基于特征之间独立的假设，使得计算概率大大的简化，节省内存和时间，算法也比较简单，但是对于特征之间关联程度高的数据效果很差。把附件 2 的数据划分为 80%的训练集，20%的测试集，用训练好的朴素贝叶斯分类器对测试集的留言分类，F-Score 为 0.7409。

4. 热点问题挖掘

4.1 问题分析

为了更高效地解决居民反映的问题，有关部门根据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，并且把某一时段内群众集中反映的某一问题称为热点问题，我们需要制定一个热度评价指标去判断这一类问题是不是一个热点问题，并且要求我们给出排名前五的热点问题。

我们发现附件 3 的群众留言是没有经过归类处理的，所以我们首先需要对这些群众留言进行分类，将 4000 多条留言分类后就开始热度评价指标的制定。附件 3 的数据中含有 7 个属性，经过我们在网上查询的相关资料发现附件中对热度评价指标有贡献的属性有“反对数”与“点赞数”，通过对两个属性叠加我们可以创造一个新的属性：“关注度”，而且能够知道该条留言有多少人在关注，再将每一类留言所含的个数以及该类留言中所有的关注度相加就可以得到一个数值，我们称之为该类留言的频数，并以此作为热度评价指标，排名前五的就是我们要的结果。

4.2 数据预处理

4.2.1 数据清洗

对附件 3 中的数据进行去重操作，在去重时，我们对附件 3 的“留言详情”进行去重操作，因为“留言详情”是不可能完全一样的，经过处理后附件三去重后还剩下 4221 条留言。由于“留言主题”、“留言详情”中的开头、结尾含有许多空格字符会影响后面的操作，所以我们需要去除这些空格字符。

4.2.2 增加属性

为了热度评价指标我们需要在数据中添加新的属性:“关注度”，它是由每一条留言的“反对数”和“点赞数”相加所得，清洗后的数据见‘数据清洗.csv’。

4.2.3 分词处理

为了能够对所有的留言进行分类处理,选取清洗过后的数据中的“留言主题”,并且使用 Python 开发的一个中文分词库 jieba 对“留言主题”做分词处理，由于“留言主题”的内容是“留言详情”的一个概括，于是在这里我们认为只需要对“留言主题”进行分类即可，分词后的数据见‘cutWords.txt’。

4.2.4 去除停用词

停用词是指那些功能普遍，与其他词相比没有什么实际含义的词，它们通常是一些单字、特殊字符、字母、高频词汇，比如语气助词、副词、介词、连接词等，去除这些无关的词汇，可以极大地提升文本分类的效果，停用词表见‘stopWords.txt’。

数据预处理的代码见‘fenci.py’。

4.3 特征工程

4.3.1 文本表示

这里采用与上文相同的$TF-IDF$算法对去除停用词后的分词做文本表示，以便让计算机能够识别这些文本。

4.3.2 特征选择

由于$TF-IDF$算法得到的权值矩阵是高维的稀疏矩阵，需要耗费大量的时间取运行，我们这里采用主成分分析法（PCA）对得到的权值矩阵进行降维，增加我们的运行效率。

PCA的基本原理就是将一个矩阵中的样本数据投影到一个新的空间去。对

一个矩阵来说，将其对角化即产生特征根及特征向量的过程，也是将其在标准正交基上投影的过程，而特征值对应的即为该特征向量方向上的投影长度，因此该方向上携带的原有数据的信息越多。

PCA算法步骤：

Step1：将原始数据按行排列组成矩阵X；

Step2：对X进行数据标准化，使其均值变为零；

Step3：求X的协方差矩阵C；

Step4：将特征向量按特征值由大到小排列，取k 个按行组成矩阵P；

Step5：通过计算$Y = PX$，得到降维后的数据Y；

这里我们直接调用 Python 第三方库$sklearn$中的PCA函数实现权值矩阵的降维。降维处理见代码‘fenci.py’的最后一部分。

4.4 构建模型

4.4.1 余弦相似度

我们采用层次聚类的方法将群众的留言进行分类，得到我们的分类结果。首先我们需要先刻画文本之间的距离，这里我们采用余弦相似度作为两个文本之间的距离，因为相比欧氏距离度量，余弦相似度更加注重两个向量在方向上的差异，而非距离或长度上的差异。

余弦相似度 ：

A 和 B为两个向量属性，其余弦相似度θ由点积和向量长度给出

A_i 和 B_i分别代表向量A和B的各分量

$$similarity = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}.$$

余弦值的范围在[-1,1]之间，值越趋近于 1，代表两个向量方向越接近；越趋近于-1，代表他们的方向越相反。为了方便聚类分析，我们将余弦值做归一化处理，将其转换到[0,1]之间，并且值越小距离越近。

4.4.2 内部衡量指标

接下来引入常用的聚类性能度量内部指标：DB指数、$Dumn$指数来度量分类的效果，DB指数的计算方法是任意两个簇内样本的平均距离之和除以两个簇的

中心点距离，并取最大值，DBI的值越小，意味着簇内距离越小，同时簇间的距离越大；$Dumn$指数的计算方法是任意两个簇的最近样本间的距离除以簇内样本的最远距离的最大值，并取最小值，DI的值越大，意味着簇间距离大而簇内距离小。因此DBI的值越小，同时DI的值越大，意味着聚类的效果越好。

考虑聚类结果的簇划分$C = \{C_1, C_2, \cdots, C_k\}$，定义

$$avg(C) = \frac{2}{|C|(|C|-1)} \sum_{1 \le i \le j \le |C|} dist(x_i, x_j).$$

$$diam(C) = max_{1 \le i \le j \le |C|} dist(x_i, x_j).$$

$$d_{min}(C_i, C_j) = min_{x_i \in C_i, x_j \in C_j} dist(x_i, x_j).$$

$$d_{cen}(C_i, C_j) = dist(\mu_i, \mu_j).$$

μ表示聚类簇C的中心点位置；

$avg(C)$表示某一聚类簇内部样本点距离的均值；

$diam(C)$表示聚类簇C中样本间的最大距离；

$d_{min}(C_i, C_j)$表示聚类簇C_i与C_j间的最小样本距离；

$d_{cen}(C_i, C_j)$对应于聚类簇C_i与C_j中心点之间的距离；

DB指数：

$$DBI = \frac{1}{k} \sum_{i=1}^{k} max_{j \ne i} \left(\frac{avg(C_i) + avg(C_j)}{d_{cen}(C_i, C_j)} \right).$$

$Dumn$指数：

$$DI = min_{1 \le i \le k} \left\{ min_{j \ne i} \left(\frac{d_{min}(C_i, C_j)}{max_{1 \le l \le k} diam(C_l)} \right) \right\}.$$

4.4.3 层次聚类模型

层次聚类算法试图在不同层次对数据集进行划分，从而形成树形的聚类结构。数据集划分可采用“自底向上”的聚合策略，也可采用“自顶向下”的分拆策略。这里我们采用“自底向上”的策略，先将数据集中的每个样本看作一个初始聚类簇，然后找出两个聚类最近的两个簇进行合并，不断重复该步骤，直到达到预设的聚类个数或某种条件。关键是如何计算两个簇之间的距离，每个簇都是一个集合，因此需要计算集合的某种距离即可。下面给出三种计算方法：

给定簇C_i, C_j

$$最小距离：d_{min}(C_i, C_j) = min_{x \in C_i, z \in C_j} dist(x, z)$$

$$最大距离：d_{max}(C_i, C_j) = min_{x \in C_i, z \in C_j} dist(x, z)$$

$$平均距离：d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{z \in C_j} dist(x, z)$$

最小距离由两个簇的最近样本决定，最大距离由两个簇的最远样本决定，平均距离由两个簇的所有样本决定。

在模型中将分类个数k以及两个簇之间的距离d作为调试的阈值，其中$k \in [0, k]$，$d \in [0,1]$，通过调试它们的数值得到每个分类结果的DBI值和DI值，以此得到好的分类效果。

4.4.4 热度评价指标

我们将热度指标定义为：

X_i：第i类的热度指标；a_i：第i类的所有点赞数之和b_i：第i的所有反对数的之和；c_i：第i类的留言个数之和

$$X_i = a_i + b_i + c_i$$

4.5 测试集训练

在‘数据清洗.csv’中任取 500 条 “留言主题”，将其放进层次聚类模型中训练，得到数据如下图 2：

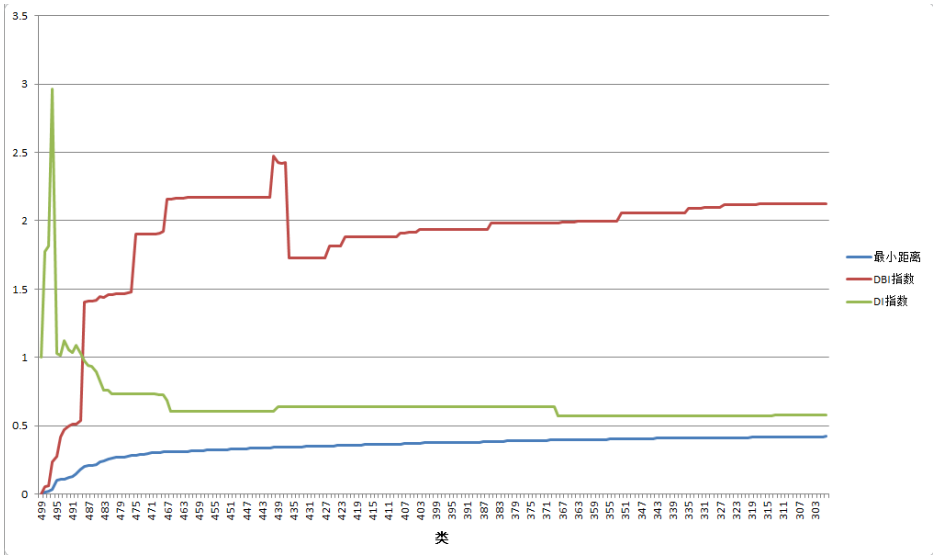

图 2

由于DBI的值越小，同时DI的值越大，意味着聚类的效果越好，结合上图发现最

优值可能出现在436到427之间，根据我们的观察，认为划分为430类比较合理，并且可以观测出此时$r = 0.35$，在后面的求解过程中我们将两个簇之间的距离d设为0.35，并且为了提高效率，我们不对测试集聚类的每一个结果求DB指数和$Dumn$指数，并将其设置为0。

4.6 模型求解

将两个簇之间的距离d设为0.35，我们把4221条数据代入，得到‘聚类结果.txt’，我们一共得到了2432个簇，运行部分结果如图3。

2453.0	0.34785954043307254	0.0	0.0
2452.0	0.3479098378802761	0.0	0.0
2451.0	0.34796819462760004	0.0	0.0
2450.0	0.34799393232284126	0.0	0.0
2449.0	0.34809384021220036	0.0	0.0
2448.0	0.34818290531570856	0.0	0.0
2447.0	0.34835678175893137	0.0	0.0
2446.0	0.3483987845234211	0.0	0.0
2445.0	0.34841156141624763	0.0	0.0
2444.0	0.3484499310241591	0.0	0.0
2443.0	0.34846348206404526	0.0	0.0
2442.0	0.34848542931356374	0.0	0.0
2441.0	0.3485964760843897	0.0	0.0
2440.0	0.34904819107326546	0.0	0.0
2439.0	0.3490870927646579	0.0	0.0
2438.0	0.34909572894333096	0.0	0.0
2437.0	0.3491200702329394	0.0	0.0
2436.0	0.34917730847590234	0.0	0.0
2435.0	0.34944745618008827	0.0	0.0
2434.0	0.3496033736407551	0.0	0.0
2433.0	0.3496043247296121	0.0	0.0
2432.0	0.35001263639842917	0.0	0.0

图3

通过问题2的代码我们就可以得到一个分类结果，并且经过我们使用excel的查找功能，使得分类的结果更为准确。关于人工查找方面我们是先得到了最初的一个结果‘各簇指标.xls’，展示的内容是2432个簇分别包含了哪些留言，并给出了热度指标。通过excel对热度指标的排序我们可以得到初步的结果，接下来我们对排名前几的一些热度问题进行筛查，由于后面的每类所包含的留言个数较少（多是1），并且它们的关注度也基本为0，所以只需筛查前面的即可。做法

是在原本排名前几的簇中查找它的具体留言，然后根据这些留言所提取的关键词去查找整个数据中是否还有类似的留言存在，如果有则进行添加，并且在添加时还需要对添加进去的簇进行删除，最后得到最终结果‘各簇指标人工调整.xls’，并按题目中要求输出结果‘热点问题表.xls’和‘热点问题留言明细表.xls’。

详细操作请看附件中问题二代码的‘使用说明.txt’。

5.答复意见的评价

5.1 问题分析

附件 4 里有我们评价的对象“答复意见”一列的数据，单从这一列的数据能够看出答复的完整性，即是否满足某种规范，还有答复的可解释性，即答复意见中的内容有没有相关解释，而与答复相对应的留言和答复联系起来可以看出答复的相关性，即答复的意见是否和问题相关，还有从留言时间和答复时间的间隔时长可以看出答复的及时性，即是否有效率、是否及时。我们可以从相关性、及时性、完整性、可解释性四个角度看出答复的质量，而这四个指标在一个答复里有不同的重要性，需要给它们划分权重比例，而且评价的规则各不相同，需要分部制定评价规则，最后这四个指标的评价汇总在一起就反映出答复意见的质量。

5.2 构建评价方案

我们把评价的总分设为 100 分，从相关性、及时性、完整性、可解释性四个角度评价答复意见，根据各自的重要程度，人为划分出这四个指标的权重，这里总分 100 分相关性占 30%，及时性占 25%，完整性占 25%，可解释性占 20%，如下表 5-1 的四个指标的权重。

表 5-1 四个指标的权重

指标	相关性	及时性	完整性	可解释性
权重	30	25	25	20

四个指标用不同的评价规则给出相应得分，最后把这些得分相加的出总分，用于衡量答复意见总体的质量。

5.3 相关性评价

5.3.1 相关性做法

我们想通过每条留言的 “答复意见”与“留言详情”文本相似度来评价相关性，然后将结果映射到[0,30]上，这样我们就得到了相关性的评价指标。

5.3.2 潜在语义索引（LSI）

相似度的计算我们使用 Python 自带的 LSI 模型进行处理，LSI 又称 LSA，它和传统向量空间模型一样使用向量来表示词和文档，并通过向量间的关系(如夹角)来判断词及文档间的关系;不同的是,LSA 将词和文档映射到潜在语义空间，从而去除了原始向量空间中的一些“噪音”，提高了信息检索的精确度。

LSI 算法原理：

通过对大量的文本集进行统计分析，从中提取出词语的上下文使用含义。技术上通过 SVD 分解等处理，消除了同义词、多义词的影响，提高了后续处理的精度。

流程：

（1）分析文档集合，建立词汇-文本矩阵 A;

（2）对词汇-文本矩阵进行奇异值分解;

（3）对 SVD 分解后的矩阵进行降维;

（4）使用降维后的矩阵构建潜在语义空间.

5.3.3 模型求解

对附件 4 的数据进行数据清洗，并通过调用 python 的 gensim 包得到每条留言中答复意见与留言详情的相似度，输出结果‘相似度.xls’，详细过程见问题三代码的‘xiangsidu.py’。

5.4 及时性评价

5.4.1 间隔工作日计算

及时性是根据从留言时间到答复时间之间的天数衡量的，考虑到有关部门的工时制度，这里的及时性根据留言时间到答复时间之间的工作日天数来衡量，即只看间隔期间星期一到星期五的天数，并作了适当简化，不考虑法定节日假期的情况，即假如某天是节假日而且又是星期一到星期五，那么仍然将该天算进工作日天数。这里计算两个日期间的工作日天数可以用 Python 的 business_calender 来实现。

5.4.2 构造及时性平均值的分段函数

因为及时性的权重占总分的 25%，所以这里设及时性指标的总分为 25。及时性指标的分数即为及时性评价值，评价值随着答复越早，分值也越高，换句话说评价值随着间隔期间的工作日天数越多，分数越低，成反比例；因为人们在不同的时间段内收到答复的满意度变化是不一样的，这里构造了一个分段函数来逼近真实情况，假设人们在三个月（大概 65 个工作日内）内收到答复满意度的变化是一样的，然后三个月到半年（大概 130 个工作日内）人们收到的满意度是一样的，半年后才收到答复算回复太迟了，那么我们设 y 为评价值，x 为间隔期间工作日天数，则有如下关系：

$$y=\begin{cases} y=25-\frac{3}{13}x\,,65>x\geq 0 \\ y=20-\frac{2}{13}x\,,130>x\geq 65 \\ 0\,,\ x\geq 130 \end{cases}.$$

函数图像如下图 3 及时性评价值的分段函数。

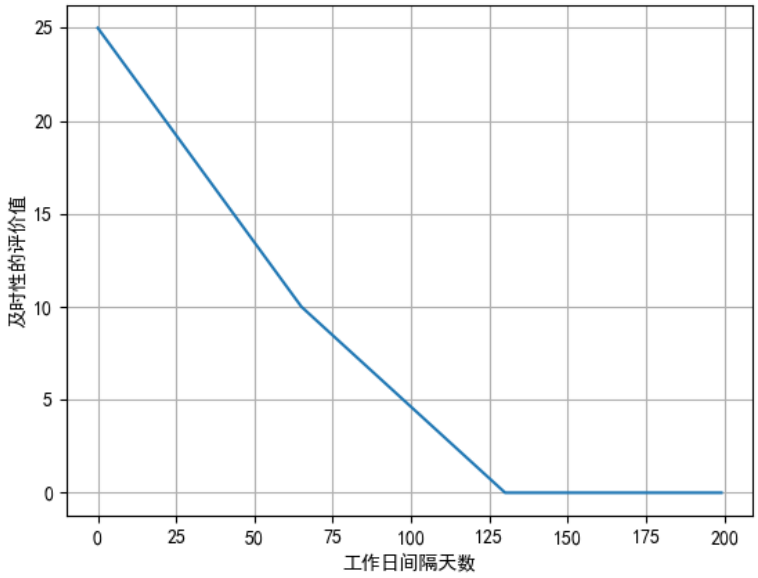

图 3 及时性评价值的分段函数

然后把答复意见的间隔留言时间的工作日天数带入函数得出及时性的评价值。

5.5 完整性评价

5.5.1 构规范词词典

评价一条答复意见的完整性，我们看它是否符合某种规范，本文讲的规范指的是答复的开头是否使用了礼貌用语，答复的中间部分是否有步骤、是否表达已作出相应的工作，答复的结尾是否有表达祝福、感谢。

根据经验建立开头规范词、中间规范词、结尾规范词三个词典，词典里包含了上述所说的规范的答复里应该存在的词或短语，并按词语出现在答复的位置分成三类，对应三个词表。

5.5.2 制定评价规则

根据完整性的在总分的权重，完整性的评价值总分为 25，按照答复是否满足开头、中间、结尾三部分的规范，再把 25 分的完整性总分细分，若答复满足开头部分的规范，则给 9 分；若答复满足中间部分的规范，则给 8 分，若答复满足结尾的规范，则给 8 分。对答复完整性的评价按要点给分，满足一部分的规范给相应的分值，最后把三部分的分值想加得到完整性评价值。

5.6 可解释性评价

答复的可解释性指答复里的有没有给留言的人一个解释，表达的内容有没有事实、理论支撑或者法律条文的依据。

可解释性的评价规则和完整性的评价规则类似，且作了进一步简化，构建一个可解释性词典，里面记录了表达事实、理论、法律条文依据会用到的词，若某答复包含了词典里任意一个词，则我们就是这条答复具有可解释性，按可解释性的权重给出可解释性评价值满分 20 分，若答复里不包含词典里的其中一个，我们就说这条答复不具有可解释性，可解释性评价值为 0 分。

5.7 评价方案实现

按照设计出来的四个指标的计算规则，分别得到四个指标的分值，四个值相加即为总体答复质量的评价值。

6.结论

本文通过分析留言的，使用 Python 对语料预处理、词向量化、特征选择和调用高斯朴素贝叶斯模型，用 sklearn 给的划分数据集函数取其中 80%用于训练模型，剩余的 20%用于测试，最后某次算出的 F-score 值为 0.7409，与人工一条一条的去分类留言的效率相比，本文的分类效果可行。

答复意见的评价方案能满足基本的衡量答复意见质量的需求。

参考文献

[1]米硕,孙瑞彬,明晓,赵汝程.基于 TF-IDF 算法的文本特征词提取模型[J].中国战略新兴产业,2017(40):113.

[2]张亚萍,胡学钢.基于 K-means 的朴素贝叶斯分类算法的研究[J].计算机技术与发展,2007(11):33-35.

[3]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,2017,31(05):138-145.

[4]周洲,侯开虎,姚洪发,张慧.基于TF-IDF及LSI模型的主观题自动评分系统研究[J].软件,2019,40(02):158-163.

[5]李金,马文超,何兵,王琭璐,杨岸宁,王颖,梁洪.基于改进文本特征的文本相似度研究[J].黑龙江大学工程学报,2018,9(01):46-52+2.

[6]张良均，陈俊德，刘名军，陈荣.数据挖掘：实用案例分析.机械工业出版社，2013.

[7]张良均，王路，谭立云，苏剑林.Python 数据分析与挖掘实战.机械工业出版社，2016.

附录清单：

附件	名称	路径
1	data_process.py	作品附件\问题一\
2	fscore.py	作品附件\问题一\
3	model.py	作品附件\问题一\
4	stopword.txt	作品附件\问题一\
5	不会被分开字典.txt	作品附件\问题一\
6	分类结果	作品附件\问题一\
7	预处理后.csv	作品附件\问题一\
8	使用说明.txt	作品附件\问题二\
9	Cluster.py	作品附件\问题二\测试结果
10	cut.txt	作品附件\问题二\测试结果
11	fenci.py	作品附件\问题二\测试结果
12	quanzhi.txt	作品附件\问题二\测试结果
13	stopWords.tx	作品附件\问题二\测试结果
14	降维.txt	作品附件\问题二\测试结果
15	聚类过程.txt	作品附件\问题二\测试结果
16	聚类结果.txt	作品附件\问题二\测试结果
17	留言主题.txt	作品附件\问题二\测试结果
18	shuchu.py	作品附件\问题二\测试结果\输出结果
19	分类结果.xls	作品附件\问题二\测试结果\输出结果
20	各簇指标.xls	作品附件\问题二\测试结果\输出结果
21	各簇指标人工调整.xls	作品附件\问题二\测试结果\输出结果
22	类 index.txt	作品附件\问题二\测试结果\输出结果
23	热点问题表.xls	作品附件\问题二\测试结果\输出结果
24	热点问题留言明细表	作品附件\问题二\测试结果\输出结果
25	fenci.py	作品附件\问题二\数据预处理
26	shujuqingxi.py	作品附件\问题二\数据预处理
27	留言主题.txt	作品附件\问题二\数据预处理
28	数据清洗.csv	作品附件\问题二\数据预处理
29	Cluster.py	作品附件\问题二\训练集
30.	cutWords.txt	作品附件\问题二\训练集
31	TF-IDF.txt	作品附件\问题二\训练集
32	聚类过程.txt	作品附件\问题二\训练集
33	聚类结果.txt	作品附件\问题二\训练集

34	性能展示.png	作品附件\问题二\训练集
35	训练集.txt	作品附件\问题二\训练集
36	答复质量评价.py	作品附件\问题三\
37	第三题答复评价结果.xls	作品附件\问题三\
38	及时性评价.py	作品附件\问题三\
39	可解释性评价.py	作品附件\问题三\
40	可解释性词典.txt	作品附件\问题三\
41	完整性评价.py	作品附件\问题三\
42	开头规范词.txt	作品附件\问题三\
43	中间范词.txt	作品附件\问题三\
44	结尾规范词.txt	作品附件\问题三\
45	xiangsidu.py	作品附件\问题三\相关性评价