“智慧政务”中的文本挖掘应用

摘要

近年来，随着一些网络问政平台逐渐成为政府了解民意、 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。因此，运用自然语言和文本挖掘技术对问政数据进行挖掘，对于提升政府的管理水平和施政效率具有重大的意义。

对于问题 1，首先进行数据清洗和分词，使用 pandas 清除脏数据，之后定义一个分词预处理函数，采用 jieba 分词工具，将处理好的文本进行分词，这里先不对停用词进行处理，因为 TF-IDF 中有相关的参数。再将文档信息转化为基于 TF-IDF 的向量，使用 TfidfVectorizer 将文档集合转为 TF-IDF 矩阵，并传入 stop_words 参数去除停用词，进而利用朴素贝叶斯算法进行分类训练，通过尝试选择 MultinomialNB，得出准确率 f1_score 的值是 0.8953706077083338。

对于问题 2，由于本题是一个无监督学习数据，没有给相应的标签，所以选择聚类模型对文本进行聚类操作。由于层次聚类想要分多少个 cluster 都可以直接得到结果，所以本文选择层次聚类来进行操作，将 DBI 作为聚类的评价标准来尝试 single 和 Average linkage 两种策略，得出最佳距离。根据问题热度评价指标构建热度评价体系，采用因子分析模型得出最终热度。

对于问题 3，本题是开放性问题，采用和第二问相同的方式找到合适的评价因子，问题答复的好坏主要由文本相似度、长度以及时效性作为评判标准，之后构建因子分析模型得出答复文本评价指标并保存。

关键词：TF-IDF;朴素贝叶斯；层次聚类；因子分析

目录

1、挖掘目标

本文建模目标是利用互联网公开来源的群众问政留言记录，及相关部门对部分群众留言的答复意见的数据，分别利用 jieba 中文分词工具、朴素贝叶斯算法和层次聚类方法进行分词、分类和聚类，达到以下三个目标：

1）利用文本分词和朴素贝叶斯算法对非结构化的数据进行文本挖掘，基于 TF-IDF 权重法对留言详情进行一级标签分类。

2）根据留言主题以及问题描述的数据，定义合理的热度评价指标，并给出评价结果。

3）根据研究的目前相关部门对热点问题留言结果，给答复意见的质量提出一套评价方案。

2、分析方法与过程

本文总体流程图如图 2-1 所示

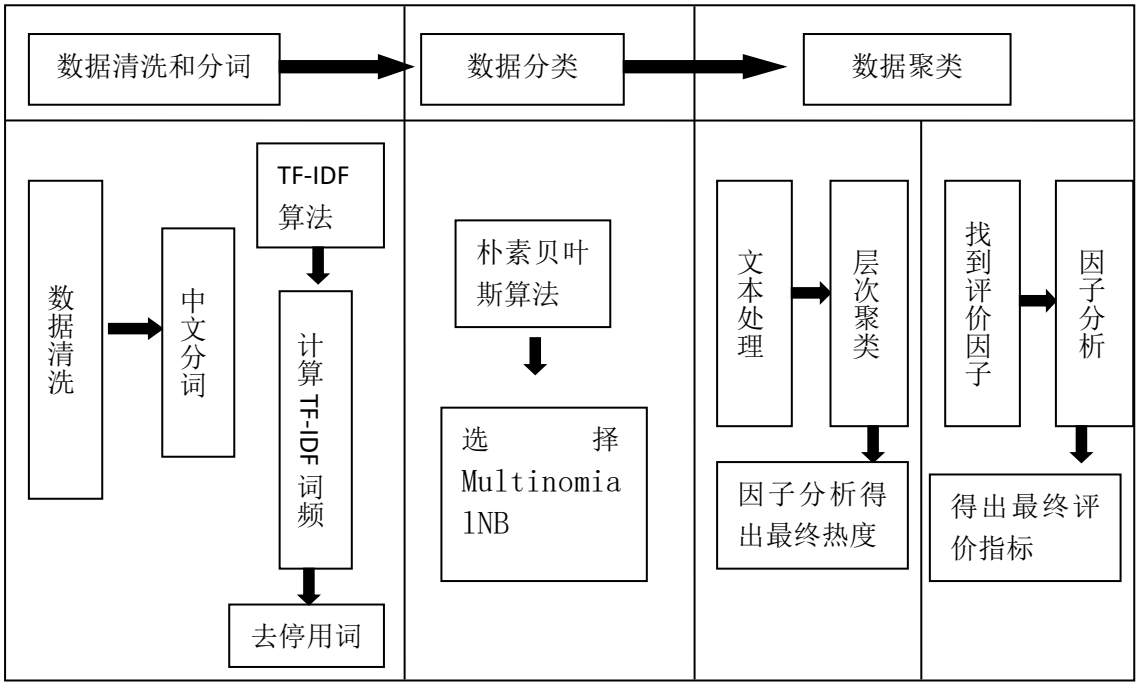


图 2-1 总体流程图

本题应用主要包括如下步骤：

步骤一：数据清洗和分词，首先使用 pandas 清除脏数据，之后定义一个分词预处理函数，采用 jieba 分词工具，将处理好的文本进行分词，这里先不对停用词进行处理，将文档信息转化为基于 TF-IDF 的向量，使用 TfidfVectorizer 将文档集合转为 TF-IDF 矩阵，并传入 stop_words 参数去除停用词；

步骤二：数据分析，利用朴素贝叶斯算法进行分类训练，通过尝试选择 MultinomialNB；

步骤三：数据聚类，根据文本向量，计算文档间的余弦相似度，再基于层次聚类算法对各个留言问题描述进行聚类。提取问题热度评价指标，构建热度评价体系，使用因子分析模型计算出问题的最终热度。

步骤四：提取答复评价指标，构建答复评价体系，使用因子分析模型计算出答复

文本的最终评价指标。

2.1 问题 1 分析方法与过程

2.1.1 流程图

问题 1 的流程图如图 2-2 所示

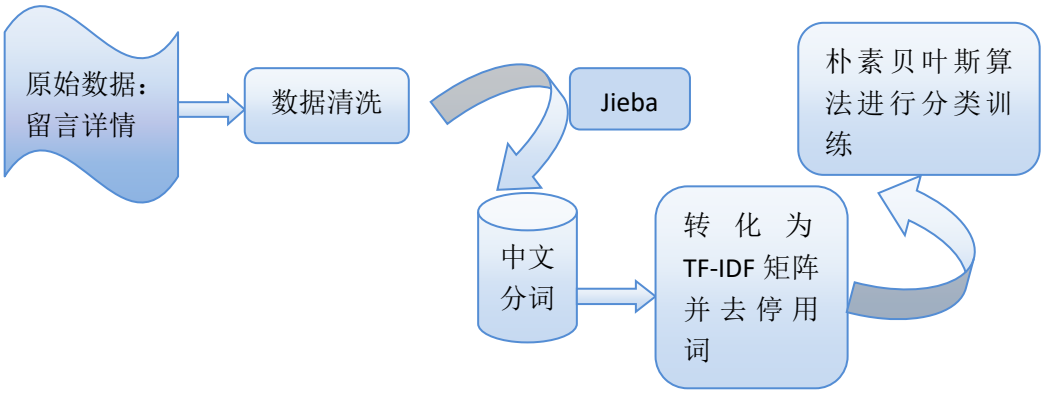


图 2-2 问题 1 流程图

2.1.2 数据清洗和分词

2.1.2.1 对留言详情进行中文分词

在对留言详情进行挖掘分析之前，先要把非结构化的文本信息转化为计算机能够识别的结构化信息。在附件 2 中，以中文文本的方式给了数据，为了便于转换，先要对这些留言详情描述信息进行中文分词。首先，，定义一个分词预处理函数，采用 python 的中文分词包 jieba 进行分词，主要去掉对文本分类无用的标点符号和数字，以及换行符等等，这里先不对停用词进行处理，因为 tf_idf 中有相关的参数。jieba 采用了基于前缀词典实现的高效词图扫描，生成句子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，对于未登录词，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效果。

通过对数据的观察不难发现：

1）留言主题和留言内容里都在不同程度上反映出了留言的类别，所以我们不做舍弃，用字符串拼接的方法，把两个以逗号拼接在一起

2）所给数据标签存在着一定的不平衡，这对后续的训练会造成影响，这个我们不选择数据增强或是欠采样的方法，具体操作在模型训练部分进行说明

2.1.2.2 TF-IDF 算法

在对留言详情描述信息进行分词后，需要把这些词语转换为向量，以供挖掘分析使用。这里采用 TF-IDF 算法，把留言详情描述信息转换为权重向量。TF-IDF 算法的具体原理如下：

第一步，计算词频，即 TF 权重（Term Frequency）。

$$\text{词频}(TF)=\text{某个词在文本中出现的次数} \quad (1)$$

考虑到文章有长短之分，为了便于不同文章的比较，进行"词频"标准化，除以文本的总词数或者除以该文本中出现次数最多的词的出现次数即：

$$词频（TF）=\frac{某个词在文本中出现的次数}{文本总词数} \quad （2）$$

或

$$词频（TF）=\frac{某个词在文本中出现的次数}{该文本出现次数最多的词的出现次数} \quad （3）$$

第二步，计算 IDF 权重，即逆文档频率（Inverse Document Frequency），需要建立一个语料库（corpus），用来模拟语言的使用环境。IDF 越大，此特征性在文本中的分布越集中，说明该分词在区分该文本内容属性能力越强。

$$逆文档频率（IDF）=\log（\frac{语料库的文本总数}{包含该词的文本数+1}）\quad （4）$$

第三步，计算 TF-IDF 的值（（Term Frequency Document Frequency）。

$$TF-IDF=词频（TF）\times 逆文档频率（IDF）\quad （5）$$

实际分析得出 TF-IDF 值与一个词在附件 2 中文本出现的次数成正比，某个词文本的重要性越高，TF-IDF 值越大。计算文本中每个词的 TF-IDF 值，进行排序，次数最多的即为要提取的职位描述表中文本的关键词。

2.1.2.3 生成 TF-IDF 向量

生成 TF-IDF 向量的具体步骤如下：

1）CountVectorizer

CountVectorizer 类会将文本中的词语转换为词频矩阵。

例如矩阵中包含一个元素 a[i][j]，它表示 j 词在 i 类文本下的词频。它通过 fit_transform 函数计算各个词语出现的次数，通过 get_feature_names()可获取词袋中所有文本的关键字，通过 toarray()可看到词频矩阵的结果

2）TfidfTransformer

TfidfTransformer 用于统计 vectorizer 中每个词语的 TF-IDF 值。具体用法如下：

将文档信息，即每条评论被分好词之后的词集合，转为为基于词频-文档词频（TF-IDF）的向量，向量的每个元素都是对应于某个词在这个文档中的 TF-IDF 值，在不同文档中，同一词的 TF-IDF 是不一样的。所有文档的 TF-IDF 向量堆放在一起就组成了一个 TF-IDF 矩阵。这里包含了除停用词之外的所有词的 TF-IDF 值，词的个数构成了向量的维度。我们使用 TfidfVectorizer 将文档集合转为 TF-IDF 矩阵，并传入 stop_words 参数去除停用词。本文使用的停用词集是中文停用词表, 哈工大停用词表, 百度停用词表, 四川大学机器智能实验室停用词库四份停用词表进行了合并去重得到的词集。

2.1.3 选择分类器进行训练

通过查阅资料可知，在文本分类方面朴素贝叶斯的效果显著，本文选择朴素贝叶斯模型对文本数据进行分类训练，朴素贝叶斯有三种形式：先验为多项式分布（MultinomialNB）、先验为高斯分布型（GaussianNB）以及先验为伯努利分布

型（BernoulliNB），通过尝试发现 MultinomialNB 的效果是最好的。MultinomialNB 假设特征的先验概率为多项式分布，即如下式：

$$P(X_j=x_{jl}\mid Y=C_k)=\frac{x_{jl}+\lambda}{m_k+n\lambda}$$

其中，$P(X_j=x_{jl}\mid Y=C_k)$是第 k 个类别的第 j 维特征的第 l 个取值条件概率。

m_k 是训练集中输出为第 k 类的样本个数。λ 为一个大于 0 的常数，常常取值为 1，即拉普拉斯平滑，也可以取其他值。

在分词的时候我们就已经观察到数据的标签存在不平衡的情况，这里我们采用交叉验证的方式削弱标签不平衡对模型的影响，将数据集分成 10 折，做一次交叉验证实际上是计算了十次，将每一折都当作一次测试集，其余九折当作训练集，这样循环十次，这样不仅从有限的数据中获取了尽可能多的有效信息，而且在一定程度上减小了数据过拟合。为了使结果更有说服力，我们一共进行 10 次交叉验证，score 参数选择 f1_score，并取平均值，最后我们进行调参，在朴素贝叶斯模型中可供我们调整的参数并不多，主要是 alpha，是添加拉普拉斯平滑，即为上述公式中的 λ，通过不断进行循环缩小范围发现，当 alpha 在 0.02 到 0.05 之间时 f1_score 是最大的，最终通过观察，令 alpha=0.027。f1_score 的值是 0.8953706077083338。

f1_score 的值如图 2-3 所示

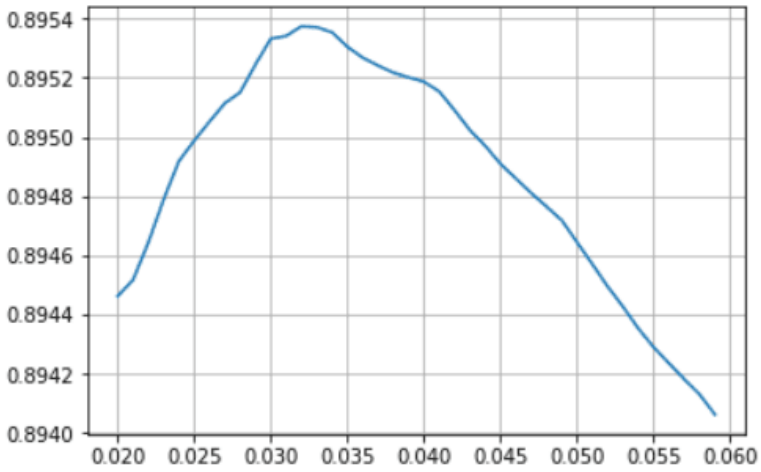


图 2-3 f1_score 值

2.2 问题 2 分析方法与过程

2.2.1 聚类模型

由于本题是一个无监督学习数据，没有给我们相应的标签，所以选择聚类模型对文本进行聚类操作。

常用的聚类模型有 KMeans、DBSCAN 以及层次聚类等等，其中 KMeans 算法的复杂度比较低，具有出色的速度，但是在实际操作中 KMeans 往往效果不是特别好，而且 KMeans 需要自己确定 K 值,K 值的选定是比较难确定，因为我们一开始并不知道要聚成多少类。

因此本文选择层次聚类，层次聚类可以根据距离来进行对类的划分。层次聚类有两大类，一种是从下而上地把小的 cluster 合并聚集，另一种是从上而下地将大的 cluster 进行分割。用的比较多的是第一种，层次聚类较大的优点，就是它一次性地得到了整个聚类的过程，只要得到了聚类树或者是样本之间的距离，想要分多少个 cluster 都可以直接得到结果，改变 cluster 数目不需要再次计算数据点的归属。层次聚类的缺点是计算量比较大，因为要每次都要计算多个 cluster 内所有数据点的两两距离。另外，由于层次聚类使用的是贪心算法，得到的显然只是局域最优，不一定就是全局最优，这可以通过加入随机效应解决。

2.2.1.1 尝试 DBI 作为聚类的评价标准

对于聚类好坏的评价有很多的标准，在大方向上被分成两类：一种是分析外部信息，另一种是分析内部信息。外部信息就是能看得见的直观信息，这里指的是聚类结束后的类别号。但是这种办法没法应用到实际生活（举个例子，如果要进行文本聚类，最后聚出了几个类，聚类是否正确可以根据分析文章内容来判断这几个样本是不是一个类，要是 1w 篇文章还这么做的话就很麻烦）。还有一种分析内部信息的办法，这种方法是聚类完后通过一些模型生成参数，这个参数表示聚类结果的优劣，诸如熵和纯度这种数学评价指标，本文我们首先选择 DBI 作为聚类的评价标准进行尝试。

2.2.1.2 尝试 single 作为层次聚类的 linkage 策略

层次聚类中比较重要的是对计算距离策略的选取，一共有四种：

1）Ward 策略：让所有类簇中的方差最小化。

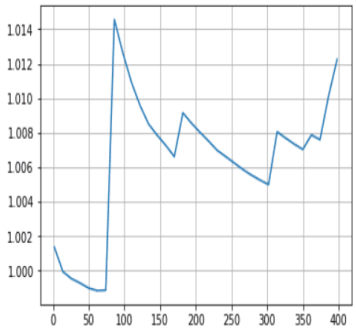
2）Maximum 策略：也叫 completed linkage（全连接策略），力求将类簇之间的距离最大值最小化。

3）Average linkage 策略：力求将簇之间的距离的平均值最小化

4）single 策略：单次使用所有观测之间的最小距离两组中的一组。

用的比较多的是 Average linkage 和 Ward，但是这里要提到的是根据文献显示，对于文本数据的距离计算来说，使用余弦相似度比使用欧氏距离的效果要好很多，因为余弦相似度更加注重方向了，然而如果使用 Ward 策略的话就必须要使用欧氏距离了，所以这里我们尝试 single 和 Average linkage 两种策略。

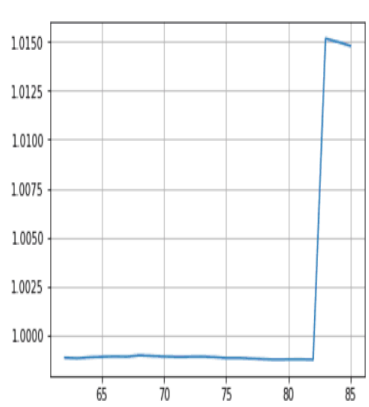
首先是 single，我们在 0 到 600 类之间进行测试,图 2-4 给出了测试的 DBI 指数



[1.0013515605823686, 0.999931327900426, 0.9995279256852597, 0.9992741495729307, 0.9989826620943149, 0.998834561942 69, 0.9988632805682446, 1.0145762615135387, 1.0126142751153242, 1.0109181036560149, 1.0095795931228382, 1.0085000526393153, 1.0078534025603896, 1.0072664963223021, 1.0066127101507731, 1.00915365928253 03, 1.0085540552297 71, 1.0080211277348 11, 1.0075132558801905, 1.0069759007666033, 1.0066156656815801, 1.0062409069463 97, 1.0058676110314853, 1.0055383053533 94, 1.0052514869734 08, 1.0049901606688147, 1.0080533909489389, 1.0076840988283442, 1.0073391711024176, 1.0070311118970108, 1.0078705543336 11, 1.0075928434370418, 1.0101178189243012, 1.0122592120942713]

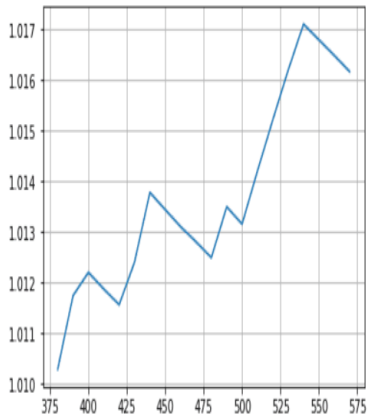
[0 0 0 ... 0 0 0]

图 2-4 DBI 指数（a）



[0.99883456194269, 0.9988054479829753, 0.9988515980921998, 0.9988761937074665, 0.9988984540014526, 0.998882593622479, 0.9989539566187186, 0.998923191369425, 0.9988936384582238, 0.9988709954695822, 0.9988760874770305, 0.9988847437946744, 0.9988632805682446, 0.9988188219203852, 0.9988223751579035, 0.9987846247819744, 0.9987621781284967, 0.9987321992081055, 0.9987467054307537, 0.9987508816938987, 0.9987295478762528, 1.0151801194061443, 1.0150013995957172, 1.0147994005520966]

图 2-4 DBI 指数（b）

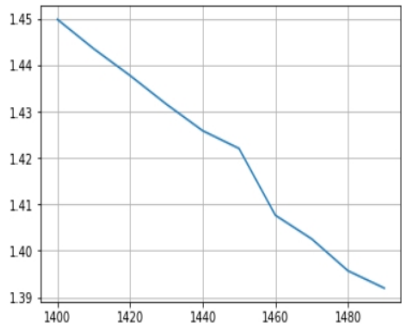


[1.0102900567867703, 1.011745558141875, 1.0122057221975462, 1.011875417088191, 1.0115676850315565, 1.0124065503912156, 1.0137842418027363, 1.013444599246568, 1.0131083797508853, 1.0128103403386932, 1.0124977586210016, 1.0135013319638415, 1.013161184815911, 1.014208015660095, 1.015221832741505, 1.0162016814309105, 1.0171119375951607, 1.0168025689926128, 1.0164932810555443, 1.0161752603858625]

图 2-4 DBI 指数（c）

2.2.1.3 尝试 Average 作为层次聚类的 linkage 策略

使用 Average linkage 策略，并把距离计算方式改为余弦相似度（cosine）DBI 指数与类的关系如图 2-5 所示



[1.4499307026584005, 1.443603750475549, 1.4378220878033974, 1.4316380891910845, 1.4259075632867768, 1.4220715311057763, 1.4076613084488063,]
[859 891 122 ... 160 160 160]

图 2-5 DBI 指数与类数关系

这里我们可以明显发现的是，DBI 指数出现随着类数的增加持续下降的现象，

查阅相关文献之后发现，DBI 的计算方式是欧式距离，已经不适合作为本题的评判。通过对多种评价指标的对比，不难看出 S_Dbw 这种评价指标是最好的，我们对这种算法进行实现，但在运用到实际的过程中发现复杂度比较高，计算一轮需要耗费大量的时间，导致我们主要还是通过人工的方式去检查聚类的准确性。

那么这里就需要使用距离作为一个分类的标准了，首先使用 linkage 函数去实现一遍，可以得到相应的类和距离之间的关系，通过观察我们把距离设置在 0.3 到 0.9 之间，进行画图,图 2-6 给出了类与距离的关系

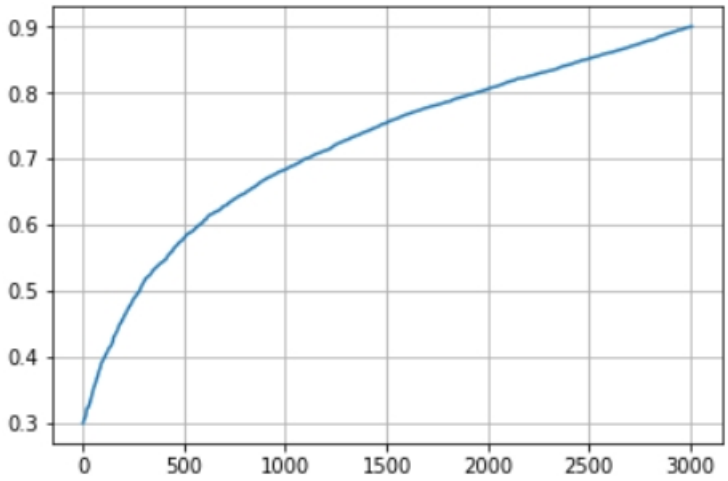


图 2-6 类和距离的关系

使用 Agglomerative Clustering 凝聚聚类，循环尝试不同的距离，发现只有在将近 1 的时候，才会有比较好一点的效果，但是结果还不是完全准确。

这里我们做一个调整，我们可以看出，留言主题很多时候比留言内容更能表达留言的意思，所以我们选择调整留言主题所占的权重，具体操作是在做数据清洗的时候将留言主题尝试重复两遍和三遍之后再加上留言内容，以加大留言主题所占的比重，同时开始逐渐将距离放小，进行测试，可以看出，在将留言主题重复加上两遍时数据聚类结果有了很大的改善，但是三遍的时候又会出现会很多因一个相似词就组在一起的例子，所以选择重复两遍的形式，最终确定距离为 0.82 的时候呈现的效果比较好，并将计算出的标签保存。

2.2.2 生成热点问题留言明细表

保存的标签读成 DataFrame 的格式命名为问题 ID，用 pandas 中的 concat 函数把标签合并到附件 3 中的第一列，并使用 sort_value 函数以标签作为依据对文章进行排序，最后保存为 excel 格式的热点问题明细表

2.2.3 生成热点问题表

2.2.2.1 热度指标体系构建

指标体系的构建原则通常根据要求和对象的不同分为三个层面：指标选取成眠一般采取客观性原则、系统性原则和敏感性原则，客观性原则是指指标体系的选择必须从客观实际出发，全面准确地反映投诉的热度情况,克服因人而异的主观因素的影响。系统性原则是指指标体系的设计应从系统整体出发,能够包络形成热门问题的各个因子,各指标间既相互独立又相互联系,共同构成一个有机整体。计算与操作层面一般采用数据的可得性和可操作原则,是指在设计指标体系时用较少指标反映较多的实质性内容，而且指标便于收集和量化。

问题热度是指某位市民利用网络问政平台发布信息,并引起政府和其他民众对该信息的广泛关注，共鸣和讨论的热烈程度,其实质上是--种信息传播活动。根据新闻传播学理论,信息传播活动涉及 4 个要素,即信源、信宿、信道、信息。由此对信息传播效果产生影响的因素主要有 4 个方面，即传播者、受传者、传播渠道和传播内容 4。这与马尔科姆·格拉德威尔提出的流行三要素理论具有相似之处,他认为物体想要流行必须具备流行的基本要素，即关键人物法则、环境威力法则和内容附着力法则。投诉问题热度与事物流行异曲同工,关键人物法则具体到问题热度中是指发布这个问题的人。环境威力法则是指在不同的环境、不同的时间内发布不同的内容所引起的热度效果决然不同,但是这里体现并不多，所以我们不做考虑内容。附着力法则是指问题热度还取决于问题内容是否具有简洁性、不可预期性、具体性、可信性、情绪性、叙事性等特征,在表现形式上是否带有鲜明主题,长度是否适宜等。

综上我们构造三个特征影响力

(1) 发布人特征热度影响力。问题发布人的身份、信息和特征对问题的热度有相当重要的影响。但是在我们的数据中看不出问题发布人的信息，所以我们选取每一类问题的发布人数和发布条数做两个二级指标

(2)内容特征热度影响力。根据流行三要素理论可知留言自身内容特征对其传播热度具有很强的影响力。问题字数越多其表达出的信息越充实,也越容易引发讨论。所以这里我们选择留言主题和留言内容的长度作为二级指标

(3)传播特征热度影响力。问题的传播特性可以最直观的反映出-条问题引起的关注度。而这里的传播方式即为点赞数和反对数,问题被点赞的次数和反对的次数都会反映问题的热度。又因为每一类基本都有不止一个的问题，所以把这里提到的点赞数和反对数都改写成平均点赞数和平均反对数,作为两个第二级指标

表 1：问题热度综合评价指标体系

	一级指标	二级指标	指标内涵
问题热度综合评价指标体系	发布人特征热度影响力	发布人数	发布问题的总人数
		发布条数	发布问题的数量
	内容特征热度影响力	留言主题的长度	留言主题的字数
		留言内容的长度	留言内容的字数
	传播特征热度影响力	平均点赞数	每一类问题被点赞的平均次数
		平均反对数	每一类问题被反对的平均次数

选取完指标后就可以进行模型的构建了

2.2.2.2 因子分析模型构建

因子分析是指通过少数不相关的因子反映多个具有相关性的原始信息,起到降维和剔除相关性的作用。因子分析的前提是具有一定的相关性，因此必须通过了 kmo 和 bartlett 球形度检验的数据才能进行因子分析。kmo 值要大于 0.7，bartlett 球形度检验 p 值要小于 0.05，则认为通过了适用性检验后进行因子分析。在对数据进行标准化处理之后进行，kmo 和 bartlett 球形度检验，其中 kmo 指数为 0.6485389091511172，bartlett 指数为 0.1377385977161934，都比要求的差一

点，但是我们主要还是以实际效果为准。最终我们根据权重确定三个因子，其中有三个参数需要确定：

（1）rotation:旋转的方式，包括None:不旋转,'varimax':最大方差法,'promax'：最优斜交旋转；

（2）n_factors:公因子的数量；

（3）method:因子分析的方法，包括'minres':最小残差因子法,'principal':主成分分析法；

经过多次试验确定把 fa = FactorAnalyzer(n_factors=3, method='principal', rotation='varimax')中的 n_factors 设为 3，methor 设为 principal 即主成分分析的方法 rotation 设为 varimax 即最大方差法旋转

相关系数：

	问题ID	留言编号	反对数	点赞数
问题ID	1.000000	-0.056658	-0.004475	-0.016385
留言编号	-0.056658	1.000000	0.016655	-0.024378
反对数	-0.004475	0.016655	1.000000	0.047778
点赞数	-0.016385	-0.024378	0.047778	1.000000

协方差矩阵：

```
 [[1.00062112 0.03001679 0.01280856]
 [0.03001679 1.00062112 0.03210319]
 [0.01280856 0.03210319 1.00062112]]
[1.00004379 0.99997895 0.99997726]
[0.33334793 0.33332632 0.33332575]
[0.33334793 0.66667425 1.        ]
```

每个的因子得分：

	0	1	2
0	0.145104	3.828821	0.043728
1	4.776549	-0.045458	-0.003264
2	-0.177293	0.107074	-0.018521
3	-0.187900	-0.501027	-0.064909
4	0.596751	-0.394628	-0.040086
...	...	...	...
1606	-0.438896	0.123728	-0.086174
1607	-0.349347	5.591538	-0.052499
1608	-0.445367	-0.587627	0.250589
1609	-0.408859	1.957783	-0.074879

最后得出因子的最终打分作为热度指标，并保存。

以 DataFrame 的格式读入热度指标并命名为热度指数，使用 pandas 中的 concat 的方法插入一列问题 ID 并使用 sort_value 的方法进行排序，最后用 head 的方法取热度前五的类并保存为热度问题表。

2.3 问题 3 分析方法与过程

2.3.1 答复评价指标体系构建

本题是开放性问题，采用和第二问相同的方式先找到合适的评价因子，问题答复的好坏主要有这个几个因素做为评判

（1）答复文本的相似度，首先评价这个答复文本好坏与否最重要的一点就是答复文本的内容是否和评论文本内容是是在描述同一件事，所以文本相似度作为一级标签

（2）答复文本的长度，这个是判断答复是否是仅仅一两句话带过，并未对解决问题起到一定作用，所以答复文本长度作为一级标签

（3）答复文本时效性，判断答复的时间和反映问题的时间之间的差值，如果是隔了很长时间才答复，很明显这样的答复失去了时效性，也是质量不高的，所以答复文本的时效性作为一级标签

表 2：答复评价指标体系

	一级标签	指标内涵
答复评价指标体系	文本相似度	答复文本与评论文本内容是否描述同一件事
	答复文本长度	答复文本字数
	答复文本时效性	答复的时间

2.3.2 因子分析模型构建

具体的模型解释第二问已给出，这里我们不再赘述，仅仅描述我们的模型，在对数据进行标准化处理之后进行，kmo 和 bartlett 球形度检验，其中 bartlett 的值变成（inf，NAN），我们对因子进行重新处理，经过尝试我们去掉时效性这个因子，再次进行 kmo 和 bartlett 球形度检验，其中 kmo 指数为 0.5000000000000003，bartlett 指数为 1.9112588081893207e-60，同样的分数显示不是很好，我们以实际效果为主，最终确定把 fa = FactorAnalyzer(n_factors=3, method='principal', rotation='varimax')中的 n_factors 设为 3，methor 设为 principal 即主成分分析的方法 rotation 设为 varimax 即最大方差法旋转

```
协方差矩阵：
 [[1.00035524 0.3013139 ]
 [0.3013139  1.00035524]]
[1.30119413 0.69880587]
[0.65059706 0.34940294]
[0.65059706 1.        ]
```

```
每个的因子得分：
                0         1
0       1.171118  0.591540
1      -0.811239 -0.462303
2       1.084875  0.796566
3       0.558702  0.537498
4       1.491674  1.640789
...          ...       ...
2811   -1.858158 -0.485861
2812   -1.798224 -0.458143
2813    0.593970 -0.342699
2814    1.044836  0.398325
2815   -0.640726 -0.158757
```

最后得出因子的最后得分作为答复文本的评价指标，并保存。

以 DataFrame 的格式读入答复文本指标并命名为答复文本指标，使用 pandas 中的 concat 的方法插入一列答复文本指标并使用 sort_value 的方法进行排序并保存为答复文本质量表。

3、结论

总结此次比赛，我们首先使用 pandas 清洗数据，之后采用 jieba 进行文本分词处理，将文档信息转换为基于 TF-IDF 的向量，使用使用 TfidfVectorizer 将文档集合转为 TF-IDF 矩阵，并传入 stop_words 参数去除停用词。利用朴素贝叶斯算法对文本进行分类训练，最终选择 MultinomialNB 求得正确率 f1_score 的值是 0.8953706077083338。。层次聚类对留言主题和内容进行聚类；根据问题热度评价指标构建热度评价体系，采用因子分析模型得出最终热度。提取答复评价指标，构建答复评价体系，使用因子分析模型计算出答复文本的最终评价指标。但是我们这个分数还有提高的空间，通过对比训练集的准确率和测试集的准确率可以很明显的看到，模型还是存在一定程度上的过拟合，这个后续可以通过增加噪声和正则项来进行进一步的完善，同时数据的标签不平衡问题可以通过数据增强或是欠采样来进一步的完善。

4、参考文献

[1] 金燕.国内外 UGG 质量研究现状与展望.郑州大学.2016

[2] 杨洋洋,谢雪梅.基于 QCA 的网络舆情热度影响因素构型分析.北京邮电大学.2019

[3] 聂卉.基于内容分析的用户评论质量的评价与预测.中山大学.广东省哲学社会科学 2013 年度项目.2014

[4] 梁昌明,李冬强.基于新浪热门平台的微博热度评价指标体系实证研究.山东师范大学.北京科技大学.2015

[5] M. Halkidi, M. Vazirgiannis. Clustering validity assessment: finding the optimal

partitioning of a data set. Proceedings IEEE International Conference on. Data Mining, 2001. ICDM 2001
[6] 丁晟春,王小英,刘梦露.基于本体和加权朴素贝叶斯的网络舆情主题分类.南京理工大学.2018
[7] 胡青云.中文信息处理发展报告.中文信息学会（Chinese Information Processing Society of China）.2016