“智慧政务”的分析与挖掘

摘要

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。因此，运用自然语言处理技术分析与文本挖掘技术对“智慧政务”的研究具有重大意义。

对于问题 1，通过对附件 2 进行留言详情去重处理并保存第一条记录，得到不重复的留言详情信息；用 0—5 去替换一级分类中的 7 个标签。利用 jieba 中文分词工具对留言详情进行分词，并通过停用词表和自定义停用词去除停用词，并通过 TF-IDF 算法提取每条留言详情描述的前 22 个关键词，再利用 TF-IDF 算法得到每条留言详情的 TF-IDF 权重向量。将数据分为测试集、训练集，比例为 2：8，通过多项式朴素贝叶斯进行分本分类，并通过不断调整拉普拉斯平滑系数优化分类模型，提高模型精度。

对于问题 2，通过对附件 3 留言主题空缺、不合理的地方进行修改和填充，对留言主题进行分词并去停用词，对每条留言详情构建一个词袋，计算出每条留言详情的 TF-IDF 权值向量，根据权值向量对留言详情权值向量两两间用余弦相似度计算文本距离，根据距离对问题进行层次聚类，把同一问题进行归类。对聚类结果进行计数，通过同一问题的初始质量数值表达和内容质量的数值表达，对问题进行热度评价，通过热度得分选出排名前五的问题。

对于问题 3，通过对附件 4 中的答复意见、留言详情进行分词，每个样本构建词袋并算出 TF-IDF 权值向量，对答复意见、留言详情向量计算余弦相似度对答复意见的相关性进行评价；对答复意见提取关键词对留言详情的对比分析答复意见的可解释性；衡量答复意见是否满足打招呼、收到留言、经调查、回复、感谢、落款年月日这些流程去衡量完整性；并通过计算答复意见的时效性和情感分析评分对答复意见进行评价。最后对相似性、完整性、可解释性、时效性、情感态度评分进行量化构建答复意见评分模型

关键词：TF-IDF 算法 多项式朴素贝叶斯 拉普拉斯平滑系数 余弦相似度 层次聚类

目录

1．挖掘目标

本次建模的目的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 答复意见的相关数据，利用 jieba 中文分词工具对留言详情进行分词、多项式朴素贝叶斯的方法对留言详情进行一级标签分类；计算文本相似度并进行层次聚类对热点问题进行挖掘；通过计算答复意见和留言详情相似度、完整性、可解释性等对答复意见进行评价。本文主要达到以下三个目标：

（1） 利用文本分词和文本分类的方法对非结构化的数据进行文本挖掘，根据附件 2 的留言特征进行训练，得出一定的分类规则，对未知留言进行一级标签分类，并用 F-score 对分类方法进行评价，并通过 F-score 来不断调整模型。

（2） 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。统计出排名前 5 的热点问题。

（3） 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、、时效性四个角度角度对答复意见的质量给出一套评价方案。

2. 分析方法与过程

本文主要包括以下步骤：

步骤一：数据预处理，在题目中给出的数据中，有部分数据会出现较多重复的留言详情信息，在原始数据上进行去重处理，并在此基础上进行中文分词。

步骤二：数据分析，在对留言详情信息分词后，通过 TF-IDF 算法提取每条留言详情描述的前 22 个关键词并把这些词语转换为权重向量。采用多项式朴素贝叶斯算法对留言详情进行一级标签分类，根据类别在整体上占多大比例去判断所属类别。

步骤三：对附件 3 留言主题空缺或概括不当的进行修补和填充，对留言主题进行分词并去停用词。对每个文本构建一个词袋模型，计算出每个文本的 TF-IDF 权值向量。计算每个留言主题间的余弦相似度，通过层次聚类将问题进行聚类，并对问题进行热度评价，得到排名前 5 的热点问题。

步骤四：先对留言详情和答复意见进行分词处理，每个样本构建一个词袋，求出每个样本的 TF-IDF 权值向量，对留言详情和答复意见向量进行余弦相似度计算。通过答复意见文本是否满足某一规范性进行完整性分析。对答复意见的回复进行关键词提取查看文本是否对客户留言起到价值。并通过时效性和情感评分对答复意见进行评价。

2.1 问题 1 分析方法与过程

2.1.1 流程图

2.1.2 数据预处理

2.1.2.1 留言详情的去重

在题目附件 1 给出的数据中，出现较多重复的留言详情信息。考虑到同一用户由于反映问题还没解决，会在不同时间段可能会反映同一问题。因此需要把重复的留言信息去掉保存一条记录即可。

2.1.2.2 对留言详情信息进行分词并去除停用词

在对留言详情信息进行挖掘分析之前，先把非结构化的文本信息转换为计算机能够识别的结构化信息。在附件 1 数据中，以中文文本形式给出，为了便于转换，先将这些留言详情进行中文分词。这里采用 python 中的 jieba 库进行分词，采用精确模式进行分词，对于未登录词，则采用基于文字成词能力的 HMM 模型，使得能更好的实现中文分词效果。

在分词的同时，去除对留言详情进行一级标签分类无帮助或无意义的词。

根据分词结果绘制词云图，通过词云图的展示可以对留言详情各分类数据分词后的高频词予以视觉上的强调突出效果，使得阅读者一眼就可获取到主旨信息。

然后采用 TF-IDF 提取关键词算法，抽取每条留言详情中的前 22 个关键词，节省文本分类的空间存储。

2.1.2.3 TF-IDF 算法[1]

在对留言详情数据分词并提取关键词后，需要把这些词转换为向量，以供挖掘分析使用。这里采用了 TF-IDF 算法，把留言详情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。

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

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：

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

或

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

第二步，计算 IDF 权重，即逆文档频率，需要建立一个语料库，用来模拟语言的

使用环境。IDF 越大，此特征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。

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

第三步，计算 TF-IDF 值。

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

实际分析得出 TF-IDF 值与一个词在留言详情表中文本出现次数成正比，某个词文本重要性越高，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留言详情数据中的关键词。

2.1.2.3 生成 TF-IDF 向量法[1]

生成 TF-IDF 向量的具体步骤如下：

（1） 使用 TF-IDF 算法，找出每条留言详情的前 22 个关键词；

（2） 对每条留言数据的前 22 个关键词合并成一个集合，计算每条留言详情数据对于这个集合中词的词频，如果没有则记为 0；

（3） 生成每条留言详情数据的 TF-IDF 权重向量，计算公式如下：

（4） $\text{TF-IDF}=\text{词频(TF)}\times\text{逆文档频率（IDF）}$ （6）

2.1.3 留言详情的一级分类

生成留言详情 TF-IDF 权重向量后，根据每条留言详情的 TF-IDF 权重向量，对留言详情进行一级分类，这里采用多项式朴素贝叶斯算法进行分类。

多项式朴素贝叶斯算法原理如下：

朴素贝叶斯法利用贝叶斯定理首先求出联合概率分布，再求出条件概率分布。这里的朴素是指在计算似然估计时假定了条件独立。基本原理可以用下面的公式给出[2]：

$$P(Y|X)=\frac{P(Y)P(X|Y)}{P(X)} \quad (7)$$

其 中 ， $P(X|Y)=P(X1,X2,\dots Xn|Y)=P(X1|Y)P(X2|Y)\dots P(Xn|Y)$ （8）

$P(Y|X)$叫做后验概率，$P(Y)$叫做先验概率，$P(X|Y)$叫做似然概率，$P(X)$叫做证据。

多项式朴素贝叶斯算法步骤如下：

（1）训练阶段

先验概率

$$P(C|c)=\frac{\text{属于类 c 的文档数}}{\text{训练集文档总数}} \quad (9)$$

条件概率

$$P(\omega_i|c)=\frac{\text{词}\omega_i\text{在属于类 c 的所有文档中出现次数}}{\text{属于类 c 的所有文档中的词语总数}} \quad (10)$$

注：
（1）条件概率$P(\omega_i|c)$表示的是词ω_i在类别c中的权重
（2）条件概率独立性假设，丢失了词语的位置信息，在文本表示上来说，就是它失去了语义信息。当然可以通过 ngram 的特征来减少损失，但是也不能有效解决语义上的损失。
（3）先验概率和条件概率的计算都利用了最大似然估计。它们实际算出的是相对频率值，这些值能使训练数据的出现概率最大。
拉普拉斯平滑（加 1 平滑）

$$P(\omega_i|c)=\frac{\text{词}\omega_i\text{在属于类 c 的所有文档中出现次数+1}}{\text{属于类 c 的所有文档中的词语总数}} \quad (11)$$

加 1 平滑可以认为是采用均匀分布作为先验分布，即每个词项在每个类别中出现一次，然后根据训练数据对得到的结果进行更新。也就是说未登录词的估计值为 1/词汇表长度。
（2）预测阶段

$$\begin{aligned}&arg\max_{c\in C}P\left(c\middle|\omega_{1,}\omega_2,\dots,\omega_n\right)\\&=arg\max_{c\in C}\left[P(c)P(\omega_{1,}\omega_2,\dots,\omega_n|c)\right]\\&=arg\max_{c\in C}\left[P(c)P(\omega_{1,}|c)P(\omega_2|c)\dots P(\omega_n|c)\right]\\&=arg\max_{c\in C}\left[logP(c)+logP(\omega_{1,}|c)+logP(\omega_2|c)+\cdots\right.\\&\left.+logP(\omega_n|c)\right]\quad(12)\end{aligned}$$

其中，$P\left(c\middle|\omega_{1,}\omega_2,\dots,\omega_n\right)$哪个值最大，则把样本判为哪一类。

由于附件 2 留言详情数据表给出了 9210 条记录，去重后还有 9052 条记录，把分词后的记录进行随机抽取分为训练集和测试集，其中训练集与测试集比例为 8：2，再求 TF-IDF 向量用多项式朴素贝叶斯算法进行分类。

2.1.4 留言详情的一级分类评价

本模型采用 F-Score 对分类结果进行评价[3]。
（1）精确度表示的是分类为负类的样本中实际为负类的样本所占的比例，精确度越高，模型某类的分类效果越好。

$$\text{精确度（Precision）}=\frac{TN}{TN+FN} \quad (13)$$

其中，TP（True Positive）、TN（True Negative）：预测答案正确， FN（False Negative）：本类标签预测为其他类标。
（2）召回率表示被正确分类的负类的比例，召回率越高，表示模型将负类误分为正类的模型概率越低，模型效果越好。

$$\text{召回率（recall）}=\frac{TN}{TN+FP} \quad (14)$$

其中，FP（False Positive）：错将其他类预测为本类。

（3）准确率(accuracy)：代表分类器对整个样本判断正确的比重。

$$准确率（accuracy）=\frac{TP+TN}{TP+TN+FP+FN} \quad （15）$$

（4）每个类别下的 f1-score

$$f1_k = \frac{2\times \text{Precision}_k \times \text{recall}_k}{\text{Precision}_k + \text{recall}_k} \quad （16）$$

（5）F1-Score：综合考虑精确度与召回率，其中 P 指精确率，R 指召回率。即各个类别下的 f1-score 求均值。

$$\text{F1} - \text{Score} = \frac{1}{n}\sum_{k=1}^{1} f1_k （k 为类别） \quad （17）$$

2.2 问题 2 分析方法与过程

2.2.1 文本预处理

观察附件三中的数据，留言内容包含大量的无关语句或是与问题相关性不大的句子，尝试利用提取关键词处理，但效果并不理想，并且留言主题包括了有效人群和问题地点，有利于热点问题的挖掘，多次尝试后决定利用留言主题进行聚类处理。

（1）检查留言主题内容是否完整、概括是否恰当，对空缺或概括不当的进行修改和补充；

（2）根据留言内容利用 python 提供专门的中文库 jieba 进行分词；

（3）利用下载好的停用词表 stopword.txt 对分词后的数据进行筛选去除无用字词及标点符号。

（4）数据中存在同一用户在短时间内反映相同内容的问题，属于恶意刷票行为，本文只选取其中一个进行排名。

2.2.2 构建词袋模型[6]

文本被切分成单词后，无论是中文、英文还是标点符号，都是机器无法直接做计算的，需要先将文本转化为可计算和比较的数学表示的形式。先将所有文本中的词汇构建成一个词条列表，其中不含重复的词条。然后对每个文本，构建一个向量，向量的维度与词条列表的维度相同，向量的值是词条列表中每个词条在该文本中出现的次数，这种模型叫做词袋模型。

2.2.3 计算 TF-IDF 权值[6]

TF-IDF 是一种统计方法，用来评估一个词条对于一个文件集中一份文件的重要程度。TF-IDF 的主要思想是：如果某个词在一篇文章中出现的频率 TF 高，并且在其他文件中很少出现，则认为此词条具有很好的类别区分能力，适合用来分类。将词袋向量转换为 TF-IDF 权值向量，更有利于判断两个文本的相似性。

具体原理如上文介绍。

2.2.4 计算相似度[6]

利用 TF-IDF 权值矩阵计算文本的距离矩阵，而对于文本相似度有以下几种计算方式：

1、欧氏距离

欧氏距离是一种常用的距离定义，指在 m 维空间中两个点之间的真实距离，对多维向量 A=(A1，A2，……,An)，B=(B1，B2，……,Bn)，欧氏距离的计算公式如下：

$$dist(A,B)=\sqrt{\sum_{i=1}^{n}(A_i-B_i)^2} \qquad (18)$$

2、余弦相似度

余弦相似度用向量空间中两个向量夹角的余弦值作为衡量两个个体差异的大小。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。余弦值的计算公式如下：

$$cos\,\theta=\frac{\sum_{i=1}^{n}(A_i\times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\times\sqrt{\sum_{i=1}^{n}(B_i)^2}} \qquad (19)$$

相对于欧氏距离，余弦相似度更适合计算文本的相似度。首先将文本转换为权值向量，通过计算两个向量的夹角余弦值，就可以评估他们的相似度。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，具体数据如 HSet.csv 所示。

2.2.5 层次聚类[7]

由于本文主题的分类总量未知，层次聚类为不给定聚类数目的情况下对数据对象进行聚类，符合题意需求，决定采用层次聚类对文本进行分类。

层次聚类的合并算法通过计算两类数据点间的相似性，对所有数据点中最为相似的两个数据点进行组合，并反复迭代这一过程。简单的说层次聚类的合并算法是通过计算每一个类别的数据点与所有数据点之间的距离来确定它们之间的相似性，距离越小，相似度越高。并将距离最近的两个数据点或类别进行组合，生成聚类树。合并过程如下：

我们可以获得一个的距离矩阵 X，其中表示和的距离，称为数据点与数据点之间的距离。记每一个数据点为将距离最小的数据点进行合并，得到一个组合数据点，记为 G 数据点与组合数据点之间的距离: 当计算 G 和的距离时，需要计算和 G 中每一个点的距离。组合数据点与组合数据点之间的距离：主要有 Single Linkage，Complete Linkage 和 Average Linkage 三种。这三种算法介绍如下：

（1）Single Linkage

Single Linkage 的计算方法是将两个组合数据点中距离最近的两个数据点间的距离作为这两个组合数据点的距离。这种方法容易受到极端值的影响。两个很相似的组合数据点可能由于其中的某个极端的数据点距离较近而组合在一起。

$$d_{min}(C_{i,}, C_{j,}) = min_{x \in C_{i,} x \in C_{j,}} dist(x, z) \quad (20)$$

（2）Complete Linkage

Complete Linkage 的计算方法与 Single Linkage 相反，将两个组合数据点中距离最远的两个数据点间的距离作为这两个组合数据点的距离。Complete Linkage 的问题也与 Single Linkage 相反，两个不相似的组合数据点可能由于其中的极端值距离较远而无法组合在一起。

$$d_{max}(C_{i,}, C_{j,}) = max_{x \in C_{i,} x \in C_{j,}} dist(x, z) \quad (21)$$

（3）Average Linkage

Average Linkage 的计算方法是计算两个组合数据点中的每个数据点与其他所有数据点的距离。将所有距离的均值作为两个组合数据点间的距离。这种方法计算量比较大，但结果比前两种方法更合理。

$$d_{avg}(C_{i,}, C_{j,}) = \frac{1}{|C_{i,}|\ |C_{j,}|} \sum_{x \in C_{i,}} \sum_{x \in C_{j,}} dist(x, z) \quad (22)$$

经过观察多次测试效果，Average Linkage 比较更符合需求。

2.2.6 热度模型建立[8]

对聚类结果进行计数，为了减小维度，本文选取问题个数大于 3 的聚类进行模型分析，具体数据如 data_mini.csv 所示，热度模型基于该数据进行分析。

参考 Reddit 的排名算法，对问题建立热点模型计算综合得分，Reddit 排名算法综合得分公式如下所示：

$$Score = log_{10} z + \frac{yt}{45000} \quad (23)$$

观察数据发现问题的点赞量并不多而且问题有效数最多是 13，数量也较小，经过量化决定将热点划分为两个因素：

W：初始质量的数值表达，本文采用有效问题的个数表达问题的初始质量，越多群众反映留言问题质量越高

I：内容质量的数值表达，问题内容引起的关注度，本文认为反对也是对问题的一种关注形式，所以采用反对数和点赞数之和来表示，这部分的权重因子设为 0.01。

由于问题的留言时间间隔并不长，并且有些问题是持续存在并非短时间可解决，所以本文不考虑将时间作为一个影响热点得分的负因子。

最终热点得分计算公式如下所示：

$$Score = W + 0.01I \quad (24)$$

利用上述提取的数据 data_mini.csv 计算综合得分并按综合得分排序得到排名前五的设点问题，数据如热点问题留言明细表.xlsx 所示，再整理出热点问题的相关内容整理到热点问题表.xlsx。

2.3 问题 3 分析方法与过程

通过查看附件 4 数据，该数据集不存在缺失值和重复值。本模型通过答复意

见的时效性、政府部门情感分析、留言详情与答复意见的相关性、完整性和可解释性展开研究。

2.3.1 政府部门答复意见完整性分析

本模型在研究答复意见完整性时，主要研究政府在答复意见上是否满足一定的规范性。此模型的规范性主要由以下方面的关键词进行评价，如：打招呼（您好）、经实际调查再进行回复（经调查核实）、收到客户留言（反映、已收悉）、进行回复（回复如下、答复如下）、表达感谢（感谢您对我们工作、支持）、答复时间（××××年××月××日）。如果答复意见满足以上打招呼、收到留言、经调查、回复、感谢、落款年月日这些则可认为这样的回复是比较规范的。

因此，本模型通过对答复意见进行分词并去停用词后对出现这些流程的关键词进行统计，次数越多则可认为该答复意见的文本的完整性是比较好的。

2.3.2 政府部门答复意见相关性分析[9]

相关性分析步骤如下：

（1）处理留言详情

第一步：把每个留言详情文本分词并去停用词，成为词包（bag of words）。

第三步：统计留言详情样本总数 M。

第三步：统计第一个留言详情词数 N，计算第一个留言详情第一个词在该留言详情中出现的次数 n，再找出该词在所有文档中出现的次数 m。则该词的 tf-idf 为：n/N * 1/(m/M)

第四步：重复第三步，计算出一个留言详情所有词的 tf-idf 值。

第五步：重复第四步，计算出所有留言详情每个词的 tf-idf 值。

（2）处理答复意见

第一步：对答复意见进行分词。

第二步：根据留言详情库（文档）的数据，计算答复意见中每个词的 tf-idf 值。

（3）相似度的计算

使用余弦相似度来计算用户查询和每个网页之间的夹角。夹角越小，越相似。

2.3.3 政府部门答复意见可解释性分析

通过对答复意见进行分词并去除停用词，根据 TF-IDF 提取前 10 关键词组成留言详情的短文本，查看文本与对应留言问题，看是否能对留言问题起到具体价值意义。

2.3.4 时效性

在用户反映信息给政府部门，然后政府部门根据用户提出问题进行答复，这期间留言时间和答复时间是具有时间间隔性的。如果用户在提出问题后政府部门

能够及时回复，即时间间隔越短，证明政府部门越高效的处理客户留言，做到更好地为人民服务。

在计算时效性时，本模型先算出答复时间和留言时间的时间间隔，为了衡量时效性，还要根据平均间隔时间进行衡量。其中在进行时效性计算时涉及以下公式。

时间间隔=答复时间—留言时间（以小时为单位）　　（25）

$$时效性=\frac{时间间隔}{平均时间间隔}　　（26）$$

如果该值越小，则证明政府部门在进行人民留言回复的工作效率越高。

2.3.5 政府部门答复意见情感分析

为了对政府部门答复意见的情感态度进行评分，先将答复意见这一列抽取出来，再用隐马模型进行分词并去掉停用词，停用词表中没有但分词结果出现的符号进行自定义去掉。情感分析主要是判别文本的情感倾向性，即属于正面、负面、中性。本模型采用 BosonNLP_sentiment_score.txt 文件作为情感词表，将分词遍历情感表并对每条答复意见进行评分：

$$第\ i\ 条答复意见总分：Sumscore = score[i] * word[i]　　（27）$$

根据 $Sumscore > 0$ 则为积极态度；$Sumscore < 0$ 则为消极态度；$Sumscore = 0$则为中性态度。

3. 结果分析

3.1 问题 1 结果分析

3.1.1 留言详情的一级分类数据分布图

图 1 留言详情的一级分类数据分布图

从饼图各分类比例来看，各分类数据分布没有特别大的差异，数据在各分类中分布比较均匀。

3.1.2 留言详情各一级分类词云图

各一级分类留言详情词云图如下所示：

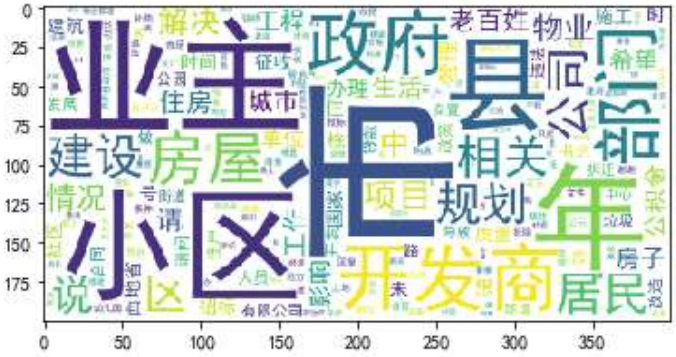

图 2 城乡建设词云图

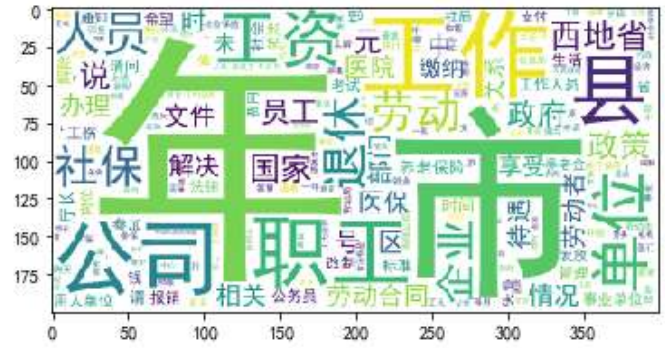

图 3 劳动和社会保障词云图

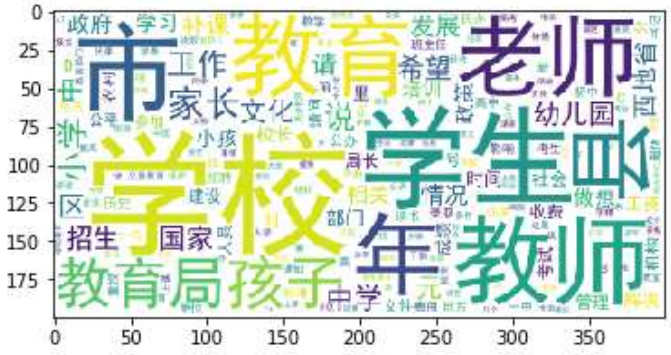

图 4 教育文体词云图

图 5 商贸旅游词云图

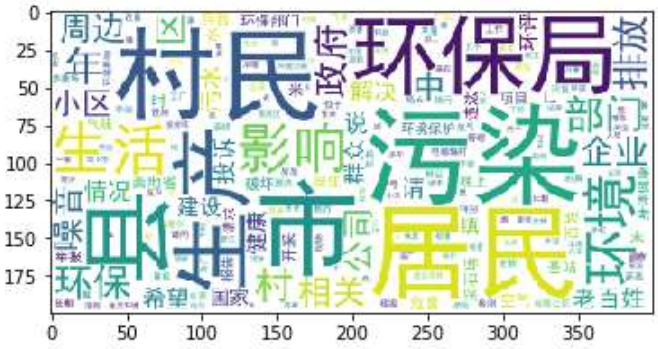

图 6 环境保护词云图

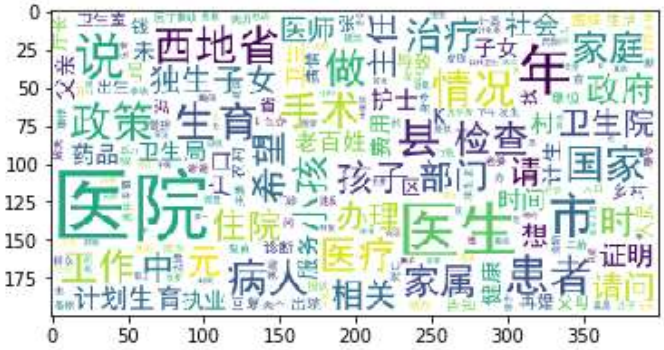

图 7 卫生计生词云图

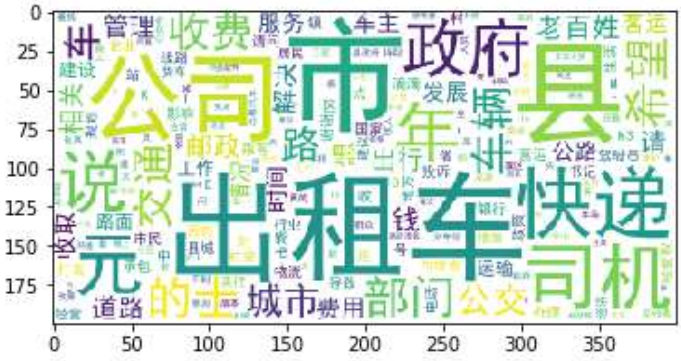

图 8 交通运输词云图

通过词云图的展示可以对留言详情各分类数据分词后的高频词予以视觉上

的强调突出效果，使阅读者更容易的获得各分类的主旨信息[2]。

3.1.3 多项式朴素贝叶斯分类模型

3.1.3.1 模型分类结果

通过对去重后的数据集进行分词去除停用词后，根据 TF-IDF 提取每条留言详情前 22 个关键词，再将数据集分成训练集和测试集，比例为 8：2.然后将分词通过 TF-IDF 转换为向量。

通过多项式朴素贝叶斯进行分类，模型分类评价如下：

训练集正确率：87.5984%

训练集AUC值：0.995809

测试集正确率：74.1579%

测试AUC值：0.981348

	precision	recall	f1-score	support
0	0.60	0.95	0.74	385
1	0.67	0.97	0.80	381
2	0.93	0.85	0.88	321
3	0.86	0.41	0.55	239
4	0.95	0.75	0.84	164
5	1.00	0.49	0.66	198
6	0.94	0.13	0.23	123
accuracy			0.74	1811
macro avg	0.85	0.65	0.67	1811
weighted avg	0.81	0.74	0.72	1811

图 9 拉普拉斯平滑系数为 1 时多项式朴素贝叶斯进行模型分类评测结果

AUC (Area Under Curve) 被定义为 ROC 曲线下的面积，显然这个面积的数值不会大于 1。又由于 ROC 曲线一般都处于 y=x 这条直线的上方，所以 AUC 的取值范围一般在 0.5 和 1 之间。使用 AUC 值作为评价标准是因为很多时候 ROC 曲线并不能清晰的说明哪个分类器的效果更好，而作为一个数值，对应 AUC 更大的分类器效果更好[5]。

通过上述评测结果，由准确率=0.74， 测试集和训练集的 AUC 值达到 98%以上，该分类器具有预测价值。并且前 6 类的的 f1 值都超过了 50%和查全率超过了 40%，各分类的查准率超过了 60%，因此在前 6 类分类中该模型分类结果还算不错；但是第 7 类的 f1=0.23＜0.5，虽查准率高达 94%，但查全率只有 13%，对第 7 类预测结果不是很好。因此模型还有待进一步优化。

3.1.3.2 模型改进

当我们在使用朴素贝叶斯算法去解决分类问题时，在训练集上进行训练时我

们可以发现有可能出现某些特征的概率 P 为 0 的情况，无论是在全文检索中某个字出现的概率，还是在留言详情分类中，这种情况明显是不太合理的，不能因为一个事件没有观察到就武断的认为该事件的概率是 0，拉普拉斯的理论支撑而拉布拉斯平滑处理正是处理这种情况下应运而生的[4]。

为了提升模型预测的价值，本模型通过不断调整拉普拉斯平滑系数来进一步优化模型预测结果。经过不断调整，通过交叉验证，最后统计出当 alpha=0.04 时，该模型的预测效果是比较好的，优化后模型的评测结果如下：

训练集正确率：99.4614%

训练集AUC值：0.999933

测试集正确率：84.7046%

测试AUC值：0.982385

	precision	recall	f1-score	support
0	0.79	0.87	0.83	385
1	0.82	0.94	0.88	381
2	0.92	0.90	0.91	321
3	0.81	0.72	0.76	239
4	0.88	0.93	0.91	164
5	0.96	0.78	0.86	198
6	0.83	0.59	0.69	123
accuracy			0.85	1811
macro avg	0.86	0.82	0.83	1811
weighted avg	0.85	0.85	0.84	1811

图 10 拉普拉斯平滑系数为 0.04 时多项式朴素贝叶斯进行模型分类评测结果

从优化后评测结果可以看出，该模型预测的准确率达到 85%，训练集和测试集 AUC 的值为 98%以上，该模型得到了进一步优化，使得更具有分类预测价值。训练集的正确率高达 99%，测试集的正确率高达 84.7%，该模型的分类结果比较可观。并且各分类下的查准率显著提高，都高达 79%；查全率也是提高了很多，每个分类的查全率都超过 59%。并且各 f1 值都在 69%以上。

综上所述，在对留言详情进行一级标签分类时采用 alpha=0.04 的多项式朴素贝叶斯算法得到的分类结果是具有一定的预测价值的。

3.1.3.3 模型评价

对留言详情进行一级标签分类的评价有以下方面：

（1） 节省存储空间，加快分类效率。该分类模型对每条留言提取了前 22 个关键词，降低了分类模型的维度，极大的节省了存储空间，不但提高了模型的精度，而且提高了程序运行的效率。

（2） 模型精度。选用多项式朴素贝叶斯算法进行留言详情一级标签分类，并通过不断调整拉普拉斯平滑系数提高模型精度，模型精度达到了 85%，具有较好的分类预测效果。而且通过 HMM 分词很好的解决了未登录词的识别问题，分词效果还是不错的。

（3） 但本模型在对分词效果还有待进一步改进，比如不能很好的解决分词的歧义问题和某些未登录词的分词效果。

3.2 问题 2 结果分析

通过对问题文本距离矩阵的聚类得到以下结果：

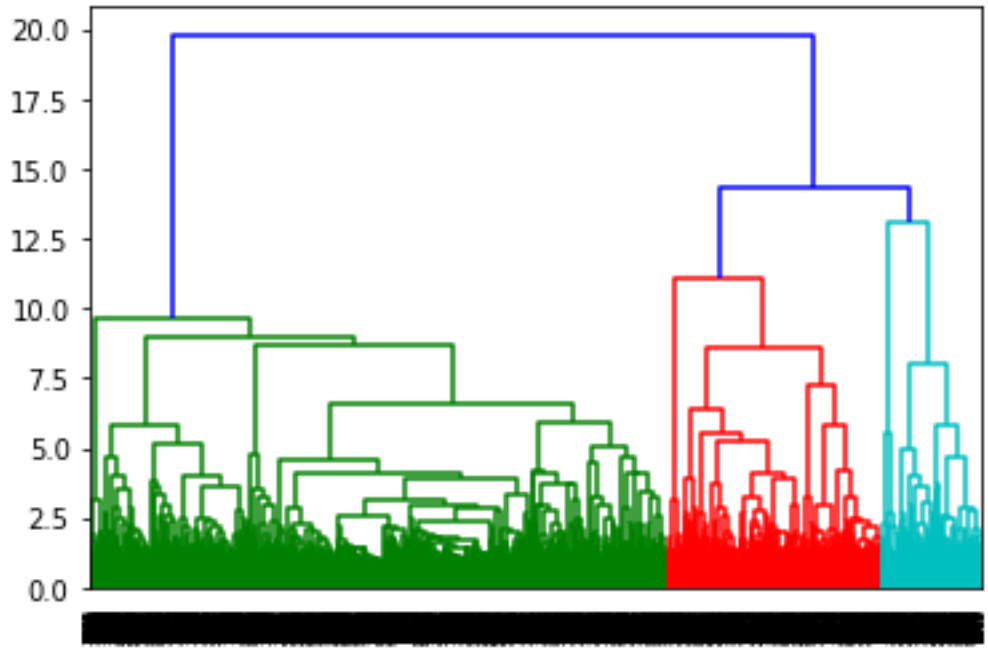

图 11 问题文本距离矩阵的聚类分析图

经过人工检验，在 fcluster 函数中参数 threshold 设置为 0.6 时聚类结果的正确率是最高的，一共有 3713 类，分类出错概率比较小，该聚类模型对问题分类起到了一定的作用价值，部分分类结果如下图所示：

ID		留言编号	留言用户	留言主题
101	5	265577	A0004787	咨询A市人才购房补贴通知问题
101	5	282104	A00090921	关于高级技师申报A市人才新政购房
101	5	224042	A00014225	咨询A市人才购房及购房补贴实施办
101	5	282248	A000106091	咨询A市人才购房补贴事宜
101	5	283494	A00085185	关于《A市人才购房及购房补贴实施
108	6	189180	A000106515	A市人才购房补贴申请是否与单位注
108	6	228559	A00082143	A市人才新政落户后屡次申请购房补
108	6	205771	A00020115	夫妻共同买的房为何申请A市人才购
108	6	244951	A00026039	反映A市人才租房购房补贴问题
108	6	270015	A00020115	为何我的A市人才购房补贴申请不通
108	6	289408	A0012413	在A市人才app上申请购房补贴为什么
113	4	232575	A0001717	咨询A市购房资格审核问题
113	4	259594	A00011365	咨询A市第二套房购房资格的问题
113	4	263429	A000108760	咨询A市独居老人的购房资格问题
113	4	281851	A000112932	咨询A市购房资格问题

图 12 问题文本经聚类部分分类结果展示

人工检测删除并整理一些数据，有可能会出现以下几种情况：

（1） 将内容相似的类整理为一组再赋予初始质量。

（2） 可能会出现同一人在短时间内反映内容相似的问题，不可视为热点问题，对其删除或整理。

最终提取热点得分前五的问题，结果如下所示：

热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述
1	959	24.97	2019/6/20至2019/9/12	A市A5区汇金路五矿万境K9县	小区群租房泛滥成灾，物业管理及其混乱
2	3	13.01	2019/7/7至2019/8/24	伊景园滨河苑广铁集团	捆绑车位销售
3	2687	11.9	2019/1/11至2019/7/8	A市58车贷特大集资	诈骗案犯罪分子逍遥法外
4	101	11.02	2019/11/51至2019/12/2	A市人才	人才购房补贴咨询
5	61	6.61	2019/8/9至2019/8/26	A市经开区泉星公园	优化公园项目规划

图 13 前五热点问题排序结果

排名第一的问题的点赞数高达 2097，说明这一问题受到的关注度比较高，该点赞数大大提高了热度指数。但这数据的真实性有待考察，杜绝恶意刷票的行为。

3.3 问题 3 结果分析

通过对答复意见与留言详情进行的相似度、完整性、可解释性、时效性、情感态度评分进行量化，构建答复意见评价模型。

因为要判断意见是否与留言相符，需要计算意见与留言的相似性，相似性越好，证明意见是根据留言进行具体回答的，很好的解决了客户的问题。所以答复意见得分与相似度成正比；完整性是考察答复时是否满足某一规范性，完整性越高，则说明答复时的流程比较好，条理框架清晰，所以答复意见得分与完整性成正比；而答复意见的时效性与答复意见得分成反比；与情感态度得分成正比。

于是构建答复意见评价模型如下：

答复意见得分=相似性得分+完整性的分-时效性得分+情感态度得分+可解释性得分 [28]

答复意见得分越高答复效果越好。由于可解释性和相似性还没量化出结果，所以本模型采取以下公式：

答复意见得分=完整性得分-时效性得分+情感态度得分 [29]

答复意见得分如下图所示：

时效性	情感评分	完整性评分	答复意见得分
0.748415377	37.46830654	4	40.71989116
0.72380994	25.80688571	3	28.08307577
0.725860393	51.42862991	2	52.70276952
0.725860393	56.4575484	2	57.73168801
0.770970361	20.92645403	3	23.15548367
1.52758755	29.70145398	4	32.17386643
2.013544932	26.70764389	4	28.69409896
1.402509912	103.0375705	6	107.6350606
0.797626251	68.59679918	6	73.79917293
0.797626251	22.66478361	2	23.86715736
3.47961889	50.85556083	8	55.37594194
1.498881207	54.74320794	5	58.24432674
0.791474892	18.81797798	3	21.02650309
0.287063432	6.100055288	3	8.812991856
0.848887578	15.95138743	3	18.10249986
3.385298047	65.97871651	8	70.59341847

图 14 答复意见评价得分

其中得分可参考附件 4。该模型结果可供参考但还有待进一步改进。

4. 结论

对“智慧政务”中进行分析研究，，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。因为近年来，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。

通过结果分析，本模型能很好的为留言详情进行自动划分归类。并且能挖掘出热点问题并对答复意见进行一定的评价，可以节省政府部门人工划分的时间，直接交给对应部门进行处理。并时刻关注社会热点问题，更好地为人民服务。

5. 参考文献

[1]孙海锋，郑中枢，杨武岳.网络招聘信息的数据挖掘与综合分析.北京林业大学.2017https://wenku.baidu.com/view/04c1a20ecdbff121dd36a32d7375a417866fc1c0.html

[2] 朴素贝叶斯算法基本原理. https://zhuanlan.zhihu.com/p/57554489

[3]机器学习中的 F1-score. https://www.cnblogs.com/yucen/p/9912063.html

[4] 皮皮猪 QAQ.朴素贝叶斯中拉普拉斯平滑算法.2019 https://blog.csdn.net/qq_39037383/article/details/89363141

[5] 京局京段蓝白猪.准确率、精确率、召回率、F1 值、ROC/AUC 整理笔记.2018 https://blog.csdn.net/u013063099/article/details/80964865

[6] 使用聚类分析算法对文本分类（分类数 k 未知） https://blog.csdn.net/leaf_zizi/article/details/82684921?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-4&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-4

[7] 层次聚类算法的原理及 python 实现 https://blog.csdn.net/u012328476/article/details/78978113

[8] 计算内容热度的算法解释 https://blog.csdn.net/weixin_34019929/article/details/91412348?ops_request_misc=%257B%2522request%255Fid%2522%253A%252215889176151972483923045 6%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=158891761519724839230456&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v25-4-91412348&utm_term=%E5%86%85%E5%AE%B9%E7%83%AD%E5%BA%A6

[9] Python 文本挖掘：使用 gensim 进行文本相似度计算 https://blog.csdn.net/chencheng126/article/details/50070021?utm_source=app