基于机器学习的
文本挖掘在“智慧政务”中的应用

目录

基于机器学习的文本挖掘在“智慧政务”中的应用

内容摘要

本文旨在利用基于机器学习来进行文本分类、文本聚类以及文本评价，为网上政务工作提供便利。

针对问题一，它是一个文本分类问题。本文文本分类的大致思路为在对文本进行完预处理之后，首先需要利用 TF-IDF 值把文本数据转变为词向量数据，接着基于 SVM 模型进行文本分类，最后依题目要求使用 F1 度量进行分类结果的评估。通过观察数据以及对题目的分析，题中获取的数据存在数据样本缺失或不平衡的问题。本文采用分层采样与欠采样来处理这个问题。另外，考虑到使用 SVM 模型进行文本多分类可能计算时间较长且效果不佳，本文把多分类模型转化为多个二分类模型，以提高分类的准确性，使用 F1 度量来进行这两种方法的比较。最后给出一个相对合理的模型评估，得到多分类拆分为二分类的 SVM 分类模型效果较好的结论。

针对问题二，首先利用 jieba 分词对留言主题进行预处理，而后通过 Simhash 算法对文本数据进行 0-1 向量化，在向量基础上使用海明距离进行两两对比检索，再基于 DBSCAN 密度聚类算法对其进行文本聚类，阈值设定为 20，划分为 1064 类，效果较好，考虑到语义的特殊性，为保证分类准确率，对提取的部分留言进行人工筛选，为了得到前 5 类热点问题，提取了排名前 40 的留言进行筛选，而后构建以点赞数、反对数、该类留言数量结合的热点评价指标，并提出前 5 类热点问题。

针对问题三，利用 jieba 分词将留言和答复意见进行分词，再利用 TF-IDF 频率提取的关键词，进而使用 Simhash 算法将文本向量化。在得到的文本向量和提取的关键词的基础上，我们给出了答复意见和留言的相关性、完整性、可解释性的量化度量并在题目给出的数据上得到了验证。

关 键 词：TF-IDF 权重　SVM 分类模型　多分类拆分 Simhash DBSCAN 聚类　余弦相似度

Application of Text Mining Based on machine Learning in "Intelligent government affairs"

Abstract

This paper aims to make use of machine learning-based text classification, text clustering and text evaluation to provide convenience for online government work.

For problem one, it is a text classification problem. The general idea of text classification in this paper is that after text preprocessing, text data should be converted into word vector data by using TF-IDF value, then text classification should be carried out based on SVM model, and finally classification results should be evaluated by using F1 measurement according to the requirements of the topic. By observing the data and analyzing the questions, the data obtained in the questions has the problem of missing or unbalanced data samples. In this paper, stratified sampling and undersampling are used to solve this problem. In addition, considering that the use of SVM model for text multi-classification may take a long time to calculate and the effect is not good, this paper transforms the multi-classification model into multiple binary classification models to improve the accuracy of classification, and USES F1 measurement to compare the two methods. Finally, a relatively reasonable model evaluation is given, and the results of SVM classification model with multiple classification into binary classification are better.

For question 2, the first message subject to make use of jieba participle preprocessing, and then through Simhash algorithm to 0-1 vectorization of text data, on the basis of vector using hamming distance pairwise comparison retrieval, then based on the density of DBSCAN clustering algorithm of text clustering, the threshold is set to 20, 1064 classes, divided into effect is good, considering the particularity of semantics, in order to ensure the classification accuracy, the extraction of part of the message to artificial selection, in order to get the first five kind of hotspot issues, to extract the top 40 message filtering, Then, a hot spot evaluation index combining thumb up number, opposition number and the number of such comments is constructed, and the first five hot spot problems are proposed.

For question 3, jieba word segmentation is used to segment comments and replies, and keywords extracted from TF-IDF frequency are used to extract the text by using Simhash algorithm. On the basis of the obtained text vector and the extracted keywords, we give a quantitative measure of the relevance, integrity and interpretability of the comments and comments and verify it on the data given by the topic.

.

Key words: TF-IDF weight SVM Multi-classification Split Simhash DBSCAN Cosine Similarity

一、问题重述

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 待解决的问题

1.2.1 群众留言分类

在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系（参考附件 1 提供的内容分类三级标签体系）对留言进行分类，以便后续将群众留言分派至相应的职能部门处理。目前，大部分电子政务系统还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。请根据附件 2 给出的数据，建立关于留言内容的一级标签分类模型。

通常使用 F-Score 对分类方法进行评价：

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

其中P_i为第 i 类的查准率，R_i为第 i 类的查全率。

1.2.2 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，给出排名前 5 的热点问题，并保存为文件“热点问题表.xls”。给出相应热点问题对应的留言信息，并保存为“热点问题留言明细表.xls”。

1.2.3 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。

二、文本数据预处理

进行文本挖掘毫无疑问必须要涉及到对文本数据进行预处理。文本挖掘的数据清洗及预处理大致包括：

1. 数据清洗：使用正则表达式去除文本中的特殊字符。
2. 文本分词：对文本的词语进行划分，必要时候需要标注文本的词性。
3. 文本去停词：过滤掉文本中无意义的词语（停词），比如：我，你，然而等。可借用停词词典。

在对政务留言进行文本挖掘的问题中，文本数据预处理贯穿全程。无论是解决哪个问题，都需要进行以上步骤。本文使用正则表达式去除标点符号与空格。又由于本文主要是对中文文本的挖掘，所以在第一题进行了对数字与英文字符的剔除。而在第二题的解决中，经过对文本数据例子的观察，英文字符与数字字符表示了地点，因此没有删除英文字符与数字字符。

在对中文文本分词上，本文利用了 jieba 分词。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。jieba 分词系统提供分词、词性标注、未登录词识别，支持用户自定义词典，关键词提取等功能。在第一题中，利用的是 jieba 分词中的混合模型，它结合了最大概率法与隐式马尔科夫模型的。在第二第三题中，本文运用的是 jieba 分词中的 Simhash 模型，从而得到 Simhash 值。另外，对于文本去停词，本文经过一系列对数据进行的统计，如计算词频等，形成了一个自定义的停词库，以更好地对文本无意义词语进行过滤。

三、问题一的分析与探究

问题一是一个文本分类问题。本文文本分类的大致思路为在对文本进行完预处理之后，首先需要利用 TF-IDF 值把文本数据转变为词向量数据，接着基于 SVM 模型进行文本分类，最后依题目要求使用 F1 度量进行分类结果的评估。

通过观察数据以及对题目的分析，可以发现政务留言数据中有 7 种一级标签的分类，且样本数量各不相同。但在附件 1 的数据中可以了解到，政务留言在一级标签下一共可分为 15 种类型。因此题中获取的数据存在数据样本缺失或不平衡的问题。本文采用分层采样与欠采样来处理这个问题。另外，考虑到使用 SVM 模型进行文本多分类可能计算时间较长且效果不佳，本文把多分类模型转化为多个二分类模型，以提高分类的准确性，最后也是使用 F1 度量来进行这两种方法的比较。

大致流程如下：

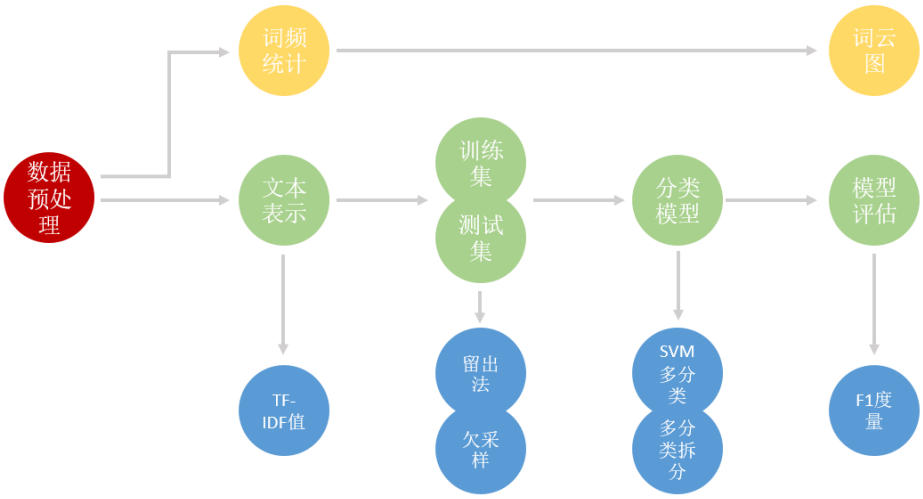

图 3.1 问题一流程图

3.1 数据观察

为了更好地对政务留言文本数据进行分类，本文先对已进行数据清洗的各类型文本数据进行词频统计以及词云图的制作，以大致了解每个类型的关键词及关键词提取的效果。

下面是七个类型的词云图。

图 3.2 城乡建设词云

图 3.3 环境保护词云图

图 3.4 交通运输词云图

图 3.5 教育文体词云图

图 3.6 劳动和社会保障词云图

图 3.7 商贸旅游词云图

图 3.8 卫生计生词云图

从 7 个词云图来看，城乡建设的关键词是：小区、业主、房屋等，环境保护的关键词是：污染、环保局等，交通运输的关键词是：出租车、司机、快递等，教育文体的关键词是：学校、教师、学生等，劳动与社会保障的关键词是：工作、职工、社保等，商贸旅游的关键词是公司、传销、景区等，卫生计生的关键词是医院、医生、生育等。虽然各组的关键词有交叉，但大体上来看还是比较能体现出各类型的特点，也就是说各类型的关键词是有区分度的。

3.2 文本表示

本文 3.1 的文本数据词云图可以看出，用词频来提取出来的关键词可以比较好地体现各分类的特点。因此我们可以基于词频来提取文本关键词，并且得出词向量，也就是用词向量进行文本表示。但是若某个词在某个文本类型的某个文档中出现频繁，那么词频不能说明这个词对此文本类型是重要的。因此本文使用另一个统计量 TF-IDF 值来进行文本表示。

3.2.1 TF-IDF 算法

TF-IDF（词频-逆向文件频率）是一种用于信息检索与文本挖掘的常用加权技术。TF-IDF 是一种统计方法，用以评估一字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。字词的重要性随着它在文件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。TF-IDF 的主要思想是：如果某个单词在一篇文章中出现的频率 TF 高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

词频（TF）表示词条（关键字）在文本中出现的频率。这个数字通常会被归一化（一般是词频除以文章总词数），以防止它偏向长的文件。

$$\mathrm{TF} = \frac{某一文本类型词条出现次数}{该类词条总次数}$$

逆向文件频率 （IDF）：某一特定词语的 IDF，可以由总文件数目除以包含该词语的文件的数目，再将得到的商取对数得到。如果包含词条 t 的文档越少, IDF 越大，则说明词条具有很好的类别区分能力。

$$\mathrm{IDF} = \log\left(\frac{文本总数}{包含该词条的文本数 + 1}\right)$$

最后得到 TF-IDF

$$\mathrm{TF} - \mathrm{IDF} = \mathrm{TF} * \mathrm{IDF}$$

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，可以产生出高权重的 TF-IDF。也就是说 TF-IDF 值越大，词条越关键，并且 TF-IDF 倾向于过滤掉常见的词语，保留重要的词语。

3.2.2 TF-IDF 矩阵

对文本数据进行处理，每个文档种取出最重要的五个关键词，得到各个词的 TF-IDF 值，形成 TF-IDF 矩阵，其中行向量是词条向量，列向量是文档向量。

表 3.1 文本数据形成的 TF-IDF 矩阵数据

列数	80559

行数	9210
0 元素	78653678

由上述数据可以看出，这个 TF-IDF 矩阵是一个高维矩阵，这对模型的构建很不利。但是从矩阵 0 元素的个数可以看出这是一个稀疏矩阵，这对模型构建来说是一个有利的消息。从另一方面来看，这个稀疏矩阵中必有大量词条对整体的影响不大，因此本文利用 TF-IDF 权重筛选词条，得到行数为 9210，列数为 8590 的 TF-IDF 矩阵。

3.3 利用 SVM 模型进行文本分类

利用 TF-IDF 权重形成词向量来进行文本表示之后，便可开始进行分类模型的构建，本文利用 SVM 模型进行文本分类。

3.3.1 SVM 分类模型

SVM（支持向量机，Support Vector Machine）的主要思想是：建立一个最优决策超平面，使得该平面两侧距离该平面最近的两类样本之间的距离最大化，从而对分类问题提供良好的泛化能力。对于一个多维的样本集，系统随机产生一个超平面并不断移动，对样本进行分类，直到训练样本中属于不同类别的样本点正好位于该超平面的两侧，满足该条件的超平面可能有很多个，SVM 正式在保证分类精度的同时，寻找到这样一个超平面，使得超平面两侧的空白区域最大化，从而实现对线性可分样本的最优分类。

SVM 是一种有监督的学习方法。支持向量机中的支持向量（Support Vector）是指训练样本集中的某些训练点，这些点最靠近分类决策面，是最难分类的数据点。SVM 中最优分类标准就是这些点距离分类超平面的距离达到最大值；“机”（Machine）是机器学习领域对一些算法的统称，常把算法看做一个机器，或者学习函数。

SVM 分类算法分为线性与非线性两种。二维的线性 SVM 就是找到一条直线，使得该直线两侧距离该直线最近的两类样本之间的距离最大化。对于非线性的情况，一种方法是使用一条曲线去完美分割样品集，如下图所示：

图 3.9 二维非线性 SVM

从二维空间扩展到多维，用一个超平面对样本进行划分，这种情况下，相当于增加了不同样本间的区分度和区分条件。在这个过程中，核函数发挥了至关重要的作用，核函数的作用就是在保证不增加算法复杂度的情况下将完全不可分问题转化为可分或达到近似可分的状态。线性不可分映射到高维空间，可能导致很高的维度，特殊情况下可能达到无穷多维，这种情况下会导致计算复杂，伴随产生惊人的计算量。但是在 SVM 中，核函数的存在，使得运算仍然是在低维空间进行的，避免了在高维空间中复杂运算的时间消耗。

根据算法思想写出算法：

在样本空间内，划分超平面可通过下面线性方程描述：

$$\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{x}+b=0$$

样本空间中任意点$\boldsymbol{x}$到超平面的距离可写为

$$\mathrm{r}=\frac{|\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{x}+b|}{\|\boldsymbol{\omega}\|}$$

分类准则为

$$\begin{cases}\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{x_i}+b\geq 1, y_i=1\\ \boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{x_i}+b\leq -1, y_i=-1\end{cases}$$

距离超平面最近的这几个训练样本点使上述式子的等号成立，它们被称为“支持向量”，两个异类支持向量到超平面的距离之和为

$$\gamma=\frac{2}{\|\boldsymbol{\omega}\|}$$

写出目标函数

$$\max_{\boldsymbol{\omega},b}\frac{2}{\|\boldsymbol{\omega}\|}$$

$$\mathrm{s.t.}\, y_i(\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{x_i}+b)\geq 1$$

这就是支持向量机的基本型。

SVM 的优点：不需要很多样本，不需要有很多样本并不意味着训练样本的绝对量很少，而是说相对于其他训练分类算法比起来，同样的问题复杂度下，SVM 需求的样本相对是较少的。并且由于 SVM 引入了核函数，所以对于高维的样本，SVM 也能轻松应对。构风险最小。这种风险是指分类器对问题真实模型的逼近与问题真实解之间的累积误差。非线性，是指 SVM 擅长应付样本数据线性不可分的情况，主要通过松弛变量（也叫惩罚变量）和核函数技术来实现。

3.3.2 多分类模型拆分

使用上述模型对 3.2.2 得出的 TF-IDF 权重矩阵进行文本分类。本文先进行了一个大致的模型评估，从而观察使用 SVM 分类算法进行多分类与二分类的区别。

本文从各类型文本中分别随机选取 490 个留言详情文本，组成一个样本平衡的，一共有 3430 个样本点的训练集，而剩余的 5780 个样本则作为测试集。可见这样的数据相差较大，且测试集比训练集多的模型评估里，得到的结果一般不尽如人意，但是本文用这种取样方法来说明 SVM 分类模型对多分类与二分类模型效果的差异。

首先使用 SVM 分类模型进行七种模型的分类，然后把属于城乡建设的样本设为正类，其他类型的文本作为反类，以进行二分类。进行十次模拟，最后的结果使用较为简单的性能度量准确率的均值来评估。以下使模型评估结果：

表 3.2 多分类与二分类准确率

	Accuracy
二分类	0.896
多分类	0.521

由上述结果可知，在同样的训练条件下，SVM 分类模型在进行二分类时比在进行多分类时的效果要好得多。即使在这种训练集较少的情况下，二分类的分类准确率也几乎能达到 90%。当然上述结果也不能说明 SVM 分类模型在进行多分类时的准确率只有一半这么低，在这个模拟里，造成准确率低的很大一部分原因是训练集样本量的不足。

由以上模型评估可知，把多分类模型拆分为二分类模型也许能提高 SVM 分类模型进行文本分类的性能。本文采用“一对其余”（OvR）的拆分策略，即每次将一个类的样本作为正例，其他所有类的样本作为反例，由此训练七个分类器，选取分类器预测为正类的对应的类别标记作为最终的分类结果。下面是本文把多分类模型拆分为若干个二分类模型的示意图：

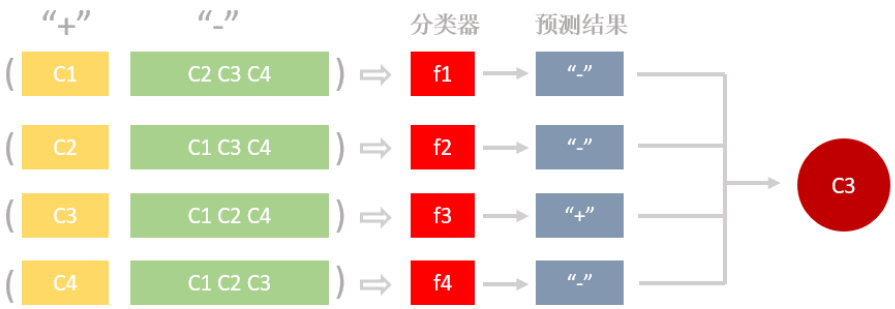

图 3.10 OvR 示意图

接下来就利用多分类拆分为二分类的 SVM 分类模型进行七种文本类型的分类，并使用 F1 度量来评估模型性能，依旧是使用上述划分训练集与测试集的方法，以更好地观察两者差异。下面是两种分类模型的结果

表 3.3 多分类的 F1 度量

真值 / Pre	城乡建设	环境保护	交通运输	商贸旅游	卫生计生	教育文体	劳动和社会保障
城乡建设	1351	16	43	561	157	336	234
环境保护	128	430	7	31	14	5	11
交通运输	21	1	71	28	1	1	1
商贸旅游	4	0	0	92	0	0	0
卫生计生	0	0	0	3	180	1	29
教育文体	8	1	0	5	0	685	8
劳动和社会保障	7	0	2	5	35	71	1196

各类 F1	0.64	0.80	0.57	0.22	0.60	0.76	0.86
模型 F1	0.63						

表 3.4 二分类的 F1 度量

真值 / Pre	城乡建设	环境保护	交通运输	商贸旅游	卫生计生	教育文体	劳动和社会保障
城乡建设	1346	33	25	216	44	155	161
环境保护	48	431	4	12	1	2	3
交通运输	51	1	93	29	1	3	8
商贸旅游	34	0	1	453	10	10	10
卫生计生	7	0	0	4	326	5	96
教育文体	22	1	0	10	0	908	37
劳动和社会保障	11	0	0	1	5	16	1164
各类 F1	0.77	0.89	0.60	0.73	00.80	0.87	0.87
模型 F1	0.79						

从上述以表格形式呈现的结果，可以看出，把多分类拆分为二分类之后，F1 度量提高了 0.16，这是一个很显著的提高。F1 度量是一个同时模型衡量查准率与查全率的数值。也就是说，结合模型查全率与查准率二者来看，把多分类拆分成二分类的模型优于直接进行多分类的模型。但是从两个结果看来，F1 都没有达到 0.8，模型评估的结果似乎不佳，但实际上还是训练集与测试集划分的问题。本文的下一小节会重新进行训练集与测试集的划分，以更准确地评价模型效果。

3.4 模型评估

接下来就利用多分类拆分为二分类的 SVM 分类模型进行文本数据的分类，且使用 F1 度量来评估模型的性能。

首先从选取样本以及划分训练集与测试集的问题开始说起。从上一小节可以看出，模型评估时样本点的选取以及训练集和测试集划分与模型评估的结果息息相关，如若这两步做不好，就会极大地影响模型效果。这一章的开头已经提到过，题中获取的数

据存在数据样本缺失或不平衡的问题，那么本文采用分层采样与欠采样来处理这个问题。

第一步是解决类别样本不平衡的问题。类别不平衡指的是分类任务中不同类别的训练样例数目差别很大。一般来说，若训练集的类别样本量之间差异过大，则会对学习过程造成困扰，这样的学习器往往是没有价值的。本题数据中除去数据里缺失的类别，样本量最多的类别是城乡建设类型，有 2009 个样本，最少的是交通运输类型，有 613 个样本。由此看来本题数据类别不平衡的问题较为严重，因此本文采用欠采样的方法。欠采样即去除一些样本量较多的类别的样本，使其样本量与样本量较少的类别的样本量相当。另外一种是过采样方法，过采样即在样本量较少的类别中增加一些样本，使其样本量与样本量较多的类别的样本量相当。考虑到本题的样本政务留言及其分类较难获取，因此采用欠采样方法。但可惜的是本文并没有解决类别样本缺失的问题，以及在欠采样的过程中有可能丢失一些重要信息。希望以后能继续完善。

第二步是解决划分训练集与测试集的问题。本文使用留出法，即将数据集划分为两个互斥的集合，一个集合作为训练样本，一个集合作为测试样本 T。为了在划分过程中，保留类别比例，因此采用分层采样的方法。这样能尽可能保持数据分布一致性，避免因数据划分过程引入额外的偏差而对最终结果产生影响。但可惜的是本文并没有解决类别样本缺失的问题，以及采取欠采样可能会丢失一部分重要信息。这两点希望之后能够进行改进。

因此本节从各类型文本中先分别随机选取 613 个留言详情文本，组成一个样本平衡的，一共有 4291 个样本的数据集。再从各类型文本中先分别随机选取 490 个文本，组成一个一共有 3430 个样本的训练集，而剩余的 861 个样本则作为测试集。可见现在这个测试集与训练集是样本平衡，且保留了类别比例，训练集的样本数亦不会过少。接下来就使用多分类拆分为二分类的 SVM 分类模型进行文本分类。

表 3.5 模型评估

真值 / Pre	城乡建设	环境保护	交通运输	商贸旅游	卫生计生	教育文体	劳动和社会保障
城乡建设	123	2	15	18	7	14	10
环境保护	0	120	4	5	1	0	1
交通运输	0	0	103	3	0	0	0

商贸旅游	0	0	0	94	5	3	1
卫生计生	0	1	0	1	109	0	12
教育文体	0	0	1	2	0	104	2
劳动和社会保障	0	0	0	0	1	2	97
各类 F1	0.79	0.94	0.90	0.83	0.89	0.90	0.87
模型 F1	0.87						

由上表可以看出，相比起之前的数据选取，模型分类的效果大大提升，F1 度量提高了 0.8，F1 度量达到 0.87，接近 0.9。并且各类别的 F1 都有了明显提高，也就是说这一模型评估的结果比较接近模型真实的效果。观察数据矩阵，可以直观地看到此矩阵类似于一个对角矩阵，即模型地准确率较高。因此无论从准确率、查全率以及查准率来看，本文运用的多分类拆分为多个二分类的 SVM 分类模型的效果不错。

四、问题二的分析与探究

问题二是一个文本聚类问题。本文文本聚类的大致思路为，在对文本进行完预处理之后，首先利用 Simhash 算法对文本数据进行向量化，得到文本内容对应的 Simhash 签名，而后在 Simhash 签名值的基础上使用海明距离进行两两对比检索，接着基于 DBSCAN 密度聚类算法进行文本聚类，构造热度评价指标进行对结果进行评价，最后提取聚类主题的关键词进行总结。

通过观察数据以及对题目的分析，发现可以从 “留言主题”层面出发，提取文本数据进行下一步的分析。提取过程中发现，其中一组留言数据中缺失“留言主题”，我们考虑提取它的“留言详情”进行替代。另外，对于聚类算法的选择，首先尝试了常规聚类的 K-means 聚类以及层次聚类，但因为文本量过大、计算时间较长且难以确定聚类数量，或是无法从层次上得到更加直观的解释，基于向量化后的文本数据形式，最终采取 DBSCAN 密度聚类进行聚类分析，聚类效果较好。考虑到计算机对语义类的理解不一定完全准确，而后进行人工筛选以增强结果的可靠性。

大致流程如下：

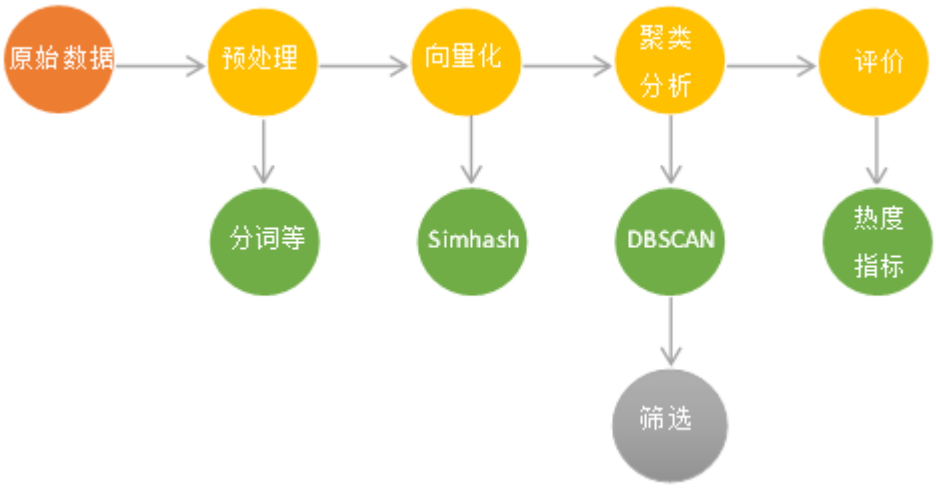

图 4.1 问题二流程图

4.1 文本预处理

在对留言主题进行分析之前，先要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 3 中，以中文文本的方式给出了数据。为了便于转换，先要对这些留言主题进行中文分词。这里采用 jieba 进行分词。

4.2 文本向量化

由于计算机不能够直接处理文本信息，我们需要对文本进行处理，将文本表示成为计算机能够直接处理的形式，即文本数字化。本文使用 Simhash 算法-传统词向量对关键词进行向量化，最后得到一个短文本的向量，再使用得到的文本向量进行后续的计算。

4.2.1 Simhash 算法

Simhash 算法，由 3 位 Google 的工程师 G. Manku、A. Jain 和 A. D. Sarma 于 2007 年提出，Simhash 算法是一种高效的高维数对象降维的指纹方法，Simhash 算法将高维数对象映射为位数较小且固定的二进制码，抽屉原理可以快速检索出所有与给定二进制码相差为 k 的二进制码。目前，Simhash 广泛用于很多领域，例如在大规模软件系统中软件代码的克隆检测、互联网中的信息检索、Web 图检测等。

传统的哈希算法通过计算将输入数据映射成特定长度的哈希值输出，输入数据的差异越大，映射出的签名值差异也越大。传统的哈希算法，原理上相当于伪随机数产生算法，对 K 比特差距的输入数据都会输出完全不同的哈希值，因此无法检测相似文档。需要对原有哈希算法进行改进，使得相似文档可以输出相似的哈希值。Simhash 算法是一种改进的哈希算法，可以高效的识别出输入数据是否相似。Simhash 算法有两个步骤：

第一步：hash 值的计算

首先，每一篇文档内容对应一个初始值为 0 长度为 f 的签名 S，一个初始值为 0 的 f 维向量 V；其次，通过分词库对文档内容进行分词，过滤掉一些语气词、助词，并去掉干扰符号后将文档内容转换为一组特征词，特征词的权重为该特征词在文档中出现的次数；第三步，将所有特征词使用相同的哈希函数映射为长度为 f 的签名 h，遍历 h 的每一位，若 h 的第 i 位为 1(i 介于 1 到 f 之间)，V 的第 i 位加上该特征词的权重，否则减去；最后，遍历 V，如果 V 的第 i 位大于 0，签名 S 的第 i 位设为 1，否则设为 0，最终生成的签名 S 就是文档内容对应的 Simhash 签名。hash 值计算示意图如下：

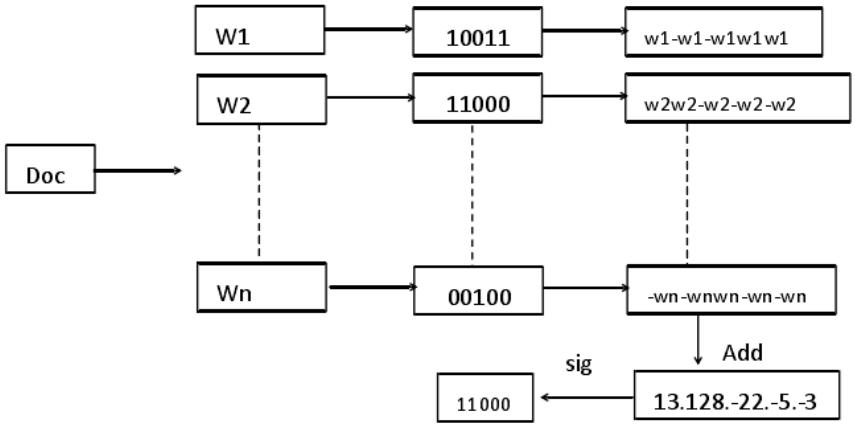

图 4.2　Simhash 算法 hash 值计算示意图

表示形式如表4.1所示，部分结果示例如图4.3，可表现其 Simhash 值及该主题的关键词。

表 4.1 文本数据形成的 Simhash 标签示例

	主题顺序编号
Simhash	11438535477625838033
keyword	一米阳光　艺术摄影　区　婚纱

此步的 Simhash 表示的权重向量仍为十进制，通过 tobin 函数将其转化为 64 位的二进制编码，从而实现 0-1 向量化，如表 3.2 所示，其中列数表示 64 位编码，行数是文本数据数量。

表 4.2 二进制表示的 Simhash 值

列数	64
行数	4326

从上表可知，Simhash 值的矩阵维数比 TF-IDF 权重矩阵的维数要小的多，且此矩阵只有 0 和 1 两种元素，因此为文本相似性研究提供了极大便利。

4.2.2 Simhash 值-海明距离

第二步：对 Simhash 签名值使用海明距离。二进制码的相似比较可使用海明距离，海明距离可比较出两个二进制码相差的位数。将所得到的 Simhash 签名值两两进行海明距离计算。

4.3 利用 DBSCAN 密度聚类进行文本聚类

4.3.1 文本相似度计算

相似度是用来衡量文本间相似程度的一个标准。在文本聚类中，需要研究文本个体将的差异大小，也就是需要对文本信息进行相似度计算，将根据相似特性的信息进行归类。目前相似度计算方法分为距离度量和相似度度量。本文采用的是利用Simhash二进制编码化后计算得到的海明距离，进而度量留言主题间差异。

假设 $H(\cdot) = (h_1, h_2, \ldots, h_j)$ 中包含了j 个独立存在的哈希函数，由这些函数共同组合而哈希函数向量，$H(\cdot)$ 分别把 q 和 o 映射成长度为 j 的二进制编码 $H(q)$ 和 $H(o)$，而第 k 比特的二进制编码对应为 $h_k(\cdot)$，则 q 与 o 的海明距离表示为

$$D\big(H(\boldsymbol{q}), H(\boldsymbol{o})\big) = \sum_{k=1}^{j} ||h_k(\boldsymbol{q}) - h_k(\boldsymbol{o})||$$

表 4.3 数据 1 与其他数据间的海明距离(部分)

	C
C	0
C1	19
C2	30
…	…
C4325	31

表 4.3 展示了第一个数据(部分)，即第一个留言，与其他留言间的海明距离，其中 C 表示第一个留言的标号，C1 表示第二个留言的编号，以此类推。数字部分即两两成分间的海明距离。

4.3.2 聚类算法的选择

所谓文本聚类就是将无类别标记的文本信息根据不同的特征(如距离准则)，将有着各自特征的文本进行分类，使用相似度计算将具有相同属性或者相似属性的文本聚类在一起。即聚类后同一类的数据尽可能聚集到一起，不同数据尽量分离。这样就可以通过文本聚类的方法对留言进行分类。本文在摸索过程中分别尝试了三种算法，即K-means聚类、层次聚类以及 DBSCAN 聚类。

(1) K-means聚类

K-means 算法是很典型的基于划分的聚类算法，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似性就越大。

在一个数据集中，我们假设其中包含 n 个数据对象，划分聚类方法可以把这些数据划分成 K 个部分，每一个部分就表示为一个簇，即，划分聚类方法将这个数据

集中的数据对象划分成为 K 个簇，而对于这 K 个簇则满足以下条件：

(i) 在 K 个簇中，每个簇中都至少要包含一个数据对象。

(ii) 每一个数据对象只能够唯一属于一个簇。

根据划分聚类算法的运算规则，首先要定义一个 K 值，然后根据这个 K 值将所有的数据对象划分为初始的 K 个簇，通过反复的运行划分聚类算法进行迭代处理，以使得每一次的对簇的划分结果都比之前的划分情况要更好一些。

将进行海明距离计算后的数据，尝试用 K-means聚类进行分析，发现存在两个问题：(i) 在寻找算法预设 K 的过程中，可能由于文本数据的特殊性，它无法确定大致类别，即数据量大、所分类别过多，导致无法找到合适的 K 值进行下一步的分析；(ii) 预设 K 值后，由于无法判断事先设定的 K 值是否合适，而无法判断模型结果的有效性。

(2) 层次聚类

在给定 n 个对象的数据集后，可用层次方法（Hierarchical Methods）对数据集进行层次分解，直到满足一定的收敛条件为止。层次聚类算法以分解形式的不同可以分成凝聚层次聚类和分裂层次聚类两种。层次聚类过程中，数据对象最初都是在一个簇中。根据相应的准则逐步对该簇进行分裂。并将簇的分裂过程逐次的依次进行下去直到一个簇中仅有一个数据对象为止。 在簇的合并过程中，也是需要遵循一定的规则进行，使得数据对象之间的距离较近的数据对象尽可能的合并到一个簇中，并逐次的按照相关的规则对簇不断的进行合并，当达到所有的数据对象都被合并在一个簇中为止。

分别尝试了层次聚类下不同的规则，如分裂层次聚类、最短距离法、水平聚类树等等。但得到的结果都是类似的：由于种类过多无法进行大致的划分判断。

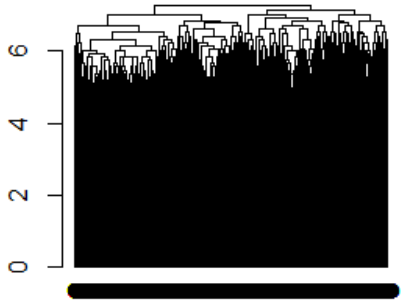

图 4.3 分裂层次聚类

层次方法最大的不足在于合并或者分裂点的选择比较困难，对于局部来说，好的合并或者分裂点的选择往往并不能保证会得到高质量的全局的聚类结果并且一个合并或者分裂过程完成后是不能再返回重新修改的。

基于以上探索后发现，DBSCAN-密度聚类可以较好地对本文中的数据进行聚类分析。

4.3.3 DBSCAN 聚类

DBSCAN (Density Based Spatial Clustering of Application with Noise) 是一个基于密度的聚类算法。该算法将具有足够高密度的区域划分为簇，并可以在带有“噪音”的空间数据库中发现任意形状的聚类。它定义簇为密度相连的点的最大集合。

作为一种基于密度的聚类算法，DBSCAN 可从针对任意形巧的数据集进行聚类。该算法需要输入一对全局密度参数如 *MinPt* 和 *Eps*，并且支持用户自行选择合适的参数值。其中，参数 *MinPt* 是指某对象一定邻域范围内数据点的个数，而 *Eps* 则为邻域范围的半径。基于这对全局唯一的密度参数，DBSCAN 可检测出数据集中的类与噪音点。一般情况下，在一个类中，类边界点的密度要比类内点的密度小；并且噪音点的密度要比类内点的密度小。基于这个事实，DBSCAN 算法可利用密度参数识别出数据集中所有较高密度的类与低密度的噪音点。

DBSCAN 算法的密度定义是，对于每一个点在给定的邻域半径 *Eps* 内至少包含 *MinPt* 个点，即在该邻域内的密度必须超过一定的阈值。一般情况下，一个类中存在两种类型的点,在类内的点称为核心点，在类的边界处的点称为边界点。在 DBSCAN 中，如果一个数据点在其 *Eps* 邻域内所包含的点个数大于阈值 *MinPt*，即 $|N_{Eps}(p)| \geq MinPt$，则称其为核心点。如果一个数据点的 *Eps*－邻域范围内点的个数小于阈值 *MinPt*，但其邻域集合中存在核也点，则称其为边界点。

在聚类的过程中，DBSCAN 从任一点 p 开始并遍历所有从 p 关于参数 *Eps*、*MinPt* 密度可达的点。如果是一个核心点，遍历的过程会生成一个类；如果 p 是一个边界点，即没有点是从 p 密度可达的，DBSCAN 会继续检索数据集的下一个点。DBSCAN 算法的实现伪代码如表 4.4 所示：

表 4.4　DBSCAN 算法

输入：　样本集 D=$\{x_1, x_2,\ldots, x_m\}$;

领域参数(ϵ,MinPts)

过程：

1. 初始化核心对象集合：$\Omega = \emptyset$
2. **for** j=1,2,…,m **do**
3. 确定样本 x_j 的ϵ-领域 $N_\epsilon(x_j)$
4. **if** $|N_\epsilon(x_j)| \geq MinPts$ **then**
5. 将样本x_j 加入核心对象集合：$\Omega = \Omega \cup \{x_j\}$
6. **end if**
7. end for
8. 初始化聚类簇数：k = 0
9. 初始化未访问样本集合：$\Gamma = D$
10. **while** $\Omega \neq \emptyset$ **do**
11. 记录当前未访问样本集合：$\Gamma_{old} = \Gamma$
12. 随机选取一个核心对象 $o \in \Omega$，初始化队列 $Q = < o >$；$\Gamma = \Gamma \setminus \{o\}$
13. **while** $Q \neq \emptyset$ **do**
14. 取出队列 Q 中的首个样本 q
15. **if** $|N_\epsilon(q)| \geq MinPts$ **then**
16. 令$\Delta = N_\epsilon(q) \cap \Gamma$
17. 将Δ中的样本加入队列 Q
18. $\Gamma = \Gamma \setminus \Delta$
19. **end if**
20. **end while**
21. k = k + 1, 生成聚类簇 $C_k = \Gamma_{old} \setminus \Gamma$
22. $\Omega = \Omega \setminus C_k$
23. **end while**

输出：　簇划分 C=$\{C_1, C_2,\ldots, C_k\}$;

DBSCAN 聚类在本题中的应用的优势在于：第一是不用提前确定最优聚类簇数，这对于像本题这种聚类的类别太多，又或者难以确定最优聚类簇数的问题来说非常有利。第二是它是从一个个核心点开始聚类，对聚类类别多，且每类中元素较少，还有许多噪音点的问题比较有效。第三是本题已确定各点之间的距离，因此通过观察数据比较容易获得较合理的参数 *Eps*、*MinPt*。

4.3.4 DBSCAN 聚类结果

观察两两间海明距离表及多次测试发现，设定阈值 20 时，聚类效果较好，并在第一次聚类的基础上，对内部再次进行聚类，以下设定二次聚类阈值 15，最终得到的聚类结果，即各种类的留言主题，聚类效果较好。

表 4.5 部分 DBSCAN 聚类结果

留言编号	留言主题
189093	A3 区西湖街道茶场村五组什么时候能启动征地拆迁
189739	请问 A3 区西湖街道茶场村五组是如何规划的
197669	A3 区西湖街道茶场村五组何时才能启动拆迁
214447	A3 区西湖街道茶场村五组是如何规划的…
215709	A3 区西湖街道茶场村五组何时启动拆迁
228423	A3 区西湖街道茶场村五组不属于拆迁部分的村民该何去何从
240551	A3 区西湖街道茶场村五组什么时候能拆迁
…	…

经过 DBSCAN 聚类后，将留言分成 1064 类，部分如表 4.5 所示，聚类效果较好，但由于文本语义的不确定性，仍存在少许误差，为保证分类准确率，对提取的部分留言进行人工筛选，为了得到前 5 类热点问题，提取了排名前 40 的留言进行筛选，而后构建热点评价指标。

4.4 热点评价指标

提取前 40 类留言，进行热点评价。由于每条留言均设有点赞数以及反对数，故构建以下热点评价指标：

热点评价指标 = 点赞数+反对数+该类留言的数量

从而得到前 5 类热点留言，接着用 tag 提取该类留言主题的关键词，进行总结。

热点问题表及热点问题明细表如下：

表 4.6 热点问题表

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	154	2019/3/26 to 2019/3/26	A6 区月亮岛路	沿线架设高压电线杆
2	2	62	2019/7/18 to 2019/7/18	伊景园滨河苑开发商	捆绑销售车位
3	3	50	2019/1/10 to 2019/1/10	A6 市	人才落户新政公积金问题
4	4	24	2019/7/8 to 2019/7/8	A7 县诺亚山林小区	在门口设立医院
5	5	20	2019/07/21 to 2019/07/21	A5 区劳动东路魅力之城小区	餐馆小摊油烟扰民

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	254865	A0009901	关于A6区月亮岛路沿线架设110kv高压电线杆	2019/4/3 1	坚决要求A市润和又一城、三润城	0	5
1	262052	A0007242	关于A6区月亮岛路沿线架设110kv高压线杆的	2019/3/26	联名信——坚决要求A市润和又一	0	78
1	272089	A0006160	关于A6区月亮岛路110kv高压线的建议	2019/4/9 1	尊敬的胡书记，您好！根据区政府	2	55
1	268250	A0007242	关于A6区月亮岛路沿线架设110KV高压电线杆	2019/3/26	联名信——坚决要求A市润和又一	0	10

图 4.4 热点问题明细表（部分）

图 4.4 为截取部分热点问题明细表，在此列出，由于文件内容较多，该热点问题明细表请见附件。从结果可以看，相关留言条数多的不一定热度最高。本文考虑点赞

数与反对数也是有一定道理的，点赞数与反对数在一定程度上反映了该条留言的被关注度，并且反对数也是被关注度的一个正向衡量。

五、问题三分析与探究

5.1 问题三流程图

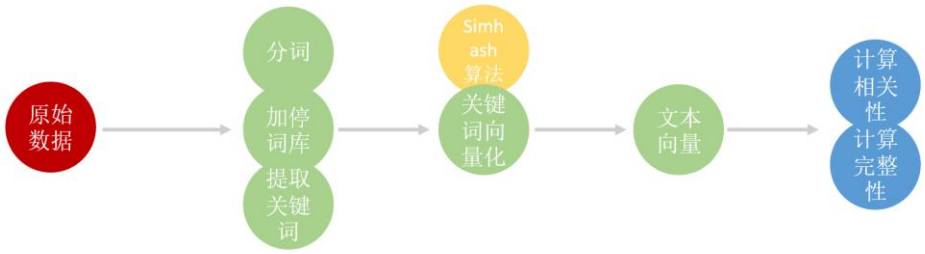

图 5.1 问题三流程图

5.2 研究方法

针对问题三，首先使用 R 语言中的 jiebar 对文本进行分词，再用 TF-IDF 权重提取 5 个关键词，对于提取出来的关键词再使用 Simhash 算法-传统词向量对关键词进行向量化，最后得到一个短文本的向量，再使用得到的文本向量进行后续的计算。

5.2.1 TF-IDF 权重

TF-IDF(Term Frequency-Inverse Document Frequency)是一种用于信息检索与数据挖掘的常用加权技术。前文已有提及，此处不再赘述。

5.2.2 Simhash 算法

Simhash 算法主要的工作是将文本进行降维，生成一个 Simhash 值，也就是论文中所提及的“指纹”，通过对不同文本的 Simhash 值进而比较海明距离，从而判断两个文本的相似度。对于文本去重这个问题，常见的解决办法有余弦算法、欧氏距离、Jaccard 相似度、最长公共子串等方法。

Simhash 的计算过程：分词、Hash、加权、合并、降维。

5.3 度量指标

5.3.1 相关性

在这里，我们使用留言文本向量和对应答复意见文本向量的余弦相似度来度量答复意见与网友留言内容的相关性。余弦相似性通过测量两个向量的夹角的余弦值来度量它们之间的相似性。0 度角的余弦值是 1，而其他任何角度的余弦值都不大于 1，并且其最小值是-1.从而两个向量之间的角度的余弦值确定两个向量是否大致指向相同方向。当余弦值为 1 时，说明两个向量的指向是相同的，-1 时向量指向完全相反，由于我们得到的向量是二进制，因此我们得到的余弦值都是在[0,1]中，则其值越接近 1，表明两个文本越相关。

5.3.2 完整性

对于完整性，我们使用答复意见对网友留言的覆盖来度量，若留言详情的关键词在答复意见中都有相似的描述，则表明答复意见满足完整性，若有一个关键词没有，则不满足完整性，也即：答复意见覆盖留言满足完整性，否则不满足。

5.3.3 可解释性

对于模型的可解释性，我们采用留言的关键词的词向量与答复意见关键词的词向量一一做相关性分析，若留言的关键词在答复意见中都能找到相关性比较强的关键词，则表明具有可解释性。

5.4 数据预处理

5.4.1 分词

首先将附件 4 导入 R 语言，导入分词包 jiebar，并设置分词器，这里采用“mix”模型并基于隐马尔可夫链分词，先对文本进行预处理，去除字母数字和标点，下面是分词的部分结果：

表 5.1 部分留言详情分词

文本编号	分词结果
1	年　月　以来　位于　市区　桂花　街道…
2	潇楚　南路　从年　开始　修到　现在　都…
3	地处　省会　民营　幼儿园　众多　小孩　是　祖国…
4	尊敬　的　书记　您好　我　研究生　毕业　后…
5	建议　将　白竹坡　路口　更名　为　马岭坡　小学
6	欢迎　领导　来市　泥泞不堪　的　小…
…	…

表 5.2 部分回复意见分词

文本编号	分词结果
1	现将　网友　在　平台　问政　西地省　栏目…
2	网友　您好　针对　您　反映　潇楚　南路…
3	市民　同志　你好　您　反应　的　请　加快…
4	网友　您好　您　在　平台　问政　西地省　上…
…	…

5.4.2 停用词

通过上述分词结果可以看出，与文本中心意思无关词语占了分词结果的多数，为了过滤掉这些词语，本文建立停词库，使用自建停词库再重新设置分词器，这里采用关键词模型，用 TF-IDF 权重提取 5 个关键词。以下是加入自建停词库以后分词的部分结果：

表 5.3 部分留言详情关键词

文本编号	关键词
1	物业公司　小区　投票　业主　业委会
2	潇楚　南路　挖　车　方便
3	幼儿园　教师　民营　工作　没交
4	公寓　研究生　毕业 买套　市
5	马岭坡　小学　原马坡　更名　白竹坡
6	泥巴　泥泞不堪　车　破　小镇
…	…

表 5.4 部分回复意见关键词

文本编号	关键词
1	业主大会　业委会　业主　停车 区
2	施工　坪塘　排水　换填　土方
3	民办 幼儿园 待遇 教师　学前教育
4	市　购房 房屋交易　含　补贴
5	均　马岭坡　来信　小学　公交站点
6	街道 含浦　学士　中　区
…	…

从上面例子看来，去停词与提取关键词前后差距较大。进行处理之后得出的关键词明显对文本具有一定的解释意义，并且能初步看出所描述的事件，比如：文本 1

讲的业主与小区业委会的之间的问题。把留言详情与回复意见进行对比，可以看出经过处理后两者的关键词有一定的关联性，这说明本文的预处理效果较好。

5.5 基于 Simhash 算法的度量指标的计算

首先利用 Simhash 算法将提取的关键词进行向量化得到加权 Hash 值，从而得到一个文本的向量，再将得到的权重向量全部 0-1 化，从而得到 64 位的二进制向量。以下是计算的部分结果：

```
      simhash                  keyword
 [1,] "11512412711503123959" Character,5
 [2,] "56315819903903577521"  Character,5
 [3,] "35849460213871521 95"  Character,5
 [4,] "78622017687741672 44"  Character,5
 [5,] "98362601232024365 23"  Character,5
 [6,] "17475821680080437169" Character,5
 [7,] "10114535444482956378" Character,5
 [8,] "78728673417826709 2"   Character,5
 [9,] "17080648310899769945" Character,5
[10,] "88449250930075229 57"  Character,5
[11,] "14815370207564758 50"  Character,5
[12,] "56324140352600713 89"  Character,5
[13,] "18112999455363683800" Character,5
[14,] "16694499832575534222" Character,5
[15,] "17271741197267143976" Character,5
```

图 5.2 部分留言详情的 Simhash 向量

由上图可知代码输出的是十进制的数据，因此我们将所有文本的向量都 0-1 化，使用 tobin 函数 0-1 化。

表 5.5 留言详情二进制表示的 Simhash 值矩阵

列数	64
行数	2816

将所有的文本都化成 64 位二进制向量以后，在计算上减少了很多麻烦，以后我们的计算都是基于二进制向量。

5.5.1 相关性

在上面我们已经对相关性进行了定义，对于两个文本向量 A 和 B，它们的余弦值计算如下：

$$cos\theta = \frac{\sum_{i=1}^{p} A_i B_i}{\sqrt{\sum_{i=1}^{p} A_i^2 \cdot \sum_{i=1}^{p} B_i^2}}$$

下面我们将计算每一条留言与对应的答复意见之间的余弦值，部分结果如下：

```
 [1] 1.0000000 1.0000000 0.9338523 0.8752736 0.8740284 0.8574929 0.8571429 0.8458258 0.8406680
[10] 0.8336550 0.8332381 0.8255008 0.8226127 0.8224396 0.8207827 0.8116794 0.8107113 0.8081352
[19] 0.8058665 0.8035074 0.8004448 0.8000947 0.8000947 0.7951923 0.7951923 0.7901837 0.7895420
[28] 0.7894737 0.7879093 0.7862159 0.7846716 0.7840702 0.7833495 0.7827804 0.7824608 0.7798129
[37] 0.7768986 0.7758098 0.7745967 0.7745967 0.7742781 0.7735268 0.7734021 0.7730012 0.7726911
[46] 0.7715885 0.7714286 0.7714286 0.7700514 0.7694838 0.7694838 0.7692308 0.7683065 0.7677002
[55] 0.7671953 0.7667485 0.7667485 0.7655318 0.7650369 0.7650369 0.7610194 0.7607258 0.7606388
[64] 0.7589709 0.7579238 0.7578647 0.7566444 0.7566444 0.7556097 0.7546729 0.7541997 0.7538191
[73] 0.7503665 0.7470422 0.7470422 0.7463518 0.7463518 0.7463518 0.7445694 0.7442084 0.7440729
[82] 0.7440729 0.7431605 0.7431605 0.7414673 0.7412493 0.7403515 0.7397954 0.7378648 0.7369555
[91] 0.7368421 0.7368421 0.7349684 0.7342172 0.7337411 0.7330490 0.7330490 0.7329409 0.7324692
[100] 0.7324670 0.7324670 0.7324670 0.7321403 0.7321403 0.7309880 0.7309880 0.7307981 0.7307981
[109] 0.7299964 0.7299964 0.7299964 0.7299964 0.7297297 0.7292929 0.7292929 0.7286121 0.7278253
[118] 0.7276069 0.7273341 0.7272727 0.7270477 0.7265477 0.7261082 0.7258662 0.7253236 0.7252267
[127] 0.7247431 0.7247431 0.7247138 0.7245688 0.7229703 0.7225010 0.7222222 0.7221854 0.7219295
[136] 0.7216054 0.7203603 0.7201579 0.7201579 0.7200640 0.7200640 0.7200640 0.7191012 0.7191012
[145] 0.7189966 0.7187500 0.7171372 0.7171372 0.7169282 0.7169242 0.7164977 0.7164977 0.7164977
[154] 0.7164977 0.7161149 0.7156264 0.7154548 0.7154548 0.7145774 0.7145774 0.7142857 0.7140055
[163] 0.7130241 0.7130241 0.7130241 0.7129310 0.7123956 0.7118321 0.7111590 0.7111590 0.7108187
[172] 0.7107725 0.7105263 0.7102848 0.7102848 0.7098728 0.7097954 0.7089176 0.7087836 0.7084473
[181] 0.7077761 0.7077761 0.7077761 0.7075276 0.7071068 0.7071068 0.7071068 0.7068454 0.7068454
[190] 0.7061879 0.7061879 0.7061388 0.7059781 0.7058824 0.7056503 0.7056423 0.7046643 0.7042952
[199] 0.7042952 0.7042147 0.7042147 0.7029595 0.7029595 0.7029595 0.7027819 0.7027819 0.7027642
```

图 5.9 留言与答复意见的余弦值

以上得到的结果是从大到小排序以后的，通过上述结果可以看出相关部门给出的回复和留言的相关性很强，表明有些部门在工作上是秉着为人民服务的宗旨在办事；当然也有相关性很小的，则需要加强相关部门的管理，做到网友的问题都能得到相应的答复。

5.5.2 完整性

完整性的度量我们上述表明了使用答复意见对留言的覆盖情况来表示，以下是我计算的部分结果：

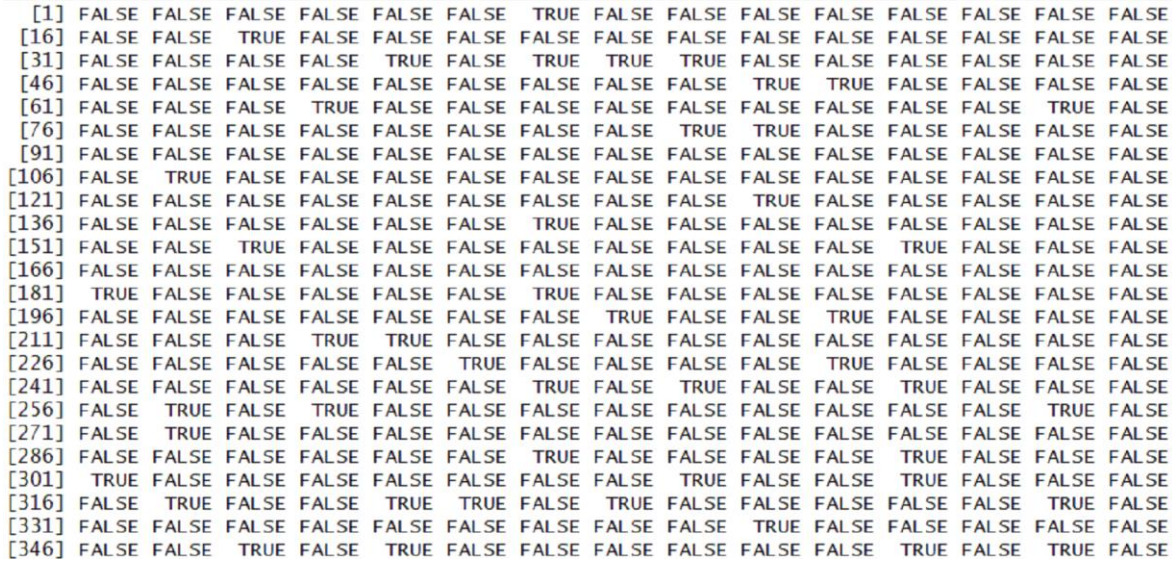

图 5.10 答复意见对留言的覆盖（部分）

当结果为 TRUE 时满足完整性，否则不满足。从上面的得到的结果看，只有极少数答复意见是满足完整性的，也就是说很多答复意见并没有完全解决留言的问题，这就表现出了相关部分在处理事情上的效率问题，若能在网上将网友提出的问题给出对应的答复，这将大大地提高社会管理能力。

5.5.3 可解释性

根据上述可解释性的定义，这里最主要就是找关键词的词向量的近义词，也就是找到词向量的近似向量的词，而如今是否能有词向量识别出近义词或者相关词还是一个值得探究的问题。语料库的不完整，再加上计算量非常大，则会导致操作时间过长并且正确率不高，该指标无法通过实际的计算来得到实现。

六、不足与展望

本文在进行文本分类时并没有解决类别样本缺失的问题，以及采取欠采样可能会丢失一部分重要信息。希望之后能够尽量获取到一些属于其他一级标签的政务留言样本数据，另外对欠采样方法进行研究，尽量避免重要信息的丢失。

在问题二中，用 Simhash 模型提取文本相似度虽然效果较好，但在选择相似的距离阈值上，没有给出合理科学的选择方法，因此仍有一些相似的文本数据被判断为不相似。另外在 DBSCAN 聚类之前，随机选择核心点，以及通过观察数据选取参数不

是一个科学的做法。而且 DBSCAN 算法无法聚类多密度数据集的问题，这可以从多角度进行考虑实现，不应该仅限于在对原始 DBSCAN 算法的基础上改进，可以从密度峰、数据流、无向图等方面展开实验，实现多密度聚类。

在问题三中，虽然我们都给出了相关性、完整性、可解释性的定义，但是否能通过词向量识别出近义词或者相关词还是一个值得探究的问题。即使可以利用词向量找到近义词，在这道题上计算量非常庞大的，将会导致操作时间过长并且正确率不高。对于通过词向量是否可以识别出近义词这个问题，还需要进一步研究，这个将会涉及到语料库的完整性问题。文本处理的发展与语料库的完善息息相关，尤其是现在出现的网络新词。

参考文献

[1] Shi, Yong. “Multiple criteria optimization-based data mining methods and applications: a systematic survey.” *Knowledge and Information Systems* 24.3(2010):369-391.

[2] 赵卓真. 一种基于密度与网格的聚类方法[D]. 中山大学.

[3] Duan, Lian, et al. “A Local Density Based Spatial Clustering Algorithm with Noise.”*Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on* IEEE, 2006.

[4] 姜建华, 杨玉免, 边海燕,等. 改进 DBSCAN 聚类算法在电子商务网站评价中的应用[J]. 吉林大学学报:理学版, 2013, 54(02).

[5] 冯振华. 基于 DBSCAN 聚类算法的研究与应用[D]. 2016.

[6] 王实美. 基于 DBSCAN 的自适应非均匀密度聚类算法研究[D]. 2017.

[7] 徐懿彬. 基于 Aho-Corasick 自动机算法的概率模型中文分词 CPACA 算法[J]. 电子科技大学学报, 2017(2).

[8] 官琴, 邓三鸿, 王昊. 中文文本聚类常用停用词表对比研究[J]. 数据分析与知识发现, 2017, 1(3):72-80.

[9] Qu Z, Song X, Zheng S, et al. Improved Bayes Method Based on TF-IDF Feature and Grade Factor Feature for Chinese Information Classification[C]// IEEE International Conference on Big Data & Smart Computing. 2018.

[10] ZHANG jiefu, XING Mingsun, NIGELL, et al. Achieving effective cloud search service Multi-keyword ranked search over encrypted Cloud data supporting synonym query [J]. IEEE Transactions on Consumer Electronics, 2014, 60(1):164-172.

[11] ManKu G S, Jain A, Das Sarma A. Detecting near-duplicates for web crawling[C]. Proceedings of the 16th international conference on World Wide Web, ACM, 2007:141-150.

[12] Paradimitriou P, Garcia-Molina H, Dasdan A. Web graph similarity for anomaly detection[J]. Journal of Internet Services and Applications, 2010, 1(1):19-30.

[13] 王源. 一种基于 Simhash 的文本快速去重算法[D]. 吉林大学.

[14] 谢瑶兵. 基于特征串的网页文本并行去重算法[J]. 微电子学与计算机, 2015, 000(002):69-72.

[15] 张荣葳. 基于 Simhash 与神经网络的网络异常检测方法研究[J]. 电脑知识与技术, 2019, 015(018):224-226.