“智慧政务”中的文本挖掘应用

摘　　要

随着网络问政平台的不断发展，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。此时利用自然语言处理和文本挖掘的方法解决群众留言分类、热点问题挖掘和答复意见等问题就显得尤为重要。

针对问题一，首先进行文本清洗，删除停用词；然后分词处理，遍历数据文件每一条记录，得到关键字，统计关键字字频，每一类留言选取字频较高及不重叠的 8 个关键字，将之向量化及归一化得到新的矩阵；再建立 SVM 模型，结合二类分类 SVM 原理和成对分类方法依靠每条留言的关键字对留言进行分类，最后取适量的样本作为训练集和测试集,并使用 F-Score 对分类方法进行评价。

针对问题二，首先进行问题识别，从所给的数据中识别出相似的留言，对相似问题的留言标记；然后进行问题归类，把相似留言归类为同一问题，并将特定地点或人群的数据进行归并，结果写入对应表格；最后进行热度评价，定义一个热度评价指标，对问题归类后的数据计算指标，并按该指标排序，结果写入对应表格。

针对问题三，我们对留言的答复意见采用百分制的评价制度，分别从答复的充实性、及时性、可解释性、相关性以及完整性四个方面对答复意见进行量化评价，以建立衡量相关部门专业水平和回复态度等方面的综合数学模型，最后得到各条留言回复的量化评价分数。

关键词: SVM 模型； 停用词；热度指标；量化评价

Abstract

With the continuous development of the network political platform, the amount of text data related to various social situations and public opinions is constantly increasing, which brings great challenges to the work of the relevant departments that mainly rely on manual workers to divide messages and sort out hot spots. It is particularly important to use the methods of natural language processing and text mining to solve the problems such as the classification of people's comments, the mining of hot issues and the answering of comments.

For problem 1, firstly, text is cleaned and stop words are deleted. Then word segmentation, traversing each record in the data file, to obtain keywords, statistics keyword word frequency, each type of message selected high frequency and non-overlapping 8 keywords, vectorization and normalization to get a new matrix; Then, the SVM model was established, and the SVM principle of binary classification and paired classification method were combined to classify the comments based on the keyword of each message. Finally, an appropriate amount of samples were taken as the training set and test set, and f-score was used to evaluate the classification method.

For problem 2, firstly, identify the problem, identify the similar message from the given data, and mark the message for similar problem; Then categorize the questions, classify similar comments as the same question, and merge the data of specific places or people, and write the results into the corresponding table; Finally, a heat evaluation index is defined to calculate the heat index of the data classified by the problem, and the results are written into the corresponding table.

For problem 3, the evaluation system we built is based on hundred mark system, respectively from the reply of completeness, timeliness, interpretability, relevance and completeness of quantitative evaluation on the four aspects to answer the opinion, to set up relevant departments to professional standard and reply attitude and so on comprehensive mathematical model, finally get the quantitative evaluation of every message reply scores.

Key words: SVM model ; Stop words ; Heat index ; Quantitative evaluation

目录

1、挖掘背景和目标

1.1 挖掘背景

近年来，网络问政平台逐步成为政府了解民意的重要渠道，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。在此形势下，用自然语言处理和文本挖掘的方法解决群众留言分类、热点问题挖掘和答复意见等问题就显得尤为重要。

1.2 挖掘目标

本文根据问题建立数学模型，并设计求解方法解决如下问题：

问题一：群众留言分类

参考附件 1 提供的内容分类三级标签体系，综合考虑 F-Score 对分类方法进行评价的式子，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。

问题二：热点问题挖掘

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，按表 1 的格式给出排

名前 5 的热点问题，并保存为文件“热点问题表.xls”。按表 2 的格式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”

问题三：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、问题分析

经我们初步的分析，该问题实际上是一个用自然语言处理和文本挖掘的方法解决群众留言分类、热点问题挖掘和答复意见的问题。

2.1 问题一的分析

经我们初步的分析，该问题是一个文本分类问题。具体步骤如下：

（1）文本清洗：删除原始数据中不完全的、有噪声的 、模糊的停用词。

（2）分词处理：运行得到关键字，遍历数据文件的每一条记录，将每句话拆散成单独一个字符，一个字保存在一个单元格，如此形成 1*N 的数组。

（3） 统计关键字字频，每一类留言选取字频较高及不重叠的 8 个关键词，将之向量化。495 个留言按关键字来统计数据：在每条留言中，某个关键词每出现一次，就在对应的 56 个关键字的位置加 1，1*56 的矩阵就代表了其中一条留言。最后归一化，将 1*56 的矩阵中逐个数字加起来，1*56 的矩阵中每个值再除以其和的值得到新的矩阵。一直到第 495 条留言也是这样处理。

（4）建立核支持向量机模型，结合二类分类 SVM 原理和成对分类方法依靠每条留言的关键字对留言进行分类。

（5）取 381 个样本作为训练集并利用分类算法训练分类器 ，使得分类器能尽量识别特定类别的文本特征，取 113 个样本作为测试集来测试已被训练的分类器的效果。,并使用 F-Score 对分类方法进行评价。

2.2 问题二的分析

问题二要求根据附件 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，求出排名前 5 的热点问题并给出相应热点问题对应的留言信息，最后按要求保存数据。

经我们结合问题对留言主题、留言详情等数据的初步分析，该问题可以分为三个子任务：

（1）问题识别：即如何从所给的数据中识别出相似的留言，这里就涉及到文本相似的问题，需要我们通过计算两条留言间的相似度，然后判断是否为相似问题，如果相似的问题留言就打上对应标签。

（2）问题归类：即把特定地点或人群的数据进行归并，当我们把问题识别完成之后，我们就需要对问题进行归类，把相似的留言归为同一问题，提取出特定地点或人群的信息，再将结果写入对应表格。

（3） 热度评价：即定义好一个热度评价指标和计算方法，对归类后的结果计算好热度指标，并按该指标进行排序，排序后的结果写入对应表格。

2.3 问题三的分析

问题三要求我们针对相关部门对留言的答复意见，从答复的各个角度对答复意见的质量给出评价方案，并尝试实现。于是我们对留言的答复意见采用满分 100 分制的评价制度，分别从答复的充实性、及时性、可解释性、相关性以及完整性四个方面对答复意见进行量化评价，以建立衡量相关部门专业水平和回复态度等方面的综合数学模型，最后得到了各条留言回复的量化评价分数。

3、模型假设

为了方便建模，我们做出如下合理的假设：

（1）假设各附表给出的数据真实可信

（2）收集的数据经过分析，均为满足要求的有效数据，据此得到的结论有效

（3）假设政务留言中的留言主题都是经过人的语言文字处理，对所留言的事件进行了有效的较为简短的概括

4、符号说明

问题一：

符　号	含　　义
x_i	输入空间
y_i	输入空间对应的标记
d_{ij}	表示 i 类留言和 j 类留言之间的决策边界
上标ij	第 i类和第 j 类之间二类分类 SVM 的参数
下标 t	i 类和 j 类的并集中样本的索引
y_{new}^{ij}	第 i类和第 j 类之间二类分类 SVM 的决策函数

表 1

问题二：

符　号	含　　义
D_i	第i条留言的热度指标值
D	同一问题留言的热度指标值总和
t	最近的留言时间与实际留言的时间之差
W_{t_i}	第i条留言的时间权重值
W_{a_i}	第i条留言的支持度权重值
N_{a_i}	第i条留言的点赞数
N_{d_i}	第i条留言的反对数

表 2

问题三：

符　号	含　　义

T_i	收到留言时间
T_i'	回复留言时间
L_i	第 i 个问题回复的有效文本长度
Q_c	第i条的问题描述
D_{1i}	第i条的问题描述的长度
A_i	第i条的答复描述
D_{2i}	第i条的答复描述的长度
F_i	第i条的答复的相应相关性（频率）
R_c	第i条的答复的完整性（匹配个数）
C_p	两个字以上的句式短语的集合
f_k	p_k在A_i中出现的次数
D_k	两个字以上的词
Score_1i	对留言回复的及时性评价的分值
Score_2i	对留言回复的内容充实性评价的分值
Score3_1i	对留言回复的相关性评价的分值
Score3_2i	对留言回复的完整性评价的分值
Score_i	对留言回复评价的总分值

表 3

5、分析方法与过程

5.1 问题一分析方法与过程

5.1.1 模型建立与求解

（1） 文本清洗：

大多数情况下，原始数据中都有很多不完全的、有噪声的 、模糊的部分，一类是人类语言中包含的功能词，这些功能词极其普遍，与其他词相比，功能词没有什么实际含义；另一类的词应用十分广泛，但是对这样的词搜索引擎无法保证能够给出真正相关的搜索结果，难以帮助缩小搜索范围，同时还会降低搜索的效率，所以通常会把这些词从问题中移去，从而提高搜索性能。例如全半角标点符号、大小写英文、停用词，这些词作为噪音，会影响后续建立模型，需要分步骤清洗。

在本例中，删掉例如“的么了吗如何把应该于是至我你她他啊阿一二三四五六七八九十零从来东西南北难大小上下左右中点职位无关省市县不为在们和个这出那里前后到多少以能也主可广么作做就去子时说发用者定事好情没动本天对过”等停用字、大写英文字母和数字，并将空格替换为空字符。继而提取人们事先未知的、但又是潜在有用的信息。

（2） 分词处理：

运行得到关键字，如图 1，遍历数据文件的每一条记录，将每句话拆散成单独一个字符，一个字保存在一个单元格，如此形成 1*N 的数组。

列 1 至 19

{'地'} {'山'} {'枣'} {'镇'} {'板'} {'托'} {'村'} {'村'} {'民'} {'父'} {'亲'} {'因'} {'有'} {'高'} {'血'} {'压'} {'糖'} {'尿'} {'病'}

列 20 至 38

{'经'} {'常'} {'要'} {'村'} {'卫'} {'生'} {'室'} {'量'} {'血'} {'压'} {'测'} {'血'} {'糖'} {'村'} {'卫'} {'生'} {'医'} {'生'} {'非'}

列 39 至 57

{'常'} {'热'} {'地'} {'接'} {'待'} {'觉'} {'得'} {'现'} {'国'} {'家'} {'实'} {'施'} {'惠'} {'民'} {'政'} {'策'} {'实'} {'行'} {'基'}

……

列 13853 至 13871

{'院'} {'工'} {'休'} {'息'} {'间'} {'每'} {'月'} {'只'} {'有'} {'只'} {'依'} {'靠'} {'短'} {'短'} {'缓'} {'解'} {'相'} {'思'} {'之'}

列 13872 至 13890

{'苦'} {'请'} {'问'} {'书'} {'记'} {'什'} {'候'} {'才'} {'享'} {'受'} {'国'} {'家'} {'法'} {'节'} {'假'} {'日'} {'轮'} {'休'} {'政'}

列 13891

{'策'}

图 1

（3）词袋模型：

词袋模型假设我们不考虑文本中字与字之间的上下文关系，仅仅只考虑所有字的权重。而权重与字在文本中出现的频率有关。词袋模型首先会进行分字，在分字之后，通过统计每个字在文本中出现的次数，就可以得到该文本基于字的特征，如果将各个文本样本的这些字与对应的字频放在一起，这就是所说的向量化。题目给出的是 7 个类别的留言，将这 7 个类别依次编号为 1，2，••••••，7。先统计关键字字频，每一类别留言取字频较高及不重叠的 8 个关键字，将之向量化得到 1*56 的零矩阵（如下图），每一类分别取的是 8 个关键字，把这 8 个关键字编码。

一共 495 个样本，得到 495*56 的零矩阵，前 8 列代表第一类，9—16 列代表第二类，以此类推。如图 2 所示。

类别 1　　类别 2　　类别 3　　类别 4　　类别 5　　类别 6　　类别 7

图 2

1-8 列	9-16 列	17-24 列	25-32 列	33-40 列	41-48 列	49-56 列

495 个留言按关键字来统计数据：在每条留言中，某个关键字每出现一次，就在对应的 56 个关键字的位置加 1，1*56 的矩阵就代表了其中一条留言。最后归一化，将 1*56 的矩阵中逐个数字加起来，1*56 的矩阵中每个值再除以其和的值得到新的矩阵。一直到第 495 条留言也是这样处理。

（4）SVM（核支持向量机）模型：。

在此处使用了 matlab 内置的 SVM 模板构建训练器，其中使用了高斯核函数，其中 γ 属于超参，要求大于 0，需要调参定义。

$$\boldsymbol{K(x,z) = e^{-\gamma||x-z||_2^2}} \qquad \boldsymbol{(1)}$$

该题目一共有 7 个分类，我们用 SVM 模型解决留言有 7 个类的分类问题。按照 SVM 模型原理[1]，先取 n 个样本作为训练集$X = \{(x_1, y_1), (x_2, y_2)$，……，$(x_n, y_n)\}$，其中$k$维向量$x_n \epsilon R^k$，类标签$y_n \epsilon \{1,2,\cdots\cdots, 7\}$，$n = 1,\cdots\cdots, 381$，利用成对分类方法[2]将 7 类留言两两比较，一共得到 21 个决策边界，d_{ij}表示 i 类留言和 j 类留言之间的决策边界$(i, j \epsilon \{1,2,\cdots\cdots, 7\})$。对于第 i 类和第 j 类留言，训练一个二类分类 SVM 来求解二次规划问题：

$$\min_{w^{ij},b^{ij},\delta^{ij}} \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \delta_t^{ij}$$

使得当$y_t = i$ 时有 $\left(w^{ij}\right)^T \varphi(x_t) + b^{ij} \geq 1 - \delta_t^{ij}$ （2）

当$y_t = j$ 时有 $\left(w^{ij}\right)^T \varphi(x_t) + b^{ij} \leq -1 + \delta_t^{ij}$

$$\delta_t^{ij} \geq 0$$

其中，上标表示是第 i类和第 j 类之间二类分类 SVM 的参数；下标 t 表示 i 类和 j 类的并集中样本的索引；φ表示输入空间到特征空间的非线性映射。索引 i,j$\epsilon\{1,2,\cdots\cdots，7\}, i < j$，一共需训练 $C_7^2 = \frac{1}{2} \times 7 \times 6 = 21$个二类分类 SVM。平均每个类包含 $\frac{381}{7}$ 个样本，所以平均每个对偶问题包含 $\frac{2\times381}{7}$个变量。

第 i类和第 j 类之间二类分类 SVM 的决策函数：

$$y_{new}^{ij} = sign\left[\left(w^{ij}\right)^T \varphi(x_t) + b^{ij}\right] = sign \sum_{support\ vectors} y_t a_t^{ij} k(x_{t,} x_{new} + b^{ij})$$

（3）上式用于判断数据是属于第 i类还是第 j 类。

将数据初步按照 23:77 的比例划分为测试集和训练集，采用 SVM 训练模型，我们的样本集为(x_i, y_i)，$x_i = (x_1, x_2, x_3, \ldots, x_n)$，$y_i \in \{1,2,3,\ldots,7\}$，模型的训练过程即为寻找支持向量，确定最大超平面的过程。对于新数据分类，此分类实际上是多类分类问题，即类别数 $k \geq 3$，对于多分类 SVM 分类问题，解决办法如下：把类l作为一类，其余的$k-1$类看成另一类，这样就把k分类问题转化为二分类问题，如图 3 SVM 分类示意图 所示。

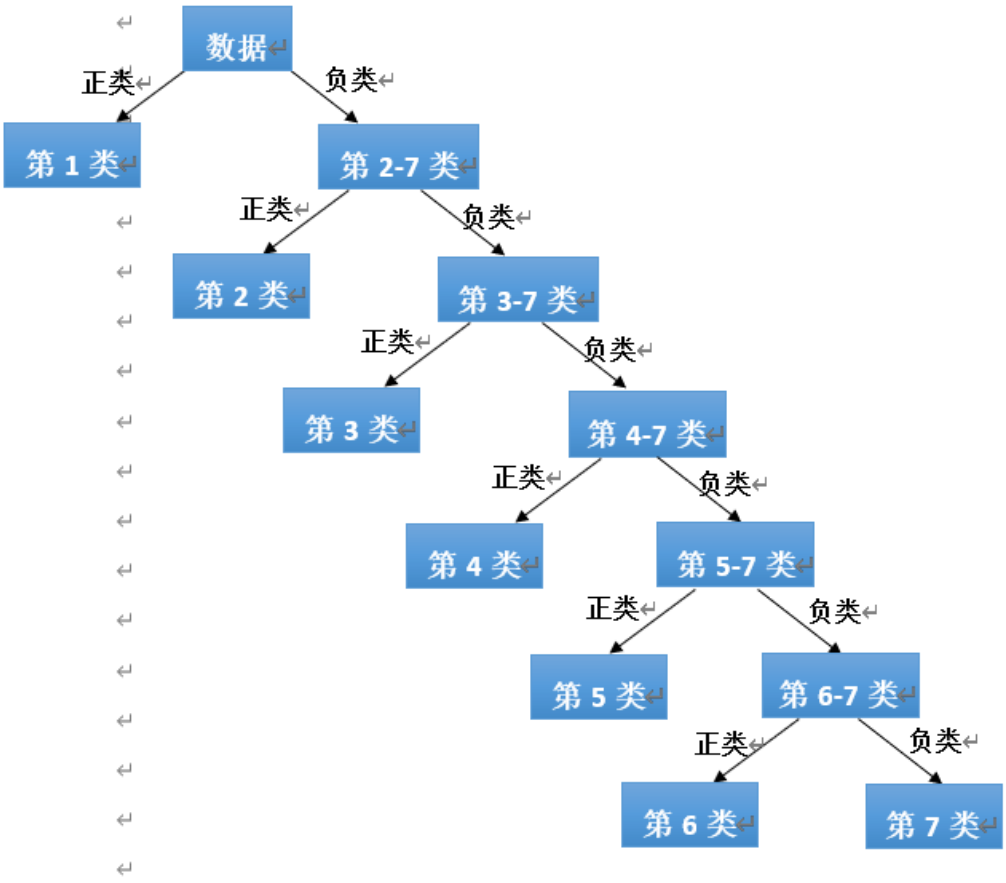


图 3 SVM 分类示意图

（5）取 381 个样本作为训练集并利用分类算法训练分类器，使得分类器能尽量识别特定类别的文本特征，取 113 个样本作为测试集来测试已被训练的分类器的效果。,并使用 F-Score 对分类方法进行评价：

$$F_i = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i} \qquad (4)$$

其中P_i为第i类的查准率，R_i为第i类的查全率。

处理自然语言一般步骤如图 5 处理自然语言一般步骤图 5 处理自然语言一般步骤所示，测试样本数据如图 4 测试样本数据

XTest

113x56 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0.1429	0	0	0	0.0714	0	0.0714	0	0	0	0	0	0	0	0	0	0	0	0.0714
2	0	0.5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0.1000	0.1000	0.1000	0	0.1000	0	0	0.1000	0.2000	0	0	0	0	0	0
4	0	0.2000	0.2000	0.2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0.5000	0	0	0	0.5000	0	0	0	0	0	0	0	0
6	0.3077	0	0.0385	0.1154	0	0.1923	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0.0313	0	0	0.1250	0.0313	0.2188	0.0938	0	0.0313	0	0	0	0.0938	0.0938	0	0	0	0.0313	0	0
9	0	0	0	0	0	0	0	0.2000	0	0	0	0	0	0	0	0	0	0	0	0
10	0.1290	0.1290	0	0.0645	0.0645	0.0968	0.0645	0	0	0.0323	0.0323	0	0	0	0	0	0	0	0	0
11	0	0.1667	0	0	0.0833	0	0	0	0.0833	0.0833	0	0	0.0833	0	0	0	0	0	0	0
12	0	0	0	0	0	0.2500	0	0.1250	0	0	0	0.1250	0	0.1250	0	0	0	0	0	0
13	0	0	0.0952	0.0952	0	0.0476	0.0476	0	0	0	0	0	0	0	0.4762	0	0	0	0	0
14	0.0870	0	0	0.1304	0.0435	0	0.0435	0	0	0	0	0	0	0	0	0	0	0.0870	0	0
15	0	0	0	0.1250	0	0.1250	0.1250	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0.0213	0.3617	0.0213	0.0638	0	0.0213	0.0213	0	0	0.0213	0	0	0.0213	0	0	0	0	0	0	0
17	0.0571	0.1429	0.0571	0.0571	0.0095	0.0762	0.1524	0	0.0095	0	0	0.0095	0	0	0.1143	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0.2000	0.4000	0	0	0	0	0.2000	0
20	0.0714	0.0714	0	0.0714	0	0.0714	0	0	0	0	0	0.0714	0	0.1429	0	0	0	0	0.0714	0
21	0.0923	0.1077	0	0.0615	0	0.0769	0.0308	0.0154	0.0615	0	0	0.0154	0	0	0	0	0	0	0.1231	0
22	0.1250	0.2188	0.0625	0.0313	0	0.0313	0	0.0313	0	0	0	0	0	0	0	0	0.0313	0	0	0.0313
23	0.2057	0.0171	0.0571	0.0857	0	0.0914	0.0343	0.0057	0	0.0114	0	0	0	0.0057	0	0	0.0457	0.0800	0	0
24	0.0074	0.0667	0.0444	0.1852	0	0.0296	0.0444	0.0148	0	0.0222	0	0	0	0	0	0	0	0.0074	0	0
25	0	0	0	0	0	0	0	0	0.0667	0.0667	0.0667	0	0.0667	0	0	0	0	0	0	0
26	0.2131	0.0656	0	0	0	0	0	0	0	0	0	0	0.0492	0.0820	0	0.0492	0	0	0	0
27	0.0392	0.0196	0.0392	0	0	0	0.0196	0.0392	0.0196	0.0196	0.0196	0.0392	0	0	0	0.1765	0	0	0	0
28	0	0.1481	0	0	0	0	0.1852	0	0.0370	0	0.0370	0.0370	0.0370	0	0.0370	0.1111	0	0	0	0
29	0	0	0.0968	0.0323	0	0.0323	0.1290	0	0	0.1290	0.1613	0.0323	0	0	0.1290	0.0645	0	0	0.0323	0
30	0	0.0217	0.0217	0	0	0.0217	0	0	0	0.0652	0	0.0435	0.2826	0.0435	0.0652	0	0	0.0217	0	0
31	0.0070	0.1197	0.0141	0.1338	0	0.0352	0.0493	0.0070	0.0070	0.0070	0	0	0.0493	0	0	0	0.0141	0	0.0986	0
32	0	0.0833	0	0.0833	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0833	0.1667	0
33	0	0	0	0.2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2000	0.2000	0
34	0	0	0	0	0.3333	0	0	0	0	0	0	0	0.1667	0	0	0	0	0	0	0
35	0	0.1111	0.1111	0	0	0	0	0.1111	0	0	0	0	0.1111	0	0	0	0	0	0	0
36	0.2000	0	0	0.2000	0	0	0	0	0	0	0	0	0.2000	0	0	0	0	0	0	0
37	0	0	0	0.3333	0	0	0	0	0	0	0	0	0	0	0	0	0.1667	0	0	0

图 4 测试样本数据

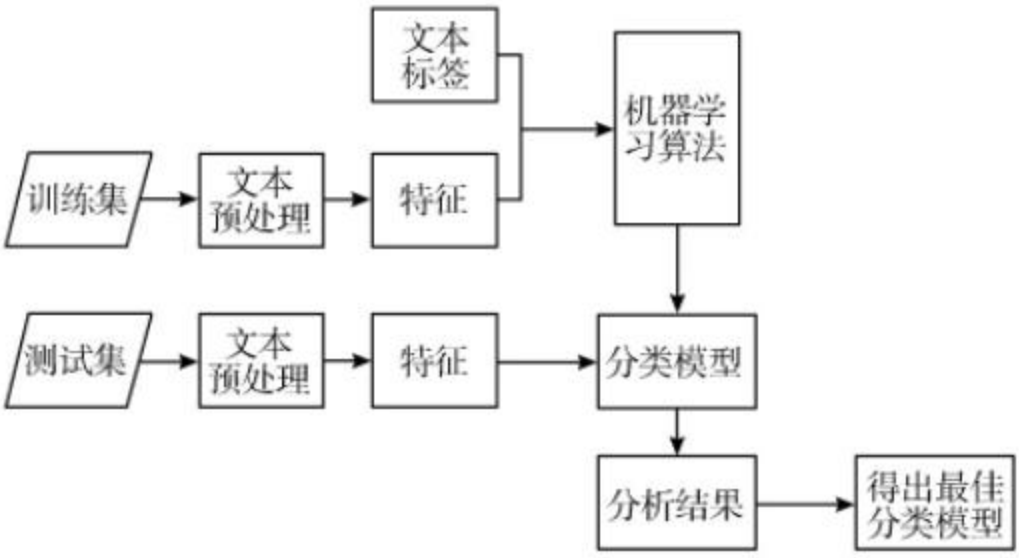


图 5 处理自然语言一般步骤

本例分类步骤如图 1 图 6

图 6　本例分类步骤

5.1.2 结果分析：

使用该模型建立的关于留言内容的一级标签分类模型是比较准确的，得到准确率是 Pp=0.9292，F-score=0.9153（如图 7　）

```
Pp =

    0.9292


Lpss =

    0.0709


F_Score =

    0.9153
```

图 7

5.2 问题二分析方法与过程

5.2.1 模型建立与求解

（1）问题识别：

我们分析后认为，在政务留言中的留言主题，都是经过人的语言文字处理，对所留言的事件进行了有效的较为简短的概括，当然一般来说，实际生活中的确也如此，因此我们只需对留言主题进行处理，降低处理难度。问题识别的核心是从所给的数据中识别出相似的留言，这就是计算机领域中自然语言处理的文本相似判断。在做分类时常常需要估算不同样本之间的相似性度量(Similarity Measurement)，这时通常采用的方法就是计算样本间的“距离”(Distance)或相关系数。采用什么样的方法计算距离或相关系数很是讲究，甚至关系到分类的正确与否。

由于缺少与赛题相关度高的语料，自行创建相关语料非常耗时耗力，使用已有的语料或训练好的模型效果也一般，有监督方法受语料影响非常大，对于这些数据上的效果非常平淡；而无监督方法能有相当好的效果，却不需要大体积的语料或模型文件，也不需要太强悍的硬件性能；因此我们首选无监督方法来实现文本相似度的计算。经过我们的大量测试，结合实际情况，我们总结出对于本题比较可行的文本相似度计算相关的模型算法：

首选 Jaccard 系数的算法，和 TF-IDF 算法，以及有监督方法中的 RoBerta/Bert 模型，实际上，如果硬件性能比较好的话，可以同时使用多

种算法，再将他们的结果取交集，这样分类结果的准确性就有更大的保证，所以我们在这里也保留多种算法，视情况切换或同时采用；同时我们也想出一种算法，即提取每条留言的关键字，与上一问中的分类结合分析，分类出相关的大类，再进行分析。

Jaccard 算法是用于比较有限样本集之间的相似性与差异性。Jaccard 的系数值越大，样本相似度越高。Jaccard 算法主要用于计算符号度量或布尔值度量的样本间的相似度。若样本间的特征属性由符号和布尔值标识，无法衡量差异具体值的大小，只能获得“是否相同”这样一种结果，而 Jaccard 系数关心的是样本间共同具有的特征。

杰卡德算法的定义如下：两个集合 A 和 B 交集元素的个数在 A、B 并集中所占的比例，称为这两个集合的杰卡德系数，用符号 J(A,B) 表示。

$$\mathbf{J(A,B)} = \frac{|\boldsymbol{A}\cap\boldsymbol{B}|}{|\boldsymbol{A}\cup\boldsymbol{B}|} \qquad (5)$$

对于中文文本来说，两个集合分别表示的是两个文本，集合中的元素实际上就是文本中出现的词语，这里我们用机器学习库 sklearn 中的 CountVectorizer 将两个文本中的词语文本特征提取提取出来。

TF-IDF 算法（Term Frequency - InverseDocument Frequency）是一种用于资讯检索与文本挖掘的常用加权技术算法模型。在一个文本中，当一个词汇出现很多次时，我们往往认为这个词是重要的，可以代表该文本。但是事实不是这样的，比如:“的”这个词，虽然在一个文本中出现很多次，但是它依然没有什么实际意义。而人们想要给文本中每个词在语义表达中，赋予一定的权重，就出现了 TF-IDF 算法。该算法优点是算法的容易理解，

便于实现；缺点是 IDF 的简单结构并不能有效地反映单词的重要程度和特征词的分布情况，使其无法很好的完成对权值的调整功能，所以在一定程度上该算法的精度并不是很高。除此之外，算法也没体现位置信息，对于出现在文章不同位置的词语都是一视同仁的，而我们知道，在文章首尾的词语势必重要性要相对高点。据此，我们可以或许也可以将处于文章不同位置的词语赋予不同的权重。

TF-IDF 的主要思想是：如果某个词或短语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。

TF-IDF 实际是 TF*IDF，其中 TF（Term Frequency）表示词条在文档中的出现的频率，其中 TF 的计算公式如下所示：

$$\mathbf{TF} = \frac{\text{词汇在文本中出现的次数}}{\text{文本词汇的总个数}} \qquad (6)$$

理论上，词汇出现的次数就应该是词汇的频率。我们这里除以文本词汇总个数，是为了排除文本长度的影响，获取到该词在文本中的相对重要程度。IDF（逆文档频率）计算如下：

$$\mathbf{IDF} = \log\left(\frac{\text{语料库中文本的总人数}}{\text{包含该词汇的文本个数+1}}\right) \qquad (7)$$

上面我们说过，有一些虽然在一个文本中出现很多次，但是它依然没有什么实际意义，比如：“的”。而 IDF 逆文档频率是衡量该词汇是否可以充分表示该文本的参数。IDF 值越大，说明包含该词汇的文本越少，则该词汇越能够代表该文本。

TF-IDF 则是由 TF 和 IDF 相乘得到，如下：

$$\mathbf{TF-IDF}=\frac{\text{词汇在文本中出现的次数}}{\text{文本词汇的总个数}}*\log\left(\frac{\text{语料库中文本的总人数}}{\text{包含该词汇的文本个数+1}}\right)\quad(8)$$

它的含义是，如果一个词，在该文本中出现次数越多，而在其他文本中出现很少时，则该词汇越能够表示该文本的信息。这里我们使用机器学习库 sklearn 中的 TfidfVectorizer 将两个文本中的词语文本特征提取提取出来这样便可以对中文文本进行 TF-IDF 计算。

经过以上计算后，可得出文章中每个词的 TF-IDF 值之后，进行排序，选取其中值最高的几个作为文章的关键字；再从中各选取相同个数的关键词，合并成一个集合，计算每篇文章对于这个集合中的词的词频，生成两篇文章各自的词频向量，进而通过欧氏距离或余弦距离求出两个向量的余弦相似度，值越大就表示越相似。

RoBERTa/BERT 模型

RoBERTa 模型属于 BERT 的强化版本，也是 BERT 模型更为精细的调优版本，RoBERTa 主要在三方面对 BERT 做了该进，其一是模型的具体细节层面，改进了优化函数；其二是训练策略层面，改用了动态掩码的方式训练模型，证明了 NSP（Next Sentence Prediction）训练策略的不足，采用了更大的 batch size；其三是数据层面，一方面使用了更大的数据集，另一方面是使用 BPE（Byte-Pair Encoding ）来处理文本数据。

BERT 可以很好的解决 sentence-level 的建模问题，它包含叫做 Next Sentence Prediction 的预训练任务，即成对句子的 sentence-level 问题。

BERT 也给出了此类问题的 Fine-tuning 方案，如图 8：

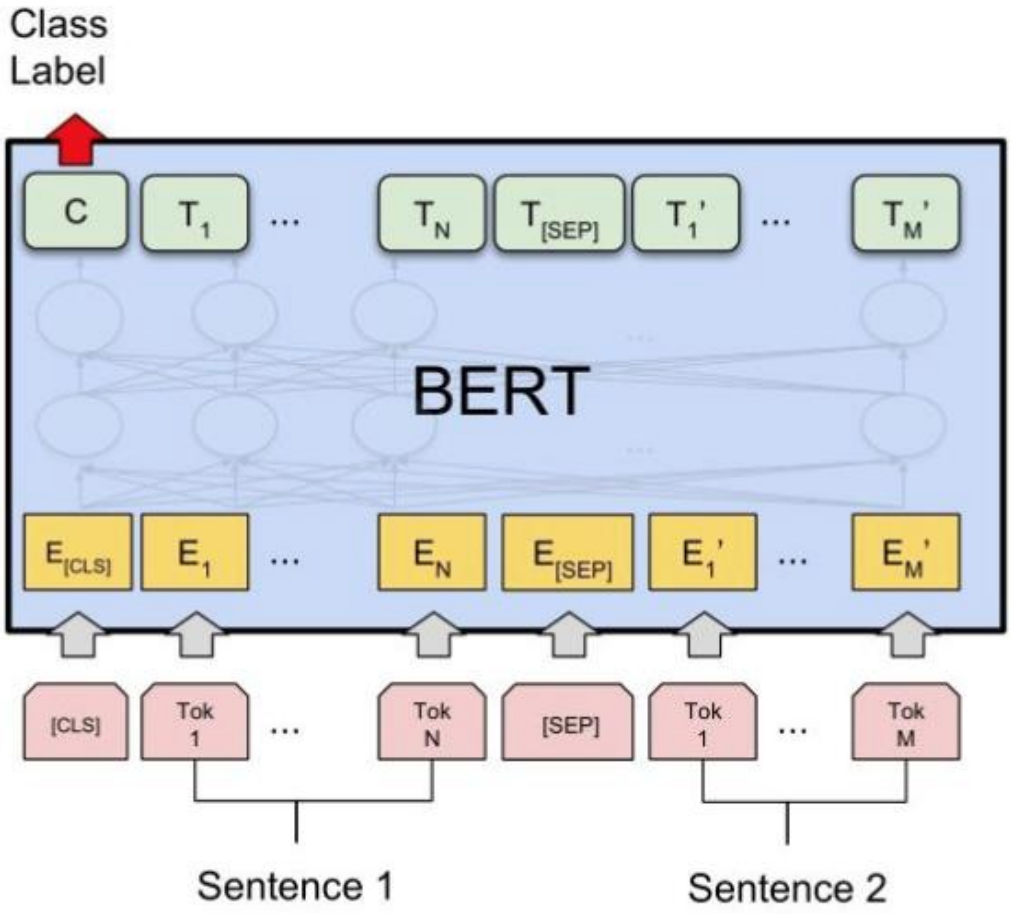


图 8

（2）对于问题归类：

对所有留言进行相似度计算后，需要提取出该类问题特定地点或人群的信息和给出问题描述。

经过分析，综合实际情况，我们认为对于提取特地的地点或人群的任务，采用自然语言处理（Natural Language Processing, NLP)的命名实体识别的 HMM 模型来实现。

隐马尔可夫模型（Hidden Markov Models, HMM）描述由一个隐藏的马尔可夫链随机生成的不可观测的状态序列，再由各个状态生成一个观测而

产生观测随机序列的过程，属于生成模型。HMM(隐马尔可夫模型)是结构最简单的动态贝叶斯网，是一种著名的有向(无环)图模型，主要用于时序数据建模（语音识别、自然语言处理等)。

HMM（隐马尔科夫模型）是统计概率模型，在 NER 使用的模式是已知观察序列(句子中的词)，求背后概率最高的标注序列(即每个字的分词状态)。在 NER 中，HMM 假设每个标注取决于前面的标注结果和当前的观察序列，构成如图 9 的概率图模型：

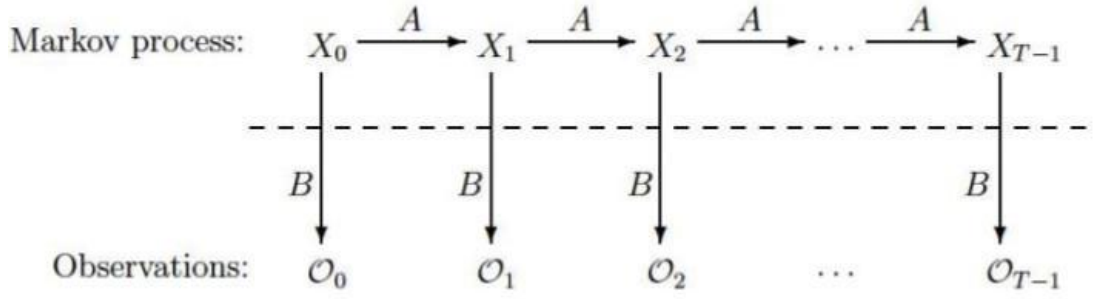


图 9

HMM 的转移概率模型

其中 A 表示上一个状态到下一个状态的转移概率矩阵，B 表示隐藏状态到当前的观测状态的转移概率矩阵，另外使用 s 表示初始状态。一个 HMM 模型通过构建（A，B，s）来表示序列概率。不过 HMM 的缺点从原理上也能看出：一个词的实体词类型，只取决于当前词以及前面的状态，无法考虑更远的词以及后面的词的影响，从而导致无法达到全局最优。因为 HMM 为了对联合概率分布进行建模，HMM 引入两条独立性假设：

马尔科夫链在任意时刻的状态 Xi 仅依赖于前一个状态 Xi-1；

任意时刻的观测 Oi 只依赖于该时刻马尔科夫链的状态 Xi。

经过多次测试,这里我们直接采用现成的 Hanlp 的命名实体识别模块，

该模块即采用了 HMM 在同一问题留言提取出特地的地点或人群，其基本流程是：

1）训练：对熟语料自动角色标注，统计单词的角色频次、角色的转移概率等，训练出一个模型，同时总结一些可用的模式串。

2）识别：根据上述模型，利用 HMM-Viterbi 算法标注陌生文本的粗分结果，利用 Aho-Corasick 算法模式匹配，匹配出可能的地址，将其送入第二层隐马尔可夫模型中。

3）从该类留言中选出占比最多的命名实体，作为该类问题的特地的地点或人群。

HMM 模型可以解决很多问题，将多个 HMM 模型层叠起来，可以发挥出更加精准的效果。不过 2 元文法依然会有误命中的情况，事实上，一些高频地名已经收录到核心词典和用户自定义词典中。在部分情况下，Hanlp 也可能无法识别出命名实体，这时候需要从留言中使用特定规则模糊搜索出命名实体。

对于同类问题的描述，我们对部分数据进行人工地概括描述，发现最后的结果与其中的留言主题非常相似，如图 10 所示：

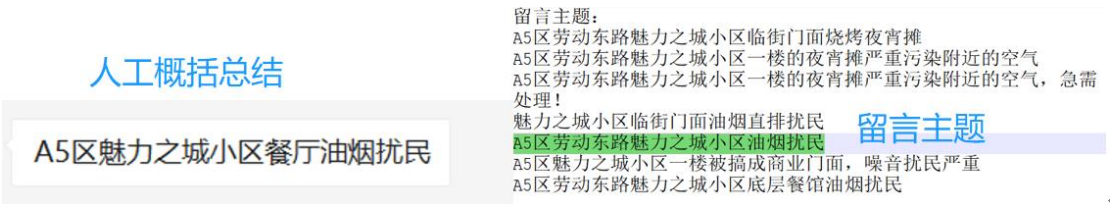


图 10

实际上在政务留言中的留言主题，可以认为都是经过人的语言文字处理，留言者对所留言的事件进行了有效的较为简短的概括。经过相似性分

类后的同一类留言主题，它们是对同一事件的不同描述，但是都有着共同的信息，这些共同信息可以描述该类问题；但是不同的人描述同一问题，可能多少都会有些差异，然而问题描述应该是对全部或大多数的留言都正确的，都可以被描述的，是可以代表该类问题的，而不是过多追求某些细节信息，更重要的是保证其不失真。

分析数据，并进行多次试验后，我们认为，问题描述需要先确保正确性，不过多追求精确性的细节。

经过分析和查找资料后，对于问题描述，我们决定采用如下方案：
用地点+动词/关键词（留言事件发生的行为动作或关键的词语）来从同类留言中选定一个较短的代表句，之后我们可以再对这个代表句进行适当精简（其实这些留言主题已经是相当的精简），便可作为问题描述，最后再将结果写入对应表格。
同时，我们也实现了用关键词造句形成问题描述的方案，在数据量较小时效果较好，但是大量数据时情况下不会首先使用
具体的实现如下图 11：

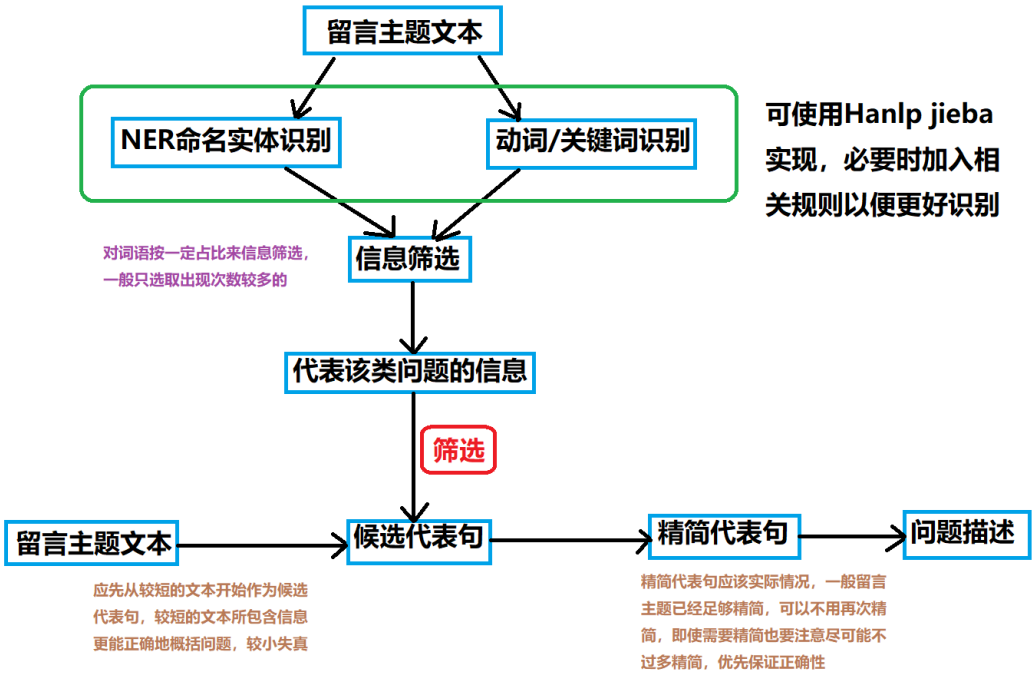


图 11

（3）对于热度评价：

参考许多热度评价资料，结合实际情况分析，我们认为对于目前的留言数据，设计热度评价指标时应当考虑同类问题留言数，留言时间，点赞数，反对数等信息。

对于同类问题留言数，应当是越多，热度越高；考虑到生活实际情况，同类问题留言数也应当有下限，如果只有一条留言，而它的点赞数很多，我们认为这应当视为异常数据排除，现实生活中一般都是热点问题多人反映，而不是只会去点赞；我们认为同类问题留言数至少有两条。

对于留言时间，应当是越近的热点越大，要对留言时间进行加权计算比较，这里我们选定了距离此刻北京时间或者文档内最近一条留言的时间（即文档内的最近时间） 30 天、60 天、90 天、120 天、180 天、365 天作为时间分割点，分别对不同时期的留言作加权计算处理，具体的加权值可以按具

体情况而定。一般每个时期之间的加权值差别越大，对应的热度指标差别也越大；也就是说每个时期之间的加权值差别越大，越久远的问题，就需要更多留言数等指标才能达到与更近时期的问题一样的热度；因此每个时期之间的加权值差别也不能太大，应按具体情况设定。

对于点赞数，反对数，显然点赞越多，热度越高；反对数越多，热度越低。这里我们对每条留言取点赞数减反对数的差值，作为群众对该条留言的关注程度。因为点击点赞/反对是非常容易操作的，这里还应考虑到人为刷点赞数/反对数的情况，所以一个点赞数/反对数的权重不能等于一条留言的权重，而是低于留言的权重；但是，也不能太低，否则点赞/反对就不能很好反映群众对该条留言的关注程度。

综合以上情况，我们设计了如下的热度评价指标计算公式：

同一问题的第 i 条留言的热度评价指标值:

$$\boldsymbol{D_i = 10 \times W_{t_i} \times [1 + W_{a_i} \times (N_{a_i} - N_{d_i})]} \qquad (9)$$

同一问题留言的热度评价指标值总和:

$$\mathbf{D} = \sum_{i=1}^{n} \mathbf{10} \times \boldsymbol{W}_{t_i} \times [\mathbf{1} + \boldsymbol{W}_{a_i} \times (\boldsymbol{N}_{a_i} - \boldsymbol{N}_{d_i})] \qquad (10)$$

$$\boldsymbol{W}_{t_i} = \begin{cases} \mathbf{1.2}, & \boldsymbol{t} \leq \mathbf{30} \\ \mathbf{1}\ , & \mathbf{31} \leq \boldsymbol{t} \leq \mathbf{60} \\ \mathbf{0.8}, & \mathbf{61} \leq \boldsymbol{t} \leq \mathbf{90} \\ \mathbf{0.6}, & \mathbf{91} \leq \boldsymbol{t} \leq \mathbf{120} \\ \mathbf{0.3}, & \mathbf{121} \leq \boldsymbol{t} \leq \mathbf{180} \\ \mathbf{0.1}, & \mathbf{181} \leq \boldsymbol{t} \leq \mathbf{365} \\ \mathbf{0}, & \boldsymbol{t} > \mathbf{365} \end{cases} \qquad (11)$$

$$\boldsymbol{W}_{a_i} = \begin{cases} \mathbf{0.2}, & \boldsymbol{t} \leq \mathbf{180} \\ \mathbf{0.1}, & \mathbf{181} \leq \boldsymbol{t} \leq \mathbf{365} \\ \mathbf{0.05}\ , & \boldsymbol{t} > \mathbf{365} \end{cases} \qquad (12)$$

其中 t 为附件 3 中最近的留言时间（或当前时间）与第 i 条留言的实际留言时间之差，W_{t_i} 为第 i 条留言的时间权重值，W_{a_i}为第 i 条留言的支持度权重值，N_{a_i}为第 i 条留言的点赞数，N_{d_i}为第 i 条留言的反对数

5.2.2 结果分析：

结果分析如图 12

图 12

使用 Jaccard 系数算法对示例数据进行相似度计算的测试得到图 13

B	C	D	E	F
留言编号	留言用户	留言主题	留言时间	留言详情
360103	A0012425	A5区劳动东路魅力之城小区临街门面烧烤夜宵摊	2019/09/25 00:31:32	A5区劳动东路魅力之城小区临街夜宵摊、烧烤摊
360107	A0283523	A5区劳动东路魅力之城小区一楼的夜宵摊严重污染附近的空气	2019/07/21 10:29:35	局长：　　你好，A5区劳动东路魅力之城小区
360108	A0283523	A5区劳动东路魅力之城小区一楼的夜宵摊严重污染附近的空气，急需处理!	2019/08/01 16:20:01	局长：　　你好，A5区劳动东路魅力之城小区
360100	A324156	魅力之城小区临街门面油烟直排扰民	2019/09/05 12:29:01	魅力之城小区楼下烧烤摊、快餐店无证经营，快
360101	A324156	A5区劳动东路魅力之城小区油烟扰民	2019/07/28 12:49:17	尊敬的政府：A5区劳动东路魅力之城小区临街门
360102	A1234140	A5区劳动东路魅力之城小区底层餐馆油烟扰民	2019/09/10 06:13:27	A5区劳动东路魅力之城小区，底层有几家餐管，

图 13

使用 Jaccard 系数算法对示例数据进行相似度的分类得到图 14（使用留言主题作为比较的文本）

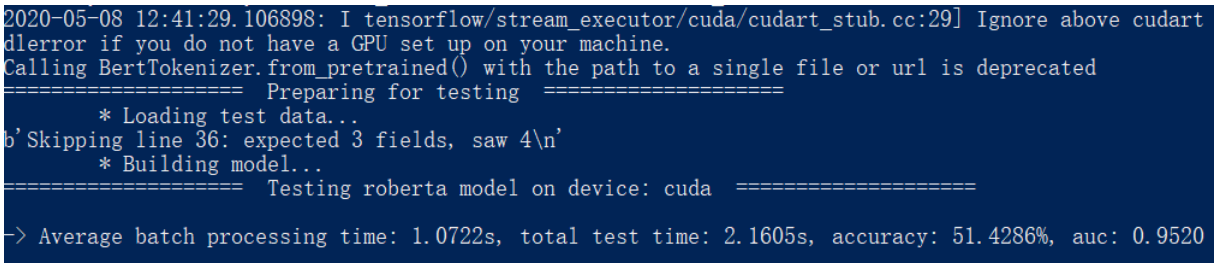


图 14

使用 RoBerta 模型对示例数据进行相似度计算的测试得到图 15

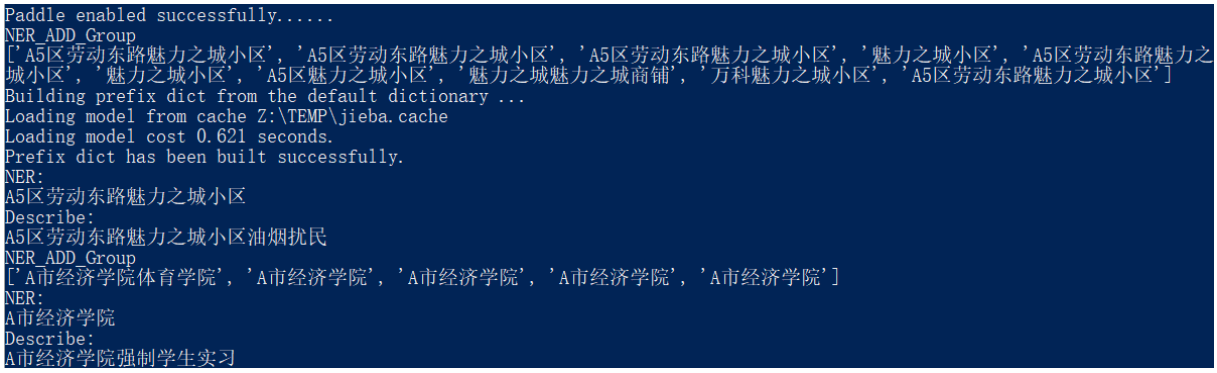


图 15

提取特定的地点或人群以及给出问题描述的程序实现结果得到图 16

A	B	C	D	E	F
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	91.8	2019/07/21 至 2019/09/25	A5区劳动东路魅力之城小区	A5区劳动东路魅力之城小区油烟扰民
2	2	26.9	2017/06/08 至 2019/11/22	A市经济学院	A市经济学院强制学生实习

图 16

使用热度指标公式计算的结果

结果分析:RoBERTa/BERT 模型受语料影响非常大，如果没有适合的语料来训练，效果并不算太理想；Jaccard 系数算法与 TF-IDF 算法的效果则相对好点，相比之下，Jaccard 系数算法速度更快，程序的复杂度更低；提取特定的地点或人群时偶尔出现提取不完整的情况，比如 A7 县被识别成 A7，这时需要我们特地加入相关规则去完善，热度指标公式中的第 i 条留言的支持度权重值，第 i 条留言的时间权重值，应按具体情况设定。

5.3 问题三分析方法与过程

5.3.1 模型建立与求解

我们对留言的答复意见采用满分 100 分制的评价制度，分别从答复的及时性、可解释性、相关性以及完整性四个方面对答复意见进行量化评价，建立衡量相关部门专业水平和回复态度等方面的综合数学模型。该模型有助于对相关部门答复意见进行评价，方便部门查缺补漏提高留言回复质量，利于群众享受优质的网络留言服务。[5]

（1） 留言回复的及时性分值占总分的 20%，也即 20 分。

对第 i 条留言回复的及时性进行评价 ，收到留言后 3 天内对留言给予回复，评 20 分；收到留言后第 4 天到第 10 天内对留言给予回复，评 15 分；收到留言后第 11 天到第 20 天内对留言给予回复，评 10 分；收到留言的第 21 天后对留言给予回复，评 5 分。

$$\mathbf{Score\_1i} = \begin{cases} \mathbf{20}, & |T_i' - T_i| < \mathbf{4} \\ \mathbf{15}, & \mathbf{4} \le |T_i' - T_i| \le |\mathbf{10} \\ \mathbf{10}, & \mathbf{11} \le |T_i' - T_i| \le \mathbf{20} \\ \mathbf{5}, & \mathbf{21} \le |T_i' - T_i| \end{cases} \qquad (13)$$

其中T_i为收到留言时间，T_i'为回复留言时间。

（2）留言回复的内容充实性分值占总分的 20%，也即 20 分。

回复的有效文本是指不含一些系统自动回复用语、标点符号、空格以及一些特殊符号等的文本。

留言回复内容的详细程度与回复的有效文本长度有直接的关系。简短内容的回复信息量一般不够，评分应该较低;同时，较长文本的回复评分不应该过高。因此，可以考虑使用对数函数来量化回复的文本长度与评分的关系，建立“回复内容是否充实”的评价项Score_-2i，其中Li为针对第i个问题回复的有效文本长度。

有效文本长度小于25的,评5分;有效文本长度处于25到50之间的，评10分；有效文本长度处于51到100之间的，评15分；有效文本长度处于101到150之间的，评18分；有效文本长度大于25的，评20分.

$$\mathbf{Score\_2i} = \begin{cases} \mathbf{5}, & \boldsymbol{L_i < 25} \\ \mathbf{10}, & \mathbf{25} \leq \boldsymbol{L_i} \leq \mathbf{50} \\ \mathbf{15}, & \mathbf{51} \leq \boldsymbol{L_i} \leq \mathbf{100} \\ \mathbf{18}, & \mathbf{101} \leq \boldsymbol{L_i} \leq \mathbf{150} \\ \mathbf{20}, & \boldsymbol{L_i > 150} \end{cases} \qquad (14)$$

（3)留言回复的相关性和完整性分值各占总分的30%,也即一共占了60分。

设第i条留言描述为Q_c，长度D_{1i}；第i条的答复描述为$\mathrm{A_i}$，长度D_{2i}，对应留言回复的相关性（频率）为$\mathrm{F_i}$，完整性（匹配个数）为$\mathrm{R_c}$。2字以上的词在A_i中出现的个数为完整性

$$\boldsymbol{F_i = \sum_k^{L_k} f_k \ , \ f_k \in C_f} \qquad (15)$$

两个字以上的句式短语的集合：

$$C_p = \{p_1, \dots, p_k, p_{lk}\}\ 1 \le k \le L_k \quad (16)$$

$$C_f = \{f_1, \dots, f_k, f_{lk}\}$$

其中f_k为p_k在A_i中出现的次数。

两个字以上的词：

$$D_k = Q_c(k_1 : k_2), L = k_2 - k_1 \ge 2 \quad (17)$$

$$C_i = \{k_i \dots k_i\} \in \{1 \dots D_i\}$$

$$R_i = L_k$$

留言回复的相关性评分公式：

$$\mathbf{Score3\_1i} = \begin{cases} 10, & C_i < 15 \\ 20, & 16 \le C_i \le 50 \\ 30, & C_i \ge 51 \end{cases} \quad (18)$$

留言回复的完整性评分公式：

$$\mathbf{Score3\_2i} = \begin{cases} 10, & R_i < 200 \\ 20, & 201 \le R_i \le 500 \\ 30, & R_i \ge 501 \end{cases} \quad (19)$$

留言回复的相关性与完整性评价总分值：

$$\mathbf{Score\_3i = Score3\_1i + Score3\_2i} \quad (20)$$

（4）留言回复的评价总分值：

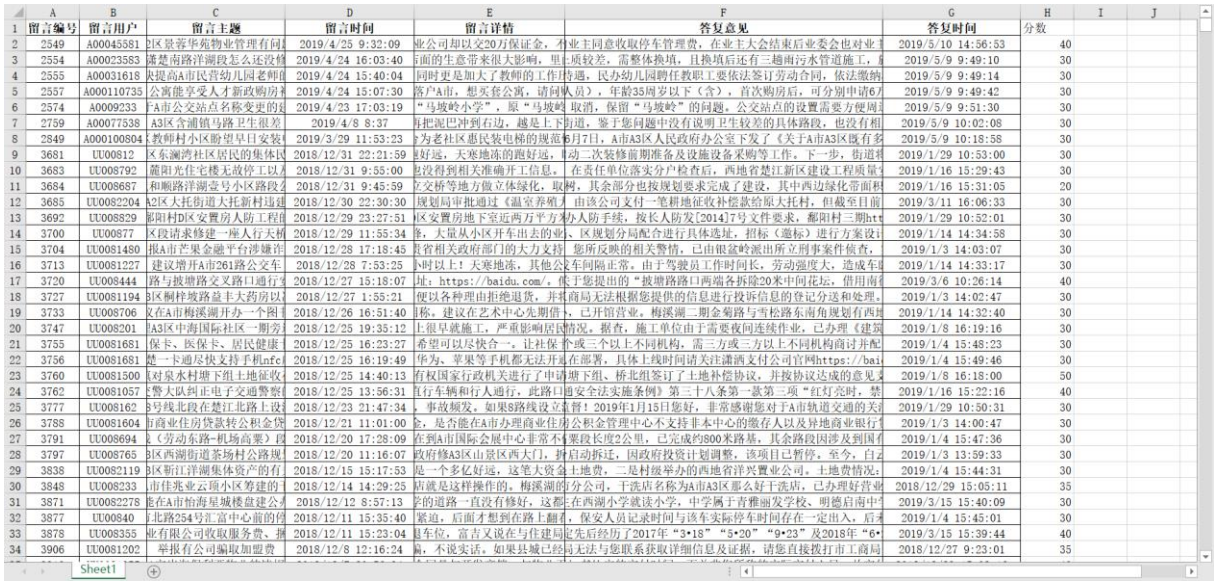
$$Score\_i = Score\_1i + Score\_2i + Score3\_2 \qquad (21)$$

完整过程如图 17

图 17

5.3.2 结果分析：

通过 matlab 运行后得到了每条留言回复的对应分数：一共 2816 条留言回复，其中 80 分及 80 分以上的有两条，处于 70 分到 79 之间的有 10 条，处于 60 分到 69 分之间的有 34 条，其余的 2760 条留言回复都是不及格的，小于 60 分。如图 18



	A	B	C	D	E	F	G	H	I	J
1	留言编号	留言用户	留言主题	留言时间	留言详情	答复意见	答复时间	分数		
2	2549	A00045581	B区景蓉华苑物业管理有问	2019/4/25 9:32:09	业公司却以交20万保证金，	业主同意收取停车管理费，在业主大会结束后业委会也对业主	2019/5/10 14:56:53	40		
3	2554	A00023583	潇楚南路洋湖段怎么还没修	2019/4/24 16:03:40	面的生意带来很大影响，里	质较差，需整体换填，且换填后还有三趟南污水管道施工，	2019/5/9 9:49:10	30		
4	2555	A00031618	关提高A市民营幼儿园老师	2019/4/24 15:40:04	同时更是加大了教师的工作	待遇，民办幼儿园聘任教职工要依法签订劳动合同，依法缴纳	2019/5/9 9:49:14	30		
5	2557	A000110735	公寓能享受人才新政购房补	2019/4/24 15:07:30	落户A市，想买套公寓，请问	人员），年龄35周岁以下（含），首次购房后，可分别申请6万	2019/5/9 9:49:42	30		
6	2574	A0009233	于A市公交站点名称变更的建	2019/4/23 17:03:19	“马坡岭小学”，原“马坡岭	取消，保留“马坡岭”的问题。公交站点的设置需要方便周	2019/5/9 9:51:30	30		
7	2759	A00077538	A3区含浦镇马路卫生很差	2019/4/8 8:37	每把泥巴冲到石边，越是上下	街道，鉴于您问题中没有说明卫生较差的具体路段，也没有相	2019/5/9 10:02:08	30		
8	2849	A000100804	教师村小区盼望早日安装电	2019/3/29 11:53:23	为老社区惠民装电梯的规范	6月7日，A市A3区人民政府办公室下发了《关于A市A3区既有多	2019/5/9 10:18:58	30		
9	3681	UU00812	区东澜湾社区居民的集体民	2018/12/31 22:21:59	好远，天寒地冻的跑好远，	动二次装修前期准备及设施设备采购等工作。下一步，街道将	2019/1/29 10:53:00	30		
10	3683	UU008792	麓阳光住宅楼无故停工以	2018/12/31 9:55:00	没得到相关准确开工信息，	在责任单位落实分户检查后，西地省建江新区建设工程质量	2019/1/16 15:29:43	30		
11	3684	UU008687	和顺路洋湖壹号小区路段公	2018/12/31 9:45:59	交桥等地方做立体绿化，取	树，其余部分也按规划要求完成了建设，其中西边绿化带面积	2019/1/16 15:31:05	20		
12	3685	UU0082204	A2区大托街道大托新村违建	2018/12/30 22:30:30	规划局审批通过《温室养殖	由该公司支付一笔耕地征收补偿款给原大托村，但截至目前	2019/3/11 16:06:33	30		
13	3692	UU008829	鄱阳村D区安置房入防工程	2018/12/29 23:27:51	区安置房地下室近两万平方	办人防手续，按长人防发[2014]7号文件要求，鄱阳村三期htt	2019/1/29 10:52:01	30		
14	3700	UU00877	区段请求修建一座人行天桥	2018/12/29 11:55:34	多，大量从小区开车出去的业	、区规划分局配合进行具体选址，招标（邀标）进行方案设计	2019/1/14 14:34:58	30		
15	3704	UU0081480	报A市芒果金融平台涉嫌诈	2018/12/28 17:18:45	省相关政府部门的大力支持	您所反映的相关警情，已由银盆岭派出所立刑事案件侦查，	2019/1/3 14:03:07	30		
16	3713	UU0081227	建议增开A市261路公交车	2018/12/28 7:53:25	小时以上！天寒地冻，其他公	车问隔正常。由于驾驶员工作时间长，劳动强度大，造成车	2019/1/14 14:33:17	30		
17	3720	UU008444	路与拔增路交叉路口通行	2018/12/27 15:18:07	址：https://baidu.com/，	于您提出的“拔增路路口两端各拆除20米中间花坛，借用南	2019/3/6 10:26:14	40		
18	3727	UU0081194	B区桐梓坡路益丰大药房以	2018/12/27 1:55:21	便以各种理由拒绝退货，并	商局无法根据您提供的信息进行投诉信息的登记分送和处理。	2019/1/3 14:02:47	30		
19	3733	UU008706	区在A市梅溪湖开办一个图	2018/12/26 16:51:40	称，建议在艺术中心先期借	、已开馆营业。梅溪湖二期金菊路与雪松路东南角规划有西	2019/1/14 14:32:40	30		
20	3747	UU008201	A3区中海国际社区一期旁	2018/12/25 19:35:12	上很早就施工，严重影响居民	情况。据查，施工单位由于需要夜间连续作业，已办理《建筑	2019/1/8 16:19:16	30		
21	3755	UU0081681	保卡、医保卡、居民健康	2018/12/25 16:23:27	希望可以尽快合一，让社保	个或三个以上不同机构，需三方或三方以上不同机构商讨并配	2019/1/4 15:48:23	30		
22	3756	UU0081681	楚一卡通尽快支持手机nfc	2018/12/25 16:19:49	华为、苹果等手机都无法开通	在部署，具体上线时间请关注潇洒支付公司官网https://bai	2019/1/4 15:49:46	30		
23	3760	UU0081500	对泉水村塘下组土地征收	2018/12/25 14:40:13	有权国家行政机关进行了申请	塘下组、桥北组签订了土地补偿协议，并按协议达成的意见	2019/1/8 16:18:00	50		
24	3762	UU0081057	警大队纠正电子交通警察	2018/12/25 13:56:31	行车辆和行人通行，此路口	道安全法实施条例》第三十八条第一款第三项“红灯亮时，禁	2019/1/16 15:22:16	40		
25	3777	UU008162	3号线北段在楚江北路上设	2018/12/23 21:47:34	事故频发，如果8路线设立	督！2019年1月15日您好，非常感谢您对于A市轨道交通的关	2019/1/29 10:50:31	30		
26	3788	UU0081604	行商业住房贷款转公积金贷	2018/12/21 11:01:00	金，是否能在A市办理商业住房	公积金管理中心不支持非本中心的缴存人以及异地商业银行	2019/1/3 14:00:47	30		
27	3791	UU008694	区（劳动东路-机场高架）段	2018/12/20 17:28:09	在到A市国际会展中心非常不	架段长度2公里，已完成约800米路基，其余路段因涉及到国	2019/1/4 15:47:36	30		
28	3797	UU008765	B区西湖街道茶场村公路规	2018/12/20 11:16:07	政府修A3区山景区西大门，拆	启动拆迁，因政府投资计划调整，该项目已暂停，至今，白	2019/1/3 13:59:33	30		
29	3838	UU0082119	B区新江洋湖集体资产的有	2018/12/15 15:17:53	是一个多亿好运，这笔大资金	土地费，二是村级举办的西地省洋兴置业公司。土地费情况：	2019/1/4 15:44:31	30		
30	3848	UU008233	市佳兆业云项小区筹建的	2018/12/14 14:29:25	店就是这样操作的。梅溪湖的	分公司，于洗店名称为A市A3区那么好于洗店，已办理好营业	2018/12/29 15:05:11	35		
31	3871	UU0082278	在A市恰海星城楼盘建公寓	2018/12/12 8:57:13	的道路一直没有修好，这都	在西湖小学就读小学，中学属于青雅湖发学校、明德启南中	2019/3/15 15:40:09	30		
32	3877	UU00840	北路254号汇富中心前的停	2018/12/11 15:35:40	紧迫，后面才想到在路上翻看	保安人员记录时间与该车实际停车时间存在一定出入，后	2019/1/4 15:45:01	30		
33	3878	UU008355	业有限公司收取服务费、摊	2018/12/11 15:23:04	县车位，富吉又说在与住建局	定先后经历了2017年“3•18”“5•20”“9•23”及2018年“6•	2019/3/15 15:39:44	40		
34	3906	UU0081202	举报有公司骗取加盟费	2018/12/8 12:16:24	骗，不说实话，如果县城已经	无法与您联系获取详细信息及证据，请您直接拨打市工商局	2018/12/27 9:23:01	35		

Sheet1

图 18

6、模型评价

6.1 问题一的模型评价

问题一建立的是词袋模型和 SVM 模型，该模型的建立具有较高的合理性。本文所建模型都是立足于题目所给的相关信息，再深入分析数据的基础上建立起来的，因此模型求解出的结果较为准确。核支持向量机是非常强大的模型，在各种数据集上的表现都很好。SVM 允许决策边界很复杂，即使数据只有几个特征。它在低维数据和高维数据（即很少特征和很多特征）上的表现都很好，但对样本个数的缩放表现不好。在有多达 10000 个样本的数据上运行 SVM 可能表现良好，但如果数据量达到 100000 甚至更大，在运行时间和内存使用方面可能会面临挑战。SVM 的另一个缺点是，预处理数据和调参都需要非常小心。

6.2 问题二的模型评价

对结果分析可知，RoBERTa/BERT 模型受语料影响非常大，如果没有适合的语料来训练，效果并不算太理想，所以我们暂不采用该模型；Jaccard 系数算法与 TF-IDF 算法的效果则相对好点，但是需要依据实际情况手动调整判断是否相似的标准值，相比之下，Jaccard 系数算法速度更快，程序的复杂度更低，所以我们首选 Jaccard 系数算法；提取特定的地点或人群时偶尔出现提取不完整的情况，比如 A7 县被识别成 A7，这时需要我们特地加入相关规则去完善，热度指标公式中的第 i 条留言的支持度权重值，第 i 条留言的时间权重值，应按具体情况设定，当留言数量非常大的时候，就需要更大的区分度，以便更好的区分热点问题。

6.3 问题三的模型评价

问题三是对留言的答复意见采用满分 100 分制的评价制度，分别从答复的及时性、可解释性、相关性以及完整性四个方面对答复意见进行量化评价，它们分别占了总分的 20%、20%、30%和 30%，以建立衡量相关部门专业水平和回复态度等方面的综合数学模型，最后得到了各条留言回复的量化评价分数。最终得到的结果最高分是 80 分，大多数是不合格的。这样的评价参数有待调整，具体实施时根据不同的实际背景对模型进行修改，在参数的

取定上也可以随时变化，如按不同的要求，结合具体的分析，对数据进行合理的统计处理，再根据模型的核心方法，就能很比较好地解决问题。

7、参考文献

[1]陈玲. 基于支持向量机的多类文本分类研究[D].重庆大学,2010.

[2]成艳洁. 基于 SVM 的多类文本分类算法及其应用研究[D].西安理工大学,2009.

[3]陈平平,耿笑冉,邹敏,谭定英.基于机器学习的文本情感倾向性分析[J].计算机与现代化,2020(03):77-81+92. （svm 图）

[4]徐磊,赵光宙,顾弘.成对耦合分类器的多球体预处理方法[J].浙江大学学报(工学版),2010,44(02):237-242.（二

[5]杨开平,李明奇,覃思义.基于网络回复的律师评价方法[J].计算机科学,2018,45(09):237-242.

附录

问题一

初始化词向量：

```
clear all;

clc


%%  获取原始数据

filename = 'C:\Users\ASUS\Downloads\teddy\附件2.xlsx';


data_r = readtable(filename,'TextType','string');


head(data_r); % 获取data_r前八行

% b=strrep(a,'is','')

data_c1=erasePunctuation(data_r.x_____4);% 删除标点符号

%

d1=data_c1(1:101,:);

d2=data_c1(102:134,:);  % 分段处理

d3=data_c1(135:189,:);

d4=data_c1(190:285,:);

d5=data_c1(286:389,:);

d6=data_c1(390:437,:);

d7=data_c1(438:495,:);
```

```
%%  数据清洗
for k_d=1:7  % 遍历分段的数据
  cl_p={};
  cl_pt={};  % 两个空数组
  k_in=1;  % ??
   p_t=eval(['d' num2str((k_d))]);  % 记录里的数字转换为字符数组
for i=1:size(eval(['d' num2str(k_d)]),1)  % 遍历转换后的数组
    del_fonts=['的了呢如何把应该于是至 我你他她阿啊1234567890一二三四五六七八九十零ABCDEFGHIJKLMNOPQRSTUVWXYZ'...
        '从来东西北南难大小上下左右中点职位无关省市县不为在们和个这出那里前后到多少以能也主可广么作做就去子时说发用者定事好情没动本天对过'
char(160)];% without 定事  c=17 停用字表
    for del_i=1:length(del_fonts)

    p_t(i)=strrep( p_t(i),del_fonts(del_i),'');  % 查找并将记录里符合停用字的字符替换为空字符串

    end
     for ip=1:size(p_t{i},2)  %  遍历p_t字符串列表行数
% cl_p{i,ip}=p_t{i}(ip);
cl_p{k_in}=p_t{i}(ip);  % 将每句话拆散成单独一个字符，一个字一格保存在1行n列新数组cl_p中
k_in=k_in+1;
```

```
    end
%   cl_pt{i}=cl_p;
end

eval(['d' num2str((k_d)) '=cl_p;']);
end
%%

%%  制表
centre_font_total=[];
inc_class=[];
for  k_d=1:7
x = categorical(eval(['d' num2str(k_d)]));
h = histogram(x);  %  生成图表，总共7个
%%  删除重复字
classCounts = h.BinCounts;  %  每个字的出现次数，作为y轴
classNames = h.Categories;  %  所有字，作为图表的x轴

c_num=23;%  选择中心词数目
cishu=sort(classCounts);  %  按出现次数由小到大排序
l_ind=min(cishu(end-c_num+1:end));%  记录前23次数多中的最小值

fdsf=classCounts>=l_ind;  % 筛选符合该出现次数字的索引（位置）
centre_font=classNames(classCounts>=l_ind); %  找出并储存字符
```

```
i_del=[];
% sum(fdsf)
if k_d>1 %  第二次循环开始进入此代码块
    total_cc=cell2mat(centre_font_total);  %  将单元格转为数组
    for c_ii=1:length(centre_font)
        te_cn=findstr(total_cc,centre_font{c_ii}); %  储存重复出现的字
的索引
if isempty(te_cn)==0
   i_del=[i_del c_ii];  %  记录要删除字的索引
end
    end
   centre_font(i_del)=[];  %  删除重复字
end
inc_class=[inc_class length(centre_font)]; % 合并七次循环每次中心词的
长度

centre_font_total=[centre_font_total centre_font];  % 合并所有筛出的字


end

ABS=min(inc_class)
min(inc_class) %% =8
tt_to=cumsum(inc_class)+1;  %  每段筛出的字的个数加一
```

```
dat_start=[1 tt_to(1:end-1)];  %  记录合并7段分段后，每段字数开头的索引
dat_end=[1 tt_to(1:end-1)]+min(inc_class)-1;  %  尾部的索引
dat_ob=[dat_start(1):dat_end(1),dat_start(2):dat_end(2)
dat_start(3):dat_end(3) dat_start(4):dat_end(4) ...
    dat_start(5):dat_end(5)  dat_start(6):dat_end(6)
dat_start(7):dat_end(7)];
%  dat_ob为合并7段的总关键字表
centre_font_total=centre_font_total(dat_ob)

%%  编码
ind_1=1:101;
ind_2=102:134;
ind_3=135:189;
ind_4=190:285;
ind_5=286:389;
ind_6=390:437;
ind_7=438:495;

d1_o=data_c1(1:101,:);
d2_o=data_c1(102:134,:);
d3_o=data_c1(135:189,:);
d4_o=data_c1(190:285,:);
```

```
d5_o=data_c1(286:389,:);

d6_o=data_c1(390:437,:);

d7_o=data_c1(438:495,:);

%  分类

label_1=ones(size(d1_o,1),1);  %  建立全1数组

label_2=2*ones(size(d2_o,1),1);

label_3=3*ones(size(d3_o,1),1);

label_4=4*ones(size(d4_o,1),1);

label_5=5*ones(size(d5_o,1),1);

label_6=6*ones(size(d6_o,1),1);

label_7=7*ones(size(d7_o,1),1);

%%

vector_n=[];

for k_d=1:7

%  注意上面七段最后合并的高频字一共56个

for i_st=1:size(eval(['d' num2str(k_d) '_o']),1)

% i_st=2;

    cc_vector=zeros(1,length(centre_font_total)); %  使用全零数组

 for i_c=1:length(centre_font_total)

     %  对每一条留言，查找高频字表的字的出现次数

t_r=strfind(eval(['d' num2str(k_d)
```

```
'_o(i_st)']),centre_font_total(i_c));
    if isempty(t_r)==1  %  没有匹配的跳回127行，重复，直到读完所有记
录
        continue;
    end
    cc_vector(i_c)=cc_vector(i_c)+length(t_r);  %  储存出现次数到零
数组中

 end

 vector_n=[vector_n;cc_vector/sum(cc_vector)];  %  取平均数

end
end
%%
%  对每一部分的分类
n_sam1=round((length(ind_1)-1)*rand(25,1))+1;
n_sam2=round((length(ind_2)-1)*rand(25,1))+1;
n_sam3=round((length(ind_3)-1)*rand(25,1))+1;
n_sam4=round((length(ind_4)-1)*rand(25,1))+1;
n_sam5=round((length(ind_5)-1)*rand(25,1))+1;
n_sam6=round((length(ind_6)-1)*rand(25,1))+1;
n_sam7=round((length(ind_7)-1)*rand(25,1))+1;
```

```
%  对留言内容分类
Train_x=[vector_n(ind_1(n_sam1),:);
vector_n(ind_2(n_sam2),:);vector_n(ind_3(n_sam3),:);vector_n(ind_4(n_
sam4),:);...

vector_n(ind_5(n_sam5),:);vector_n(ind_6(n_sam6),:);vector_n(ind_7(n_
sam7),:)];
Train_label=[label_1(n_sam1);label_2(n_sam2);label_3(n_sam3);label_4(
n_sam4);label_5(n_sam5);label_6(n_sam6);label_7(n_sam7) ];
All_x=vector_n;
All_label=[label_1;label_2;label_3;label_4;label_5;label_6;label_7 ];

%%
save('d_t','All_x','All_label');
disp('处理完毕。');
sum(isnan(All_x))
% %%
% data_c2=tokenizedDocument(data_c1);
% data_c2=addPartOfSpeechDetails(data_c2);
% %%

% data_c1=removeStopWords(data_c);
```

SVM 分类：

```
clear all;

clc

%%  获取原始数据

filename = 'C:\Users\ASUS\Downloads\teddy\附件2.xlsx';

data_r = readtable(filename,'TextType','string');

head(data_r); % 获取data_r前八行

% b=strrep(a,'is','')

data_c1=erasePunctuation(data_r.x_____4);% 删除标点符号

%

d1=data_c1(1:101,:);

d2=data_c1(102:134,:);  % 分段处理

d3=data_c1(135:189,:);

d4=data_c1(190:285,:);

d5=data_c1(286:389,:);

d6=data_c1(390:437,:);

d7=data_c1(438:495,:);

%%  数据清洗

for k_d=1:7  % 遍历分段的数据

  c1_p={};
```

```
  cl_pt={};  % 两个空数组
  k_in=1;  % ??
   p_t=eval(['d' num2str((k_d))]);  % 记录里的数字转换为字符数组
for i=1:size(eval(['d' num2str(k_d)]),1)  % 遍历转换后的数组
    del_fonts=['的了呢如何把应该于是至 我你他她阿啊1234567890一二三四五六七八九十零ABCDEFGHIJKLMNOPQRSTUVWXYZ'...
        '从来东西北南难大小上下左右中点职位无关省市县不为在们和个这出那里前后到多少以能也主可广么作做就去子时说发用者定事好情没动本天对过'
char(160)];% without 定事  c=17 停用字表
    for del_i=1:length(del_fonts)

    p_t(i)=strrep( p_t(i),del_fonts(del_i),'');  % 查找并将记录里符合停用字的字符替换为空字符串

    end
     for ip=1:size(p_t{i},2)  %  遍历p_t字符串列表行数
% cl_p{i,ip}=p_t{i}(ip);
cl_p{k_in}=p_t{i}(ip);  % 将每句话拆散成单独一个字符，一个字一格保存在1行n列新数组cl_p中
k_in=k_in+1;
     end
%   cl_pt{i}=cl_p;
end
```

```
eval(['d' num2str((k_d)) '=c1_p;']);
end
%%

%%  制表
centre_font_total=[];
inc_class=[];
for  k_d=1:7
x = categorical(eval(['d' num2str(k_d)]));
h = histogram(x);  %  生成图表，总共7个
%%  删除重复字
classCounts = h.BinCounts;  %  每个字的出现次数，作为y轴
classNames = h.Categories;  %  所有字，作为图表的x轴

c_num=23;%  选择中心词数目
cishu=sort(classCounts);  %  按出现次数由小到大排序
l_ind=min(cishu(end-c_num+1:end));%  记录前23次数多中的最小值

fdsf=classCounts>=l_ind;  % 筛选符合该出现次数字的索引（位置）
centre_font=classNames(classCounts>=l_ind); %  找出并储存字符
i_del=[];
% sum(fdsf)
if k_d>1 %  第二次循环开始进入此代码块
```

```
    total_cc=cell2mat(centre_font_total);  %  将单元格转为数组
    for c_ii=1:length(centre_font)
        te_cn=findstr(total_cc,centre_font{c_ii}); %  储存重复出现的字的索引
if isempty(te_cn)==0
   i_del=[i_del c_ii];  %  记录要删除字的索引
end
    end
   centre_font(i_del)=[];  %  删除重复字
end
inc_class=[inc_class length(centre_font)]; % 合并七次循环每次中心词的长度

centre_font_total=[centre_font_total centre_font];  % 合并所有筛出的字


end

ABS=min(inc_class)
min(inc_class) %% =8
tt_to=cumsum(inc_class)+1;  %  每段筛出的字的个数加一
dat_start=[1 tt_to(1:end-1)];  %  记录合并7段分段后，每段字数开头的索引
dat_end=[1 tt_to(1:end-1)]+min(inc_class)-1;  %  尾部的索引
```

```
dat_ob=[dat_start(1):dat_end(1),dat_start(2):dat_end(2)
dat_start(3):dat_end(3) dat_start(4):dat_end(4) ...
    dat_start(5):dat_end(5)  dat_start(6):dat_end(6)
dat_start(7):dat_end(7)];
%  dat_ob为合并7段的总关键字表
centre_font_total=centre_font_total(dat_ob)

%%  编码
ind_1=1:101;
ind_2=102:134;
ind_3=135:189;
ind_4=190:285;
ind_5=286:389;
ind_6=390:437;
ind_7=438:495;

d1_o=data_c1(1:101,:);
d2_o=data_c1(102:134,:);
d3_o=data_c1(135:189,:);
d4_o=data_c1(190:285,:);
d5_o=data_c1(286:389,:);
d6_o=data_c1(390:437,:);
d7_o=data_c1(438:495,:);
```

```
%  分类
label_1=ones(size(d1_o,1),1);  %  建立全1数组
label_2=2*ones(size(d2_o,1),1);
label_3=3*ones(size(d3_o,1),1);
label_4=4*ones(size(d4_o,1),1);
label_5=5*ones(size(d5_o,1),1);
label_6=6*ones(size(d6_o,1),1);
label_7=7*ones(size(d7_o,1),1);

%%
vector_n=[];
for k_d=1:7
%  注意上面七段最后合并的高频字一共56个
for i_st=1:size(eval(['d' num2str(k_d) '_o']),1)
% i_st=2;
    cc_vector=zeros(1,length(centre_font_total)); %  使用全零数组
 for i_c=1:length(centre_font_total)
     %  对每一条留言，查找高频字表的字的出现次数
t_r=strfind(eval(['d' num2str(k_d)
'_o(i_st)']),centre_font_total(i_c));
     if isempty(t_r)==1  %  没有匹配的跳回127行，重复，直到读完所有记
录
```

```
        continue;
    end
    cc_vector(i_c)=cc_vector(i_c)+length(t_r);  %  储存出现次数到零
数组中

 end

 vector_n=[vector_n;cc_vector/sum(cc_vector)];  %  取平均数

end
end
%%
%  对每一部分的分类
n_sam1=round((length(ind_1)-1)*rand(25,1))+1;
n_sam2=round((length(ind_2)-1)*rand(25,1))+1;
n_sam3=round((length(ind_3)-1)*rand(25,1))+1;
n_sam4=round((length(ind_4)-1)*rand(25,1))+1;
n_sam5=round((length(ind_5)-1)*rand(25,1))+1;
n_sam6=round((length(ind_6)-1)*rand(25,1))+1;
n_sam7=round((length(ind_7)-1)*rand(25,1))+1;

%  对留言内容分类
Train_x=[vector_n(ind_1(n_sam1),:);
```

```
vector_n(ind_2(n_sam2),:);vector_n(ind_3(n_sam3),:);vector_n(ind_4(n_
sam4),:);...

vector_n(ind_5(n_sam5),:);vector_n(ind_6(n_sam6),:);vector_n(ind_7(n_
sam7),:)];
Train_label=[label_1(n_sam1);label_2(n_sam2);label_3(n_sam3);label_4(
n_sam4);label_5(n_sam5);label_6(n_sam6);label_7(n_sam7) ];
All_x=vector_n;
All_label=[label_1;label_2;label_3;label_4;label_5;label_6;label_7 ];

%%
save('d_t','All_x','All_label');
disp('处理完毕。');
sum(isnan(All_x))
% %%
% data_c2=tokenizedDocument(data_c1);
% data_c2=addPartOfSpeechDetails(data_c2);
% %%

% data_c1=removeStopWords(data_c);
```

问题二

问题三

```
clear all;

close all;

clc;

filename='C:\Users\ASUS\Downloads\teddy\C题全部数据\附件4.xlsx';

data_r=readtable(filename,'TextType','string');

head(data_r);

%%  时间评分 %20

time_c1=data_r.x_____3;

time_c2=data_r.x_____6;

ask_c=data_r.x_____4;

answ_c=data_r.x_____5;

%%

score_c=zeros(size(data_r,1),1);

%%

score_1=20;

time_c1=fix(time2_dig(time_c1)*0.000001);

time_c2=fix(time2_dig(time_c2)*0.000001);

%%
```

```
score_c(abs(time_c2-time_c1)<4)=20;
score_c(abs(time_c2-time_c1)>=4&abs(time_c2-time_c1)<=10)=15;
%%
score_c(abs(time_c2-time_c1)>=11&abs(time_c2-time_c1)<=20)=10;
%%
score_c(abs(time_c2-time_c1)>=21)=5;
%%  回答文字长度评分-回答态度问题（不含标点符号，空格，特殊符号除外）20%

p_t=answ_c;
score_c2=zeros(size(answ_c,1),1);
score_2=20;
%%

for i=1:size(p_t,1)
%     del_fonts=['的了呢如何把应该于是至 我你他她阿啊1234567890一二三四五六七八九十零ABCDEFGHIJKLMNOPQRSTUVWXYZ'...
%         '从来东西北南难大小上下左右中点职位无关不为在们和个这出那里前后到多少以能也可广么作做就去子时说发用者定没动对过而给年月日地省已经进行' char(160) char(10) char(12288)];% without 定事  c=17
    del_fonts=[',,″:.。“”《》…—:!!' char(160) char(10) char(12288)];%
   p_t(i)=erase( p_t(i),'您好！您的留言已收悉现将有关情况回复如下');
       p_t(i)=erase( p_t(i),'网友UU00');
```

```
        p_t(i)=erase( p_t(i),'您好！函件收悉，经调查处理，现将有关情况回复如下');
        p_t(i)=erase( p_t(i),'网友：您好！留言已获悉，现回复如下');
        p_t(i)= erase( p_t(i),'网友');
         p_t(i)=erase( p_t(i),'您好');
         p_t(i)=erase( p_t(i),'留言');
       p_t(i)=erase( p_t(i),'收悉');
      p_t(i)= erase( p_t(i),'UUOO');
      p_t(i)= erase( p_t(i),'你好');
      p_t(i)= erase( p_t(i),'回复');
       p_t(i)= erase( p_t(i),'您现将有情况');
       p_t(i)= erase( p_t(i),'回复下');
        p_t(i)= erase( p_t(i),'有关');
    for del_i=1:length(del_fonts)

    p_t(i)=strrep( p_t(i),del_fonts(del_i),'');

    end
    if length(p_t(i))<25
       score_c2(i) =5;
    end
    if length(p_t(i))>=25&&length(p_t(i))<=50
       score_c2(i) =10;
```

```
    end
     if length(p_t(i))>=51&&length(p_t(i))<=100
       score_c2(i) =15;
    end

      if length(p_t(i))>=101&&length(p_t(i))<=150
       score_c2(i) =18;
      end

      if length(p_t(i))>150
       score_c2(i) =20;
    end

end


p_t_1=p_t;
%% 回答相关性，完整性等 %60

for i=1:size(p_t,1)
    del_fonts=['的了呢如何把应该于是至 我你他她阿啊'...
        '从来难大小点职位无关不为在们和个这出那里前后中到多少以能也可广么作做就去子时说发定没对过而给地已经进行上下' char(160) char(10) char(12288)];
```

```
    for del_i=1:length(del_fonts)

    p_t(i)=strrep( p_t(i),del_fonts(del_i),'');
%
    end

 p_t(i)=erasePunctuation( p_t(i));


end
answer_p=p_t;
p_t=ask_c;
for i=1:size(p_t,1)
    del_fonts=['的了呢如何把应该于是至 我你他她阿啊'...
        '从来难大小点职位无关不为在们和个这出那里前后中到多少以能也可
广么作做就去子时说发定没对过而给地已经进行上下' char(160) char(10)
char(12288)];


    for del_i=1:length(del_fonts)

    p_t(i)=strrep( p_t(i),del_fonts(del_i),'');
%
    end
   p_t(i)=erasePunctuation( p_t(i));
```

```
end

ask_p=p_t;
%%
score_c3=zeros(size(answ_c,1),1);
indp_r_c3=zeros(size(answ_c,1),1);
indp_r_str_total=[];
score_3=60;
 indp_r_str_totaln=[];
for ia=1:length(ask_p)
 cs=0;
 indp_r_str=[];
        for inn=1:length(ask_p{ia})

                        if cs<1
                           strm =ask_p{ia}(inn);
                        else
                            strm= [strm ask_p{ia}(inn)];
                        end
        indp=strfind( answer_p{ia} ,  strm);
              if ~isempty(indp)
        cs=cs+1;
```

```
            if cs>1
                label_w=0;
               leng_r=length(indp);
                cs_r=cs;

          indp_r_str_old=string(strm);
             end

                else
                    cs=0;
                    if cs_r ~=cs&&label_w==0

                       indp_r_c3(ia) = indp_r_c3(ia)+leng_r;
                       label_w=1;
                    end

                    if strlength(indp_r_str_old)>1
&&sum((indp_r_str_old==string(indp_r_str)))==0

                     indp_r_str=[indp_r_str indp_r_str_old];
                     indp_r_str_old="";
                    end

                end
```

```
        end

        if ia==length(ask_p)&&label_w==0

             indp_r_c3(ia) = indp_r_c3(ia)+leng_r;

                    label_w=1;

        end

    indp_r_str_total=[indp_r_str_total;  { indp_r_str}];

    indp_r_str_totaln=[indp_r_str_totaln;length(indp_r_str)];

end

    %% 词匹配个数 30%，词匹配频率30%

%

score_c3_1=zeros(size(answ_c,1),1);

score_c3_2=zeros(size(answ_c,1),1);

score_c3_1(indp_r_str_totaln<15)=score_c3_1(indp_r_str_totaln<15)+10;

score_c3_1(indp_r_str_totaln>=16&indp_r_str_totaln<=50)=score_c3_1(in

dp_r_str_totaln>=16&indp_r_str_totaln<=50)+20;

score_c3_1(indp_r_str_totaln>=51)=score_c3_1(indp_r_str_totaln>=51)+3

0;
```

```
score_c3_2(indp_r_c3<200)=score_c3_2(indp_r_c3<200)+10;

score_c3_2(indp_r_c3>=201&indp_r_c3<=500)=score_c3_2(indp_r_c3>=201&i
ndp_r_c3<=500)+20;

score_c3_2(indp_r_c3>=501)=score_c3_2(indp_r_c3>=501)+30;
%%
%      sum(score_c3_2==30)
%       sum(score_c3_1==30)

score_c3=score_c3_1+score_c3_2;
score_total=score_c+score_c2+score_c3;
%%
% sum(score_total==85);
copyfile(filename,'C:\Users\ASUS\Downloads\teddy\');
%%
xlswrite('C:\Users\ASUS\Downloads\teddy\附件4.xlsx',[{'分数
'}],'H1:H1');%%
xlswrite('C:\Users\ASUS\Downloads\teddy\附件
4.xlsx',[string(score_total)],['H2:H'
num2str(length(score_total)+1)]);
disp('数据已保存.')
```