基于“智慧政务”中留言的数据挖掘分析

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。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。本文根据已收集的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，利用自然语言处理和文本挖掘对其进行分析。

对于问题 1，通过 R 软件中的 jieba 包对留言详情进行分词并去除停用词，再利用向量空间模型（VSM）进行文本特征表示，其中每条留言详情的的权重向量用 TF-IDF 来计算。文本向量化后取词频排名前 1500 的特征词作为解释变量，一级标签作为因变量，进行建模。运用支持向量机、决策树、boosting、bagging 以及随机森林来对留言进行分类，五种分类方法的效果都非常好，其中决策树耗时最短且测试集的 F1-Score 达到 99.9%。

对于问题 2，利用与问题 1 一样的方法对留言主题进行特征表示，文本向量化后取词频排名前 1000 的特征词进行聚类，提出了一种新的聚类方法——“特征词—文本匹配”算法。聚类之后，选择每一类的留言总数、反对总数、点赞总数作为评价指标，利用熵权法确定权重，将加权和作为热度综合评价指标，加权和越大，则热度越高。

对于问题 3，选取相关性、完整性、及时性作为评价指标，其中根据留言详情与答复意见之间的文本相似度来衡量相关性，而文本相似度则利用 SimHash 算法和汉明距离来计算。对这三个指标量化后求加权和作为质量评价指标，权数用熵权法来确定，加权和越大，则答复意见的质量越高。

关键词：文本预处理；文本分类；文本聚类；评价指标；文本相似度

Data mining analysis based on comments in intelligent government affairs

Abstract: In recent years, WeChat, weibo, mayor's mailbox, sunshine hotline and other online political platforms have gradually become an important channel for the government to understand public opinion, pool people's wisdom and pool people's spirit. The increasing amount of text data related to various social situations and public opinions has brought great challenges to the work of relevant departments which mainly relied on manual workers to divide messages and sort out hot topics. At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of intelligent government affairs system based on natural language processing technology has become a new trend of social governance innovation and development, which plays a great role in promoting the management level and efficiency of the government. In this paper, natural language processing and text mining are carried out on the basis of the collected records of people's political messages from open sources on the Internet and the replies of relevant departments to some people's messages.

To solve the first problem,the message details are segmented and the stop words are removed through the jieba package in the R software, and then Vector Space Model (VSM) is used to represent the text features. The weight vector of each message detail is calculated by TF-IDF. After the text is vectorized, the top 1500 feature words in the word frequency ranking are used as explanatory variables, and the first-level labels are used as dependent variables for modeling. Using Support Vector Machine, Decision Tree, boosting, bagging, and Random Forest to classify messages, the five classification methods are very effective. Among them, the Decision Tree takes the shortest time and the F1-Score of the test set reaches 99.9%.

To solve the second problem, feature the message subject in the same way as in question 1. After the text is vectorized, the feature words with the top 1000 word frequency ranking are clustered. A new clustering method is proposed-"feature words-text matching" algorithm. After clustering, the total number of comments, total number of objections, and total number of likes of each category are selected as evaluation indicators, and the weight is determined using the entropy weight method. The weighted sum is used as the comprehensive evaluation index of heat. The larger the weighted sum, the higher the heat.

To solve the third problem, relevance, completeness, and timeliness are selected as evaluation indicators. The relevance is measured based on the text similarity between the details of the message and the reply opinion, and the text similarity is calculated using SimHash algorithm and Hamming distance. The three indicators are quantified and the weighted sum is used as the quality evaluation index. The weight is determined by the entropy weight method. The larger the weighted sum, the higher the quality of the response.

Key words: Text Preprocessing; Text Classification; Text Clustering; Evaluation Index; Text Similarity

目录

1.挖掘目标

本次数据挖掘通过 R 软件实现，通过对群众问政留言记录，及相关部门对部分群众留言的答复意见，对文本数据进行预处理。对处理后的数据根据留言主题进行分类，并且找出热度较高的留言内容进行可视化呈现。并且，从留言答复意见相关性、完整性、可解释性三个角度去构建模型形成一套答复意见评价方案。最终实现智慧政务的目标。

2.分析方法与过程

2.1 总体流程

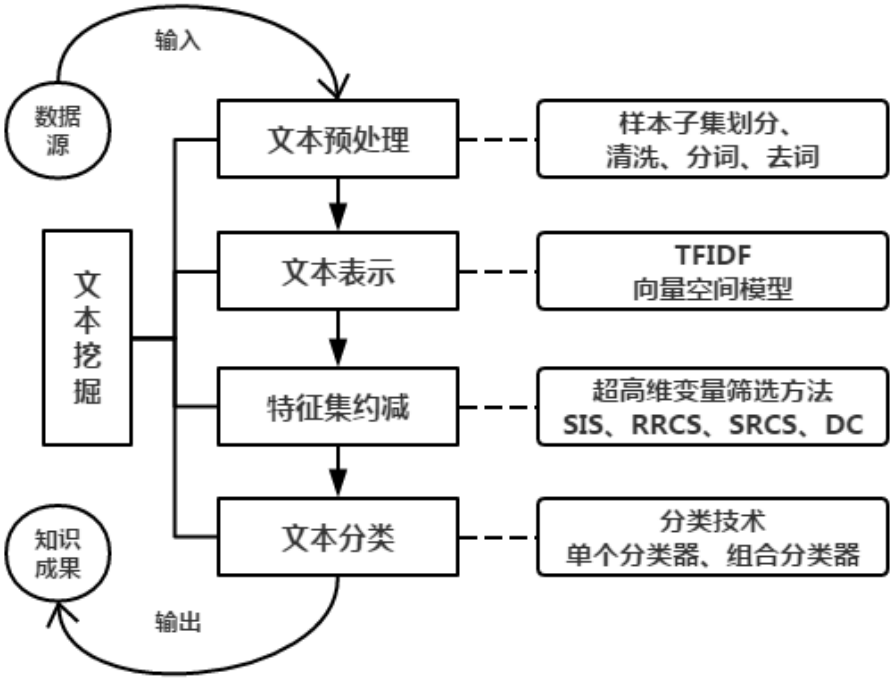

图 1 文本挖掘流程图

文本挖掘(薛为民等(2005)[1])一般包括文本预处理、文本表示、特征集约减和文本分类四个步骤.

第一步：对已有数据进行文本预处理，包括样本自己划分、清洗、分词、去词等操作。

第二步：对预处理的文本数据建立向量空间模型进行文本表示。

第三步：通过多种高位变量筛选的方法进行特征集越减。

第四步：对处理之后的数据进行文本分类，这里包括单个分类器和组合分类器。

2.2 具体步骤

2.2.1 数据介绍

本文使用的实验数据为已给出的互联网公开来源的群众问政留言记录，记录主要包括留言的三级分类、主题、时间、详情，以及对应的一级标签。另外，还提供了网络上群众对留言的反对数和点赞数。最后，相关部门对部分群众留言的答复意见也详细给出。

2.2.2 文本预处理

文本预处理是文本挖掘的第一个环节，也是关键的一环，它是文本挖掘其他环节的基础和前提，如图 2 文本预处理包含样本子集划分、数据清洗、分词和去词四项任务。

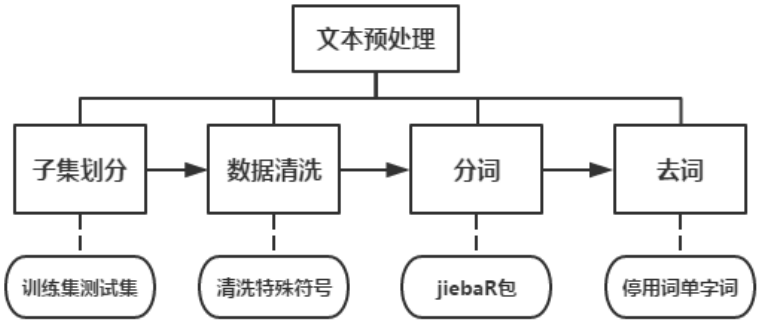

图 2 文本预处理步骤

文本预处理的第一步就是划分样本子集，如果我们用所有的数据进行文本挖掘，由于将残差等信息全都拟合在模型中，那么对于当前数据我们可能在所有样本集的基础上得到一个很好的分类模型，分类效果好且错误率低，但这可能存在模型的过拟合问题，当把模型应用到一个新的数据上时，由于出现新的残差，模型的分类效果可能并不理想。为了得到相对准确的误差估计以及避免模型出现过拟合，在建立模型前，往往将样本随机划分为两部分，训练样本集和测试样本集，训练样本集用于训练和拟合模型，测试样本集用于评估模型和调整模型。我们利用简单随机抽样的方法，抽取样本的 70%作为训练样本集，剩下 30%作为测试样本集。

由于附件 2 中给出的留言文本偏向于口语化，内容多样，里面包含一些数字、空格、英文字母和“#”、“！”、“，”、“【】”、“：”、“( )”等特殊符号，然而这些内容对文本分类并没有帮助，且会影响后续的分词效果，因此我们通过数据清洗将其清洗掉。此外，中文文本不同于英文文本，英文文本单词之间有固定的间隔符，可以通过间隔符来进行分词，而中文文本词与词之间是连在一起的，那么就需要利用算法和模型来将词与词分开。目前，在 R 语言中已经开发了专门做中文文本分词的 R 包，Rwordseg 包和 jiebaR 包。Rwordseg 包安装过程繁琐且需要配置 java 环境，而 jiebaR 包安装简单，运行所需内存更小，速度更快，且功能比 Rwordseg 包更全面。因此，我们基于 R 操作平台，利用 jiebaR

包来进行中文文本分词。在经过分词处理后，可以发现在分好的词中包含许多例如“我们”、“的”“可以”、“了”、“虽然”、“一定”这样的非特征词汇，这样的词被称为“停用词”。停用词不仅对文本分类毫无作用，且当数据量较大时会严重增加计算机计算量和内存占用率，降低计算机运行效率，因此需要做去停用词处理。除去停用词外，还有大量的只含一个字的词语，对于中文来讲，一个字可能有多个意思，因此它所代表的信息并不明确，而且所含信息量太少，从而需要把所有词语中的单字词也去掉。

在实际操作中，基于 R3.4.3 中的 jieba 包对文本进行预处理，去掉数字、特殊符号、英文小写字母、停用词和单个字

摘取出其中一条留言的分词效果，如图 3 所示：

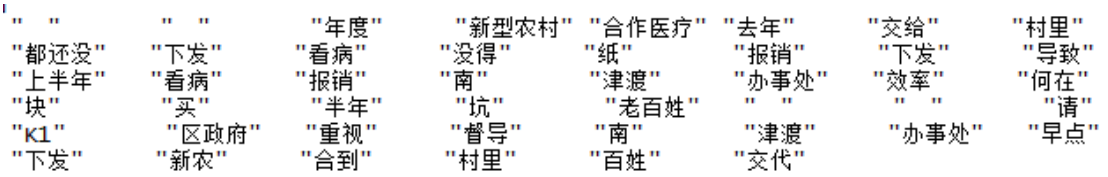

图 3 起初分词效果

可以看到，分词效果并不好，有些留言分词明显错误，破坏了句子的结构，如“新农”，“合到”应该划分成“新农合”才恰当；而且里面仍存在许多意义不大、对分类没有任何帮助的词，如“没得”、“请”等。

因此我们改进分词方法：

1.增添用户词典，在用户词典中添加我们自己的词库。

2.扩充停用词词典，在停用词词典中添加停用词。

3.对划分的词语贴词性标签。

给出第一条留言重新分词的结果，如图 4：

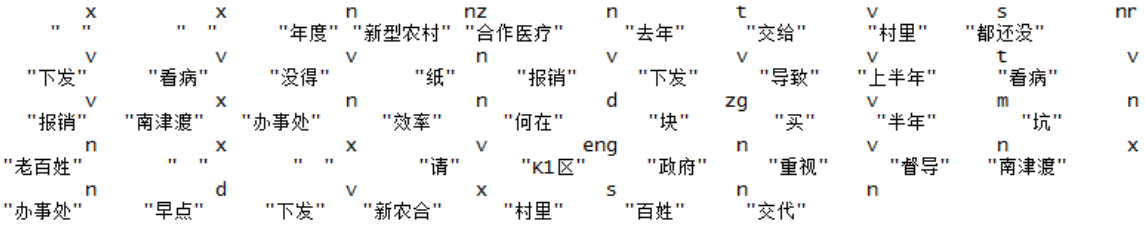

图 4 改进后的分词效果

在本次实验中，首先对文本数据的预处理中去掉数字、特殊符号、英文字母、单个字、停用词，在进行分词，并且通过去掉叹词、数词、量词、时间词、声词、介词、助词、副词、连词、代词、副形词、副动词等对词语贴上词性标签，改进了分词效果，便于下面的进一步分析。

分词后每个词都有了词性标签，我们把叹词 e、数词 m、量词 q、时间词 t、拟声词 o、介词 p、助词 u、副词 d、连词 c、代词 r、副形词 ad、副动词 vd 等对分类明显没有帮助的词去掉。

2.2.3 文本表示

在经过文本预处理后，得到的文本特征词是人类可以读懂的自然语言，然而目前计算机并不能像人类一样可以直接读懂文字，这就需要对文本进行表示(阳小兰等(2010)[2])使其转化成计算机可以识别的形式。

在文本表示的模型中，向量空间模型(VSM)的应用较多且效果较好(庞剑锋等(2001)[3])。向量空间模型的基本思想是使用向量表示文本，假定不需要考虑词的先后顺序，文本的特征词集合$T=\{t_1,t_1,\cdots,t_p\}$，令文本中的每个特征词t_i对应特征空间的一维，从而将文本表示成欧式空间中一个p维向量$X=(X_1,X_2,\cdots,X_p)$。为了体现每个特征词在文本中的重要程度，通常要对特征词赋予权重，在特征词权重指标中比较经典的是 Salton G 等(1973)[4]提出的 TFIDF 指标。TFIDF 的主要思想是：如果某个词在某文本中出现的频率高，但在其他文本中出现的频率低，那么这个词就适合用来将不同的文本区分开。TFIDF 指标的计算主要涉及两个部分，词频(TF)和逆文本频率(IDF)。

词频(TF)指某个词语在某文件中出现的次数，定义词语t_i在文件d_j中的词频为

$$TF_{ij}=n_{ij}, \tag{1}$$

其中，n_{ij}表示该词在文件d_j中出现的次数。

逆文本频率是一个词语普遍重要性的度量，某一词语t_i的逆文本频率为

$$IDF_i=\log\frac{|D|}{\left|\left\{j:t_i\in d_j\right\}\right|}. \tag{2}$$

其中，$|D|$表示文件总数，$\left|\left\{j:t_i\in d_j\right\}\right|$表示包含词语$t_i$的文件数，为了避免分母为零，一般情况下使用$\left|\left\{j:t_i\in d_j\right\}\right|+1$。

在实际应用中通常将 TF 和 IDF 指标联合使用，于是得到 TFIDF 指标

$$TFIDF_{ij}=TF_{ij}\times IDF_i\ . \tag{3}$$

字词的重要性随着它在文本中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降，因此词频 TF 越高、逆文本频率 IDF 越高，即 TFIDF 值越高，字词在该文本中越重要。

在计算出 TFIDF 后，对特征词赋予 TFIDF 指标权重，令$x_{ij}=TFIDF_{ij}$，于是初始文本便可以表示成下列易于处理的形式

$$X=(X_1,X_2,\cdots,X_p)=\begin{pmatrix} x_{11} & x_{21} & \cdots & x_{p1} \\ x_{12} & x_{22} & \cdots & x_{p2} \\ \cdots & & & \\ x_{1n} & x_{2n} & \cdots & x_{pn} \end{pmatrix}. \quad (4)$$

在进行数据处理时，由于分词之后的词量非常庞大，为了便于后面的分析，在这里我们对词频进行了排序，选取了词频排名前 1000 个词作为重要变量。

2.2.4 特征集约减

由于数据量较大，文本的特征词集合$T=\{t_1,t_1,\cdots,t_p\}$中的特征词数已达上万个。然而这上万个特征词并不是每一个都对文本分类有比较大的贡献，各个类别中普遍存在的特征词对文本分类的贡献小，而在某个类中出现次数多，在其他类出现次数少的特征词对文本分类的贡献大。其次，利用上万个特征词对文本进行分类会严重占用计算机内存，降低运行效率。因此，我们需要筛选出重要的特征词来对特征集进行约减。

令文本向量化后得到的$X_1,X_2,\cdots,X_p$作为预测变量，并定义目标变量$Y=\left(Y_1,Y_2,\cdots,Y_n\right)^T$：

$$Y_j=\begin{cases} 1,\ d_j\in D_1; \\ 2,\ d_j\in D_2; \\ 3,\ d_j\in D_3; \\ 4,\ d_j\in D_4; \\ 5,\ d_j\in D_5; \\ 6,\ d_j\in D_6; \\ 7,\ d_j\in D_7. \end{cases} \quad (5)$$

其中，d_j表示第j条留言，$j=1,2,\cdots,n$，n为留言总数，D_1,D_2,D_3,D_4,D_5分别表示“城乡建设”、“环境保护”、“交通运输”，“教育文体”、“劳动和社会保障”、“商贸旅游”、“卫计卫生”相关的留言集合。我们要做的是在预测变量$X_1,X_2,\cdots,X_p$中筛选出对目标变量Y有显著影响的重要变量。

针对超高维数据的变量筛选问题，边际筛选是目前比较认可的方法。边际筛选将超高维问题化为低维问题，能够有效降低整体的计算复杂度。目前有许多基于相关性的边际筛选方法，预测变量与目标变量的相关性越强，则该预测变量越重要。

Fan and Lv（2008）提出SIS方法，该方法基于Pearson相关将与因变量相关性弱的协变量筛选掉，可以快速有效地把维数p从大规模或巨规模(比如

$\exp(O(n^{\xi})), \xi > 0$ 降低到相对大规模 (比如 $o(n)$)。

对线性回归模型

$$Y = X\beta + \varepsilon, \tag{6}$$

考虑参数 β 的估计问题，其中，X 为 $n \times p$ 回归设计矩阵，$Y = (Y_1, Y_2, \cdots, Y_n)^T$ 为因变量向量，$\beta = (\beta_1, \cdots, \beta_p)^T$ 为待估参数向量，$\varepsilon = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)^T$ 为独立同分布的随机误差，并与 X 独立。

令 $M_* = \{1 \le i \le p : \beta_i \ne 0\}$ 为真实稀疏模型的指标集，$s = |M_*|$ 表示 M_* 中元素的个数，也就是真实模型中回归系数不为 0 的个数。

令 $\omega = (\omega_1, \cdots, \omega_p)^T = X^T Y$，这里 ω 为因变量与预测变量的边际相关变量（ω_i 表示第 i 个预测变量与因变量的边际相关），此处的 X 已被标准化。

对于任意给定的 $d_n < n$，对向量 ω 的 p 个分量绝对值进行降序排列，定义子模型

$$M_{d_n} = \{1 \le i \le p : |\omega_i| \text{在前第} d_n \text{个中}\}. \tag{7}$$

即把全模型指标集 $\{1, \cdots, p\}$ 压缩到大小为 $d_n < n$ 的子模型指标集 M_{d_n}。

在SIS方法中，由于对预测变量做了标准化处理，因此可将 $\omega_k = X_k^T Y$ 看作因变量 Y 与预测变量 X_k 之间的Pearson相关系数

$$\rho_k = \frac{\mathrm{cov}(X_k, Y)}{\sqrt{\mathrm{var}(X_k)}\sqrt{\mathrm{var}(Y)}}. \tag{8}$$

我们采用SIS方法进行特征词约减，取筛选变量个数 $d_n = [n / \log(n)]$ (Fan and Lv(2008))，其中 $[\bullet]$ 表示取整。

2.2.5 文本分类

目前，常见的分类技术有决策树、支持向量机、神经网络等单个分类器，以及 boosting、bagging 和随机森林等组合分类器(周志华(2016)[5])，一般来讲，组合分类器的分类效果要比单个分类器的分类效果好，下面对这六种分类器做一个简单介绍。

2.2.5.1 决策树（Tree）

决策树是一个树结构，由内部节点、叶节点和有向边组成，内部节点表示一个特征或属性，叶节点表示一个类。决策树在进行分类的时候，从根节点开始，对分类项中相应的特征属性进行测试，根据测试结果选择输出分枝，不断向下移动，直到到达叶节点，叶节点存放的类别即为最终分类结果。

CART 树是决策树的一种，它既可以做分类也可以做回归，在分类时，它采用基尼系数来决定子数据集生成的决策树的拓展形，其数学定义为：

$$Gini(\mathrm{S}) = 1 - \sum_{j=1}^{k} p^2(j \mid t). \tag{9}$$

其中，t 表示节点，k 表示输出变量的类别个数，$p(j \mid t)$ 代表节点 t 中输出变量取第 j 类的归一化概率。

CART 树在面对诸如存在缺失值、变量数较多等问题时，显得非常稳健，训练速度快，但容易出现模型过拟合问题。

2.2.5.2 支持向量机（SVM）

支持向量机(SVM)分类的基本思想是，基于训练集在样本空间中找到一个最优决策超平面，使其能够将不同类别的样本区分开。支持向量机把分类问题转化为寻找分类平面的问题，并通过最大化分类边界点距离分类平面的距离来实现分类。

在样本空间中，划分超平面可通过如下线性方程来描述：

$$\omega^T x + b = 0, \tag{10}$$

其中，ω 为法向量，用来表示超平面的方向，b 为位移项，用来表示超平面与原点之间的距离，x 表示样本空间中任一点。

从而支持向量机的基本型可写为

$$\min_{\omega,b} \frac{1}{2}\|\omega\|^2 \quad s.t. \;\; y_i(\omega^T x_i + b) \geq 1, \;\; i = 1, 2, \cdots, m. \tag{11}$$

支持向量机能够很好地处理小样本情况下的高维问题，可以避免维数灾难。但在不加修改的情况下只能做二分类问题，对缺失数据也比较敏感。

2.2.5.3 Boosting

Boosting 作为一种组合分类器，它可以将弱学习器提升为强学习器。Boosting 算法的思想是，先赋予训练集中每个样本相同的权重，然后进行迭代，迭代后，对分类错误的样本加大重采样权重。这样的迭代进行 T 次，于是得到 T 个模型。

对于新样本 x，每个模型都给出一个预测值 $h_t(x)$，每个预测值都有一个权重：

$$W_t(x) = -\ln\left(\frac{e(t)}{1-e(t)}\right), \tag{12}$$

其中，$e(t)$ 表示第 t 个模型的误差。对每个预测类别计算权重的总和，总和最高的即为最终分类结果。

Boosting 可以使用多种方法构建子分类器，方法简单且分类精度高，但容易受到噪声影响，算法运行速度也比较慢。

2.2.5.4 Bagging

Bagging 也是一种组合分类器，但与 Boosting 工作机制不同。首先，它基于自助采样法(bootstrap)从原始训练集中抽取 T 个样本集。然后，对每个样本集训练出一个基学习器，得到 T 种分类结果。最终，根据 T 种分类结果对每个记录进行投票，决定其最终分类。

Bagging 的输出函数为：

$$H(x) = \arg\max \sum_{t=1}^{T} I(h_t(x) = y), \tag{13}$$

其中，$h_t(x)$ 表示第 t 个基学习器的输出函数。

Bagging 算法的泛化能力很强，可以有效降低模型的方差，但对训练集的拟合程度可能会差一些。

2.2.5.5 随机森林（RF）

随机森林是 Bagging 算法的扩展，它的思想仍是 Bagging，但是进行了独特的改进。首先，随机森林使用了 CART 树作为基学习器。其次，它改进了树的建立方式，先从决策树节点的属性集合中随机选择一个包含 k 个属性的子集，然后再从这个子集中选择一个最优属性用于划分。由于随机森林训练可以高度并行化，因此在处理很大的数据时依旧非常高效，但在某些噪声比较大的样本集上，随机森林模型容易出现过拟合问题。

根据下列算法而建造每棵树：

1.用 M 来表示训练用例（样本）的个数，N 表示特征数目。

2.输入特征数目 n，用于确定决策树上一个节点的决策结果；其中 n 应远小于 N。

3.从 M 个训练用例（样本）中以有放回抽样的方式，取样 k 次，形成一个训练集（即 bootstrap 取样），并用未抽到的用例（样本）作预测，评估其误差。

4.对于每一个节点，随机选择 n 个特征，每棵决策树上每个节点的决定都是基于这些特征确定的。根据这 n 个特征，计算其最佳的分裂方式。

5.每棵树都会完整成长而不会剪枝，这有可能在建完一棵正常树状分类器后会被采用。

6.最后测试数据，根据每棵树，以多胜少方式决定分类。

在构建随机森林时，需要做到两个方面：数据的随机性选取，以及待选特征的随机选取，来消除过拟合问题。

首先，从原始的数据集中采取有放回的抽样，构造子数据集，子数据集的数据量是和原始数据集相同的。不同子数据集的元素可以重复，同一个子数据集中的元素也可以重复。第二，利用子数据集来构建子决策树，将这个数据放到每个子决策树中，每个子决策树输出一个结果。最后，如果有了新的数据需要通过随机森林得到分类结果，就可以通过对子决策树的判断结果的投票，得到随机森林的输出结果了。假设随机森林中有 3 棵子决策树，2 棵子树的分类结果是 A 类，1 棵子树的分类结果是 B 类，那么随机森林的分类结果就是 A 类。

与数据集的随机选取类似，随机森林中的子树的每一个分裂过程并未用到所有的待选特征，而是从所有的待选特征中随机选取一定的特征，之后再在随机选取的特征中选取最优的特征。这样能够使得随机森林中的决策树都能够彼此不同，提升系统的多样性，从而提升分类性能。

优点：随机森林的既可以用于回归也可以用于分类任务，并且很容易查看模型的输入特征的相对重要性。随机森林算法被认为是一种非常方便且易于使用的算法，因为它是默认的超参数通常会产生一个很好的预测结果。超参数的数量也不是那么多，而且它们所代表的含义直观易懂。随机森林有足够多的树，分类器就不会产生过度拟合模型。

缺点：由于使用大量的树会使算法变得很慢，并且无法做到实时预测。一般而言，这些算法训练速度很快，预测十分缓慢。越准确的预测需要越多的树，这将导致模型越慢。在大多数现实世界的应用中，随机森林算法已经足够快，但肯定会遇到实时性要求很高的情况，那就只能首选其他方法。当然，随机森林是一种预测性建模工具，而不是一种描述性工具。

2.2.6 相关指标定义

2.2.6.1 热度指标

如果某个问题很多人都反映，那么这个问题的热度就较高。首先进行聚类，我们选择类留言总数、类反对总数、类点赞总数作为评价指标，利用熵权法得到这三个指标的权重，建立热度综合评价指标 hot-score 为类留言总数、类反对总数、类点赞总数的加权和，hot-score 值越大，热度越高。

其中涉及到的熵权法是指在给定评价对象集后各种评价指标值确定的情况下，各指标在竞争意义上的相对激烈程度，从信息角度考虑，它代表该评价指标在该问题中提供有效信息
量的多寡程度，作为一种客观综合评价方法，它主要是根据各指标传递给决策者的信息量大小来确定其权数。

假设在 m 项指标和 n 个被评价对象中，原始评价矩阵为 Dnm，标准化处理后的矩阵为 Rnm，根据熵权法理论，由式 ： $H_j = k\sum f_{ij} \ln f_0$， $j = 1,2,\cdots,m$

计算得第 j 项指标的熵值，其中 $k = \frac{1}{\ln n}$， $f_{ij} = \frac{R_{ij}}{\sum_{i=1}^{n} R_{ij}}$。当 $f_{ij} = 0$ 时，$f_{ij} \ln f_{ij} = 0$。

由式 $\omega_j = \frac{1 - H_j}{m - \sum_{j=1}^{m} H_j}$， $j = 1,2,\cdots,m$， $0 \le \omega_j \le 1$ 且 $\sum_{j=1}^{m} \omega_j = 1$ 计算得第 j 项的熵权。

2.2.6.2 相关性指标

通过计算附件 4 里留言详情与答复意见之间的文本相似度来衡量相关性，文本相似度越高，则说明答复的相关性越大。本文利用 SimHash 算法对留言详情和答复意见进行降维，分别生成 SimHash 值，通过 SimHash 值来计算二者之间的汉明距离，然后根据汉明距离来比较留言与答复之间的相似度。下面介绍一下 SimHash 算法和汉明距离。

SimHash 算法[9]是谷歌在 2007 年发表的论文《Detecting Near-Duplicates for Web Crawling》中提到的一种指纹生成算法，被应用在谷歌搜索引擎网页去重的工作之中。SimHash 算法分为 5 个步骤：分词、Hash、加权、合并、降维，引用一张图片来描述整个流程：

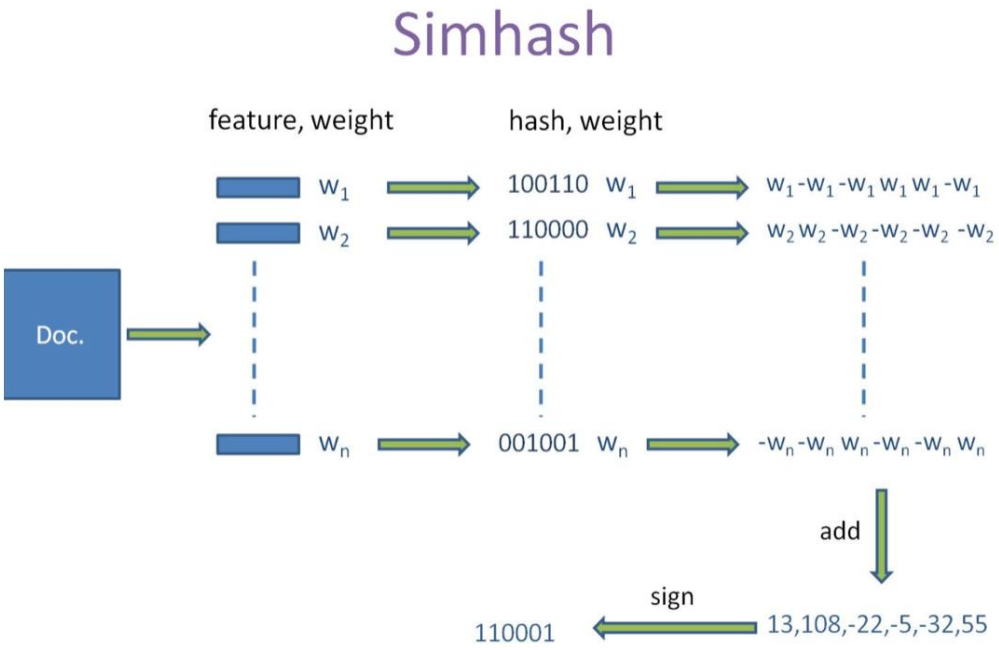

图 5 SimHash 算法流程

下面讲述 SimHash 算法的具体过程：

分词：对文本进行分词，并提取关键词，假设产生个特征词，为每一个特征词赋予权重，权重越大，代表该特征词在文本中越重要。这样就得到了一个文本的特征词向量和每个特征词对应的权重。

Hash：通过 Hash 函数计算每个特征词的 Hash 值，将词转化为二进制的字符串。

加权：上一步得到的特征词的二进制字符串中等于 1 的则用 1 乘以该特征词的权重，二进制字符串中等于 0 的则用-1 乘以该特征词的权重，按此操作得到权重向量。比如某个特征词的 Hash 值为 101011，权重为 2，那么它的权重向量为[2，-2，2，-2，2，2]。

合并：对一个文本中的每个特征词进行上一步的加权，计算出权重向量，把一个文本里所有特征词的权重向量累加，得到一个新的权重向量。

降维：对于合并得到的权重向量，大于 0 则记为 1，小于等于 0 则记为 0，从而得到文本的 SimHash 值。比如合并之后得到的文本的权重向量为[7，5，-2，

1，-3，6]，则该文本的 SimHash 值即为[1，1，0，1，0，1]。
利用上述 SimHash 算法得到文本的 SimHash 值，接着使用汉明距离来判断两个文本的相似度。所谓汉明距离，即两个相同长度的字符串对应位置上不同字符的个数，比如[1，1，0，1，0，1]与[1，1，0，0，0，0]间的汉明距离为 2。若汉明距离小于 3，则认为两个文本是相似的。

本文在 SimHash 算法的分词阶段对每条留言详情和答复意见的文本都提取 10 个关键词，按上述步骤操作得到所有文本的 SimHash 值。最后通过留言详情与其对应的答复意见的 SimHash 值来计算二者之间的汉明距离，从而根据汉明距离来判断两者的相似度，汉明距离越小，则说明留言与答复的相似度越高。为了将三个指标统一成正向指标，本文用汉明距离取负数，来衡量相关性。

2.2.6.3 完整性指标

用处理过后的答复句子的长度来衡量完整性，长度越大，则说明答复的完整性越高。这里一个中文字符占两个字节，算两个长度。

2.2.6.4 及时性指标

在关于答复评价体系中，回复的时间也是衡量的标准，所以加上了及时性指标。通过计算答复时间与留言时间的差值来衡量及时性，为了统一成正向指标，用（留言时间—答复时间）这一度量来衡量及时性，该值越大则说明答复的及时性越好。

2.3 结果分析

有了以上的理论基础，现在对结果进行分析。

2.3.1 对留言详情的分类分析结果

在对留言详情数据进行预处理之后，计算所有词语的词频，并对词频进行排序，取词频排名前 100 的词语做词云图，如图 6：

图 6 留言详情词频前 100 的词

从留言详情词频前 100 的词中可以看到，涉及到反映公司、政府、学校、工作问题的留言较多，是大家关注的焦点。

确定解释变量 X，这里是选取词频排名前列的词作为重要变量，而因变量 Y，则是需要进行识别对应的一级标签。将所有的数据进行分割，70%的数据量为训练集，其余的 30%为测试集。由于选择的词量不同可能会影响后面的分类结果，所以分别选取了词频排名前 1000 的词和排名前 1500 的词进行分类。

以下表 1 为词频前 1000 的词在不同分类方法下的分类结果。

表 1 一级标签分类结果

分类方法	训练集正确率	测试集正确率
神经网络模型	21.38979%	20.26782%
SVM	83.32558%	70.93739%
Tree	52.67566%	51.75534%
boosting	77.36932%	65.32754%
bagging	70.91671%	60.62251%
RF	96.60307%	75.82338%

可以从表 1 中看到，以上的六种分类方法效果都不太好，其中随机森林的方

法相比较而言是最好的，测试集的正确率达到了 75.82338%。

对留言详情进行分类之后，还需要对不同的分类方法进行评价，这里运用的是 F-Score 来进行评价的。

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

其中 Pi 为第 i 类的查准率，Ri 为第 i 类的查全率。

以下表 2 为词频前 1500 的词在不同分类方法下的 F1score 结果。

表 2 F1score 结果

分类方法	训练集 F1score	测试集 F1score	运行时间
SVM	1	0.9063164	42.90s
Tree	1	0.9992569	4.13s
boosting	1	0.9992569	118.39s
bagging	1	0.9992569	202.32s
RF	0.99972	0.9851783	218.92s

可以从表 2 中看到，以上的五种分类方法效果都非常好，通过增加词量的训练，整体分类效果得到了大幅度提升，决策树、boosting、bagging 以及随机森林的测试集 F1sore 都达到了 0.98 以上。

2.3.2 对热点问题的探究分析结果

任务一：对留言进行聚类

由于留言主题是对留言详情的梗概，因此我们采用留言主题来对留言进行聚类。首先，对分词之后的数据进行词频统计，通过从大到小的排序，选取前 100 个词绘制词云图，如图 7：

图 7　留言主题词频前 100 的词云图

从图 7 中，我们可以看到，大多数留言是反映 A 市及各个小区的，主要涉及到业主投诉噪音扰民及居民的一些建议和咨询。

在对留言主题分词后，我们可以发现，在词语“小区”的前面便是小区的名字，

在“街道”的前面便是街道的名字，因此，我们很容易提取出小区名、村庄名、公园名、社区名、镇名、乡名、街道名、路名、学院名等，我们取这些词作为特征词，考虑到特定人群，我们在特征词中加入"幼儿园","小学","中学","学院","大学","学生","学校","村民","职工","教师","消费者","医院","有限公司","业主","居民","开发商","政府","农民工"等词，最终得到 k 个特征词，然后利用“特征词—文本匹配”算法对留言进行聚类。

具体算法如下：

第一步：对于第一个特征词 word1，若第 i 条留言中有“word1”，则我们把它归为“word1”类。

第二步：所有留言去掉“word1”类留言，在剩下的留言中对第二个特征词 word2 继续聚类，若第 i 条留言中有“word2”，则我们把它归为“word2”类。

第三步：一直到第 k 个特征词，聚类结束。

聚类后发现，有些类的留言个数很大，且类中的留言并不是同一类的事件，因此，我们再次选择词频大于 2 的词作为特征词，在每一类中使用“特征词—文本匹配”算法对留言进行再次聚类。

详细的聚类结果可见附件：第二问留言聚类结果。

任务二：热度评价

在聚类之后，关于热度评价指数，选择类留言总数、类反对总数、类点赞总数作为评价指标，利用熵权法得到这三个指标的权重，建立热度综合评价指标 hot-score 为类留言总数、类反对总数、类点赞总数的加权和，hot-score 值越大，热度越高,将 hot-score 从大到小排序，得到前五个热点问题。熵权法中类留言总数、类反对总数、类点赞总数的权重，见表 3：

表 3 热度评价指数中的权重

变量	类留言总数	类反对总数	类点赞总数
权重	0.08933785	0.42947066	0.48119150

最终，统计排名前 5 的热点问题，保存为附件热点问题表。并且统计相应热点问题对应的留言信息，保存为附件热点问题留言明细表。

2.3.3 对评价体系的建立和分析结果

在这一部分，我们需要从答复的相关性、完整性、可解释性等角度出发来设计一套评价方案，以此来评价相关部门的答复意见的质量。

首先本文对留言主题、答复意见分词并进行词频统计，选取前 100 个词绘制词云图，以便从直观上了解，词云图见图 8 和图 9：

图 8 留言主题词云图

图 9 答复意见词云图

从图 9 答复意见词云图中，可以发现，政府感谢各位网友的留言，并且针对反映的相关问题会进行一定的调查，接受群众的监督。

通俗来讲，答复的相关性即为答复与其对应留言的相关程度，若是相关程度很小，则说明此答复有效性很低；答复的完整性即为所给的答复是否全面详尽，而不是三言两语带过。从实际角度考虑，留言人很关心的一个问题是答复是否及时，所以此处加入及时性这一指标。目前这些指标都很抽象，所以需要对指标进行量化处理。

利用 R 对上述 3 个指标做量化处理，对相关性、完整性、及时性求加权和来作为质量评价指标，权数用熵权法来确定，得到 3 个指标的权重，具体结果见

表 4。

表 4 评价体系中指标权重

指标	相关性	完整性	及时性
权重	0.118295634	0.879873433	0.001830933

然后利用得到的权重对每一条答复意见求加权和，得到样本得分，得分越高则说明答复意见的质量越高。详细的样本得分见附件：第三问答复意见质量评价得分。

3.结论

本文根据已收集的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，对其进行自然语言处理和文本挖掘，运用了支持向量机、决策树、boosting、bagging、随机森林等方法。

对留言内容这一栏文本数据，通过特征提取，进行文本分类的训练，能够从文本中提取到有效信息进行留言内容的分类。这样更加便于“智慧政务”的构建，不需要人工对每一个留言内容进行分类，降低了人工成本。留言中反映的问题有很多是同一时间段内的同种问题，这些问题可以归为热点问题。实验中通过聚类，并且利用聚类之后的类留言总数、类反对数和类点赞数构建热度评价指标，并且热度较高的问题也进行了统计。另外，针对答复意见也构建了评价体系，能够对回复进行一个有效的评价。

在实际的分析操作中，我们得到的结果还有很多是效果不够好的，比如在对热点问题聚类的过程中，我们还没能将精度再提升一些；在对答复意见的评价体系中，我们的评价指标还是不够全面。而这些文本挖掘模型上的不足以及对原始数据的清洗，我们都还有很多需要深入探讨的地方。

4.参考文献

[1] 薛为民,陆玉昌.文本挖掘技术研究[J].北京联合大学学报:自然科学版,2005,19(4):59-63.

[2] 阳小兰,钱程,赵海廷.Web 文本预处理技术探析[J].电脑知识与技术,2010,06(29):8247-8249.

[3] 庞剑锋,卜东波.基于向量空间模型的文本自动分类系统的研究与实现[J].计算机应用研究,2001,18(9):23-26.

[4] Salton G, Yu C T. On the construction of effective vocabularies for information retrieval[C]// Meeting on Programming Languages and Information Retrieval. ACM, 1973:48-60.

[5] 周志华.机器学习[M].北京:清华大学出版社,2016:73-196.

[6] 樊东辉.基于文本聚类的特征选择算法研究[D].西北师范大学.

[7] 张一文,齐佳音,方滨兴,等.非常规突发事件网络舆情热度评价指标体系构建[J].情报杂志,2010,029(011):71-75,117.

[8] 周茜, 赵明生, 扈旻. 中文文本分类中的特征选择研究[J]. 中文信息学报, 2004, 18(3):18-24.

[9]Daverain.[EB/OL].(2019-05-21)[2020-05-06].https://blog.csdn.net/Daverain/article/details/80919418

[10] AI 专家.机器学习之十大经典算法（十） 随机森林算法.[EB/OL].(2018-6-10)[2020-05-06].https://blog.csdn.net/weixin_42039090/java/article/details/80640890