“智慧政务”中的文本挖掘应用

摘要

本文旨在设计基于自然语言处理技术的智慧政务系统，以群众问政的留言记录以及相关部分对群众留言的答复意见作为研究数据，解决了对群众留言的分类和热点问题的挖掘以及提出答复意见评价方案等问题，对政府的管理水平和施政效果的提升具有十分重大的意义。

针对问题一，首先利用深度学习中的**卷积神经网络**(CNN)算法，构建基于卷积神经网络来解决自然语言处理(NLP)问题的模型(**TextCNN**)，最后得出关于留言内容的一级标签分类模型。用建立的分类模型对附件 2 的留言内容进行训练，训练后的模型对测试样本进行测试，使用准确率、**F-score** 、**混淆矩阵**等评价指标对模型进行评价，得到**准确率为91.65%**，**F-score 为0.9131**，且得到的混淆矩阵对角线上的数远远大于其他位置的数。

针对问题二，首先建立热度指数受**点赞数、反对数、时间间隔**以及**留言数量**四个指标影响的热度评价模型：$H_A = -0.1\times T + 0.3\times N_{留言} + 0.4\times N_{赞成} - 0.2\times N_{反对}$。使用**BIRCH**算法对留言内容聚类，在聚类之前先使用**K-means**算法进行预聚类，从而将留言进行归类得到类别个数为 **180 类**；对归类后的留言使用热度指数计算模型，得出热度指数排名前 5 的问题 ID 分别为：60，2，101，141，3。最后将热度排名前 5 的热点问题使用**TextRank**算法把问题描述概括出来。

针对问题三，要给出一个质量评价方案，需要建立质量评价指数计算模型。首先要将**相关性、完整性、可解释性、时效性**四个指标分别量化为**相似度、文本长度和标点字符占比、情感度、时间间隔**，建立质量评价指数模型：质量评价指数=$0.2\times S - 0.3\times C_1 + 0.1\times \cos C_2 - 0.1\times E - 0.3\times T$。利用几种二分类分类器的准确率进行人工调试出各个指标的权重，最后计算出其质量评价指数，设置指数的均值为阈值，并给出每个样本的高低质量分类。

关键字：卷积神经网络； K-means 聚类； BIRCH 聚类； TextRank 算法

Abstract

The purpose of this paper is to design the wisdom of the government affairs system based on natural language processing technique, solve the problem on the classification of the message and hot problems such as mining and reply advice on evaluation scheme, for the government's management level and the improvement of policy effect has the extremely significant significance.

In order to solve the first problem, the convolution neural network algorithm in deep learning is used to build a model based on convolution neural network to solve the problem of natural language processing. The classification model is used to train the message content in Annex 2, and the trained model is used to test the test samples. The accuracy, F1 score, confusion matrix and other evaluation indexes are used to evaluate the model, and the accuracy is 91.65%, F1 score is 0.9131.

For question 2, first set up the heat index by thumb up number, number, time interval, and message number heat evaluation model on the impact of four indicators:. Using the algorithm for the message content clustering, use before clustering algorithm for clustering, to get the number of categories of 180 kinds of messages; To categorize the message after using the heat index calculation model, it is concluded that the heat index in the top five question ID respectively: 60,2,101,141,3.

In view of the problem of three, to give a quality evaluation scheme, need to establish a quality evaluation index calculation model. Integrity first, interpretability and timeliness of the four index quantification of the similarity, the length of the text and punctuation characters of ratio, degree of emotional, time interval, the quality evaluation index model is established. Use several binary classification accuracy of classifier artificial debugging the weight of each index, set up the index of average as the threshold value, and high and low quality of each sample classification is given.

Key word: Convolutional neural network;k-means clustering;BIRCH clustering; TextRank algorithm

目录

1 问题重述

1.1 问题背景

这些年来，随着微信、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、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 要解决的问题

（1）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。

（2）请根据附件 3 将某一时段内放映特定地点或特定人群问题的留言内容进行归类，定义合理的热度评价指标，给出评价结果。

（3）针对附件 4 相关部门对留言的答复意见，从答案的相关性、完整性、可解释性等角度对答复意见给出一套评价方案。

2 问题一：群众留言分类

2.1 问题分析

题目要求建立关于留言内容的一级标签分类模型，因此，首先需要确立留言内容，根据附件 2，已知留言主题与留言详情，故可以通过合并留言主题和详情得出留言内容。接着，对留言内容与一级分类，建立文本分类模型。要测试分类结果，得先对模型进行训练，通过对数据进行划分出训练集、测试集和验证集，来训练模型，测试结果。要判断模型是否适合数据，还需要通过验证集对模型进行评估，得出适合留言内容的一级标签模型。

2.2 数据预处理

文本预处理处于整个文本处理过程中的开始部分，其对很多常规异常数据和数值的前期和预见性处理有着很大的意义，也因此直接或间接影响着后面的每一步的结果。由于留言数据的格式不够规范，无法用于分类研究，所以对文本的预

处理十分重要。数据预处理可以在算法开始前避免很多非逻辑上的意外情况，比如去除噪声数据，异常值、缺失值处理等。本文的数据预处理主要包括噪声数据去除、分词、去重和对停用词过滤等。

2.2.1 噪声数据去除

由于留言数据可能不仅包含了能够标识文本信息和语义的字符，往往还会带有较多与文本重要信息和内容关系甚少的附和结构和部分，例如超链接、表情符号、描述符等数据。这些数据对构成完整的文本或许是不可缺少的，但这些词一般情况下没有明确的语义信息，只是更加深刻表达了文本的语义信息，却会导致留言数据处理过程的时间复杂度和运算规模的增加，降低文本处理方法的准确率和可靠性。因此，在留言文本数据正式处理之前要对这些异常信息和噪音数据进行清洗和整理。

2.2.2 分词

分词是中文文本处理的一个基础步骤，也是进行文本表示的前提。分词就是在一定的中文语法语义规则下，将原本连续表达的一段中文文本分割筛选成单个的字或词的过程，在这过程中会忽略掉很多与原文本语义表达相关性不大的成分，而保留较为关键，核心的字和词。

用于分词的算法是基于正向最大匹配策略，正向最大匹配策略是从文本初始位置开始分割，设置分割单词长度，将其与词典中给定的单词进行比较，删除最长的匹配单词，再统计结果中选择频率最高的词。本文采用的是基于结巴（精确模式）库的统计方法对文本数据进行分词的。

2.2.3 去重

由于留言数据可能会有一些用户为了表达他们的意愿，采取一些重复字符来家中语气，或者错写等情况。因此，需要对重复的样本数据去除，方便之后对文本数据的处理。

2.2.4 停用词过滤

停用词是指在文本中出现频率较大，次数较多，但是从语义理解和语义表达的角度来说，其对文本信息的表达或分类的贡献度很小，主要包括了公共停用词和专业停用词。本文使用的是基于中文停用词表stopwords.txt 。

2.3 模型建立

2.3.1 TextCNN 模型算法简介

文本分类是根据给定文本数据，预测每个测试文本对应的类别。每个文本由文本之间的局部特征和全局特征相互作用决定。针对文本内部相互作用的关系，采用由多个尺寸的卷积核组成的卷积神经网络进行实验，提取文本特征项之间的复杂关系。

TextCNN 模型是由 Yoon Kim 提出的使用卷积神经网络来处理 NLP (自然语言处理) 问题的模型。相比较 NLP 中传统的 RNN 或者 LSTM 模型，CNN 能更加高效的提取重要特征，这些特征在分类中占据着重要位置其主要思想是将不同长度的短文作为矩阵输入，使用多个不同尺寸的卷积核去提取句子中的关键信息，并用于最终的分类。

2.3.2 TextCNN 算法网络架构

卷积神经网络文本分类模型[1]由卷积层、池化层、全连接层、softmax 分类函数组成。在卷积层之前，首先得先将数据进行 embedding 层处理，即文本向量化。本文利用 word2vec 模型训练文本语料库，得到词向量。单个文本的最大词语数目为 v 个:$\{t_1, t_2, \cdots, t_v\}$，每个词语的词向量为 d 维，组成一个 $v*d$ 维的矩阵，作为文本分类模型的输入，传入多尺寸卷积神经网络模型(如图 2-1 所示)。

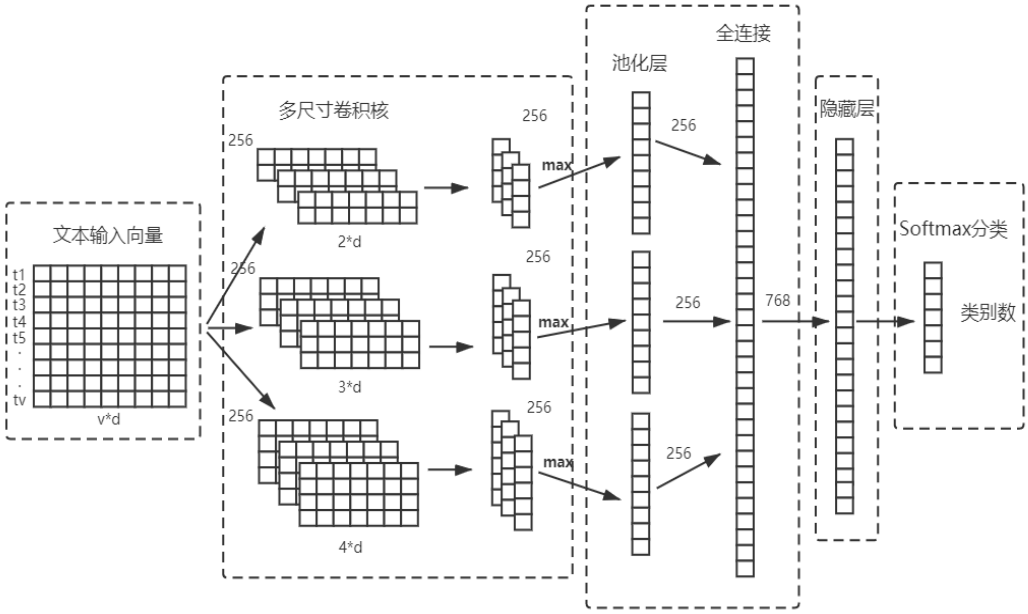


图 2-1 TextCNN 模型的网络结构

1、卷积层

将文本向量作为卷积神经网络模型的输入，使用多尺寸卷积核进行卷积，提取不同类型的多维特征。每个卷积核都设有一个固定的滑动窗口，每次对窗口内的特征进行卷积，并使用激活函数激活。本文使用的卷积核高度 h 分别为 2、3、4，滑动步长为 1，每个尺寸的卷积核数目为 256 个。

$$\{t_{1:h}, t_{2:h+1}, \cdots, t_{v-h+1:v}\} \tag{1}$$

公式(1)表示输入文本的窗口。

每个窗口 $t_{s:s+h-1}$ 卷积特征值的计算公式为：:

$$c_s^h = f(\sum_{s \in v} w_h \otimes_{s:s+h-1+b_h}) \tag{2}$$

其中 w_h 为卷积核的权重 $w_h \in R^{h*d}$ ，h 为卷积核的高度，$b_h \in R$ 为偏置，s 代表卷积核的滑动窗口的参数，$\otimes$ 为卷积计算，$f(x)$ 为激活函数，常用的激活函数为 $Sigmoid$ 、$Tanh$ 和 $\mathrm{Re}lu$ 等，本文采用 $\mathrm{Re}lu$ 函数激活，$\mathrm{Re}lu$ 函数能够更好地学习优化。

滑动窗口经过一个卷积核卷积后的特征图为：:

$$C_1^h = \left[c_1^h, c_2^h, \cdots, c_{v-h+1}^h\right] \tag{3}$$

2、池化层

句子的语义通常集中在几个单词和短语，因此经过卷积核输出的数据会带有许多无用特征，故需使用池化层缩小参数的尺寸，使得全连接层的输入参数减少，过滤一些冗余的特征。使用池化层可以加快模型计算速度，防止模型过拟合。在卷积层后使用最大池化（Max – pooling）提取特征图中最有用的特征 $C_{1,\max}^{h}$。

每个尺寸都有 256 个卷积核，则尺寸 $h=2$ 卷积后的特征图为：

$$C^{2}=\left[c_{1,\max}^{2},c_{2,\max}^{2},\cdots,c_{256,\max}^{2}\right] \tag{4}$$

尺寸 $h=3$ 卷积后的特征图为：

$$C^{3}=\left[c_{1,\max}^{3},c_{2,\max}^{3},\cdots,c_{256,\max}^{3}\right] \tag{5}$$

尺寸 $h=4$ 卷积后的特征图为：

$$C^{4}=\left[c_{1,\max}^{4},c_{2,\max}^{4},\cdots,c_{256,\max}^{4}\right] \tag{6}$$

将经过 Max-pooling 后的不同尺寸特征图进行拼接，构建文本的全局特征图：

$$C=\left[c^{2},c^{3},c_{\mathrm{x}}^{4}\right]=\left[T_{1},T_{2},\cdots,T_{768}\right] \tag{7}$$

共提取出 768 个特征，作为全连接层的输入。

3、全连接层

将 768 个特征输入至全连接层，隐藏层节点数为 256，输出层节点数为类别数，全连接神经网络的计算公式为:

$$C^{'}=f(W_{1}C+b_{1}) \tag{8}$$

$$C^{''}=f(W_{2}C^{'}+b_{2}) \tag{9}$$

其中 W_{1},W_{2} 为全连接层的两层权重，b_{1},b_{2} 为偏置，$f(x)$ 采用激活函数 $\mathrm{Re}\,lu$ 激活.

4、分类函数softmax预测

全连接层后输出的特征图为 $\mathrm{C}''=[\mathrm{T}_{1}'',\ \mathrm{T}_{2}'',\ldots,\ \mathrm{T}_{n}'']$，$n$ 为文本的类别数。采用softmax 分类函数预测第 l 个文本类别的概率值，找到最大概率值的位置即为预测出的类别：

$$p_l = \frac{\exp(C_l^{"})}{\sum_{j=1}^{n}\exp(C_j^{"})} \quad (10)$$

多尺寸卷积神经网络进行一次前向传播后，利用反向传播来对卷积核的权重进行更新，进行多次更新后，取得最优的预测模型。

2.4 模型求解

2.4.1 实验条件

实验环境如表 2 所示：

表 1 实验环境

实验环境：	操作系统	CPU	内存
环境配置：	windows 10	GeForce GTX1050	8GB
实验环境：	编程语言	开发环境	开发工具
环境配置：	Python3.6	1.0.1pytorch	Spyder

2.4.2 数据选择

本文选取来自题目附件 2 的数据集，利用 9210 条关于城乡建设、环境保护、交通运输、教育文体、劳动和社会保障、商贸旅游以及卫生设计这几个方面的群众问政留言记录作为实验的数据。将留言内容按 8：1：1 的比例划分为训练集，验证集和测试集。

2.4.3 算法流程

根据题目要求对留言内容进行分类，首先确定留言内容，由于考虑到附件 2 中同时具有留言主题与留言详情，因为留言详情为主要的留言内容，但留言主题也具有一定的重要性，因此，这里将留言主题与留言详情相结合组成留言内容。接着通过以下算法流程进行分类。算法流程主要分为以下几个步骤：

步骤 1：将留言内容数据进行预处理，去掉留言内容数据中的停用词、低频词等，然后通过 jieba 分词进行中文分词。

步骤 2：利用外部语料搜狗新闻 300 天来预训练$(\text{pre}-\text{train})$词向量（模型下载地址：https://pan.baidu.com/s/14k-9jsspp43ZhMxqPmsWMQ），接着用预

训练的词向量矩阵初始化预处理完的留言内容文本数据，获取文本的特征词向量表示。

步骤 3：将生成的词索引通过 TextCNN 模型的卷积层，使用多个卷积核来提取多个特征，通过池化层来获取文本的重要部分。

步骤 4：最后加入到 TextCNN 模型的全连接层和 softmax 分类器，对留言内容进行分类为：城乡建设、环境保护、交通运输、教育文体、劳动和社会保障、贸易旅游、卫生计生这七个一级标签，并计算评估指标，通过预测准确率、召回率 F1 值和混淆矩阵来评估模型的分类效果。

下面是对留言内容进行一级分类的算法流程图如**图 2-2**。

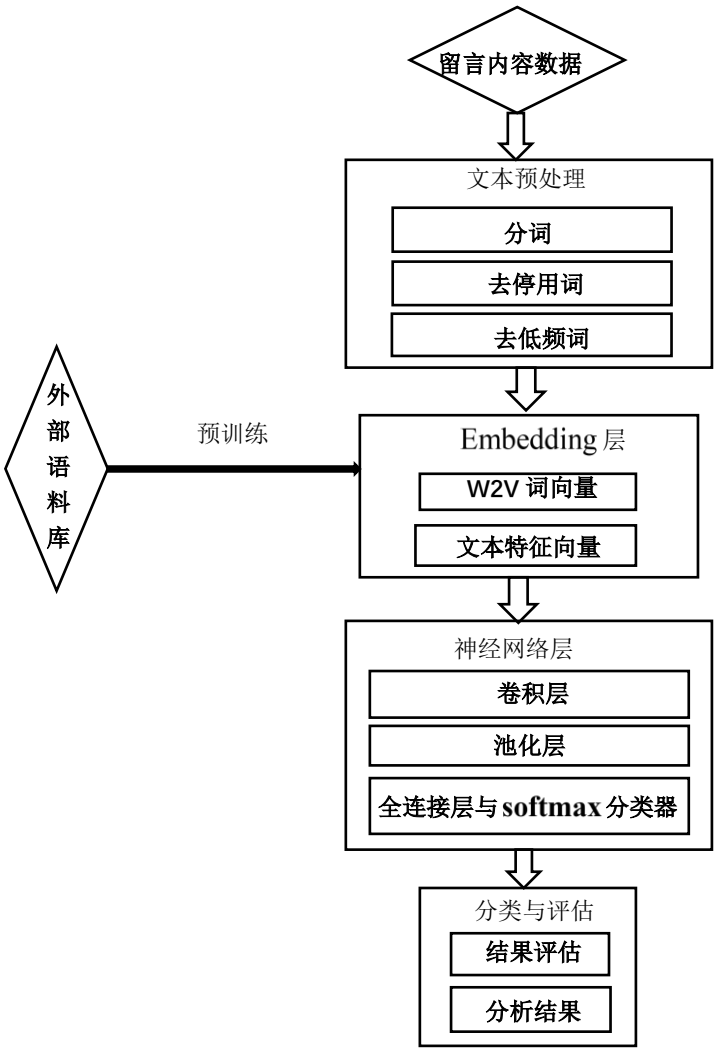


图 2-2 对留言内容进行一级分类的算法流程图

2.4.4 程序框架

模型程序主要分为数据准备和模型训练、测试两部分。

1、数据准备部分

程序代码文件为data _ set.py：读取附件 2 文本数据，将留言详情和留言主题合并为留言内容，通过建立地方自定义词典 newdic1.txt，对留言内容进行基于自定义词典的结巴分词，接着基于 stopword.txt 停用词表的清除停用词（综合了"百度停用词表"，"哈工大停用词表"，"四川大学机器学习实验室停用词表"等若干停用表），然后通过去重、词合并成句子等操作。最后以 8:1:1 的比例划分数据集为训练集，验证集和测试集，将一级分类标签转变为 0 到 6 的标签数字，以并“留言内容（空格）标签数字"的格式分别写入 txt 文件中。

2、模型训练、测试部分

程序代码文件 run.py 是设置模型的主函数部分。通过在命令行中运行：

“Python run.py—model TextCNN”

文件即可完成该部分的任务。程序先读取已处理的 txt 文件数据集，通过 utils.py 代码进行构建文本字元素集词表，并切割留言内容文本长度至 Pad _ size 长度，将处理好的留言内容输进入 embedding 层，利用预先下载好的搜狗 300 天预训练语向量模型（字向量）初始化留言内容，从而获得初始化字向量矩阵；最后程序通过 textcnn.py 模型文件以及训练测试执行代码文件 train _ eval.py 进行模型训练测试，训练完成后获取模型结果，并将该训练模型存为 TextCNN.ckpt 文件。

2.4.5 模型的训练

卷积神经网络搭建参数的不同也会影响到实验效果，本文主要通过查阅资料以及不同参数的实验结果对比，确定本实验的卷积神经的主要参数如表 2 所示：

表 2 模型主要参数

参数	值
每句话处理成的长度 **(Pad _ size)**	2000
迭代次数（**num _ epochs**）	20
丢弃率（**dropout**）	0.5
学习率（**learning _ rate**）	le-3
最小批量（**mini _ batch)**	40
卷积核数量（**num _ filters**）	256
卷积核窗口高度（**filter _ sizes**）	2,3,4

通过表 1 参数搭建卷积神经网络，采用训练集对 TextCNN 模型进行训练，得出分类结果模型。

2.4.6 模型的测试

根据测试集对模型的参数进行调整，作出模型准确率与损失值受几个参数的影响的曲线图如下所示。

四个图像分别是：准确率随训练集、测试集的样本数的变化曲线；损失值随训练集、测试集的样本数变化曲线。

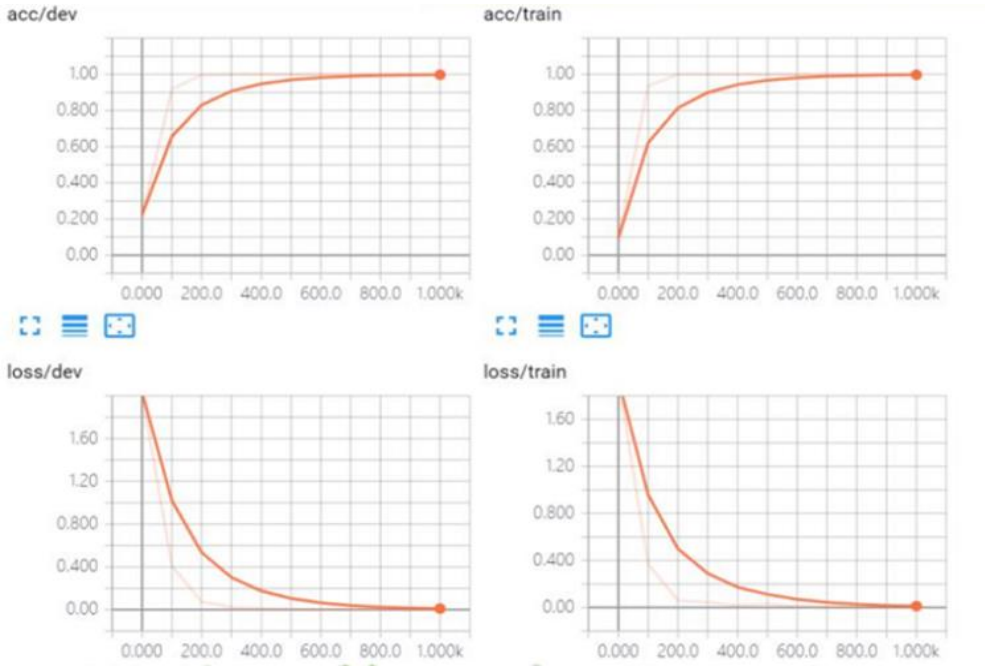


图 2-3 变化曲线

由图 2-3 中可以看出，随着训练数据量的增多，准确率在上升，损失值在下降。说明了此分类模型对留言内容的分类效果不错。

2.5 模型评价

2.5.1 评价指标

通常情况下，评价文本分类的结果主要是从算法简单性、算法复杂度以及算法有效性三方面进行考虑的。本文主要考虑的是算法的有效性能，使用的评价指标[1]为：准确率 (Accuracy) 、召回率 (Recall) 、精准率 (precision) 、F_1 得分 (F-Score) 、混淆矩阵。*TP* 表示预测为正样本且分类正确的样本数，*TN* 表示预测为负样本且

分类正确的样本数，FP 表示实际为负且分类错误的样本数， FN 表示实际为正且分类错误的样本数。

1、准确率：被正确分类的样本数占总样本数，计算公式如下。

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

2、召回率：被正确分类的正样本数占正样本总数的比例。

$$\text{Recall}=\frac{TP}{TP+FN} \tag{12}$$

3、精确率：被正确分类的正样本数占所有被分成正样本数的比例

$$\text{Precision}=\frac{TP}{TP+FP} \tag{13}$$

4、F1 值：基于精确率和召回率的调和平均值

$$F_1=\frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i} \tag{14}$$

其中， P_i 为第 i 类的查准率， R_i 为第 i 类的查全率

5、混淆矩阵：从预测值与实际值之间的匹配程度两个维度生成矩阵，对角线上元素为分类模型预测正确的数量，其它位置元素为不同种类的预测错误情况。混淆矩阵表 3 如下：

表 3 n 分类混淆矩阵

实际/预测	**预测为类 1**	**预测为类 2**	**预测为类 3**	…	**预测为类 n**
实际为类 1	类 1 预测正确数	预测类 2 实际类 1		…	
实际为类 2		类 2 预测正确数		…	
实际为类 3			类 3 预测正确数	…	
…	…	…		…	
实际为类 n			预测类 2 实际类 n	…	类 n 预测正确数

2.5.2 评价结果

1、F-score

选出最优模型后，对测试集进行验证，得出准确率为 91.65%，由于准确率还不够说明具体分类的效果，故还需求出预测标签结果的精准率、召回率、F-score 结果如**表** 4 所示，可以看出留言内容分类结果的精准率、召回率、F-score 最多能够达到可达到 96%，说明 TextCNN 模型的对留言内容数据的预测效果不错。

表 4 **F-score 结果**

分类	精确率	召回率	*f1* 得分	支持样本数
城乡建设	0.8838	0.8838	0.8838	198
环境保护	0.9149	0.9348	0.9247	92
交通运输	0.9074	0.8305	0.8673	59
教育文体	0.9551	0.9613	0.9582	155
劳动和社会保障	0.9572	0.9275	0.9421	193
商贸旅游	0.8512	0.8957	0.8729	115
卫生计生	0.9318	0.9535	0.9425	86
总体	0.9145	0.9124	0.9131	898

2、混淆矩阵

通过求解得出模型的混淆矩阵结果如下表，从表 5 中可以看出，混淆矩阵对角线上的结果达到最大，说明留言内容的分类结果的正确率较高，非对角线上基本为零，说明只有少数的留言内容是被误分类为其他标签的。

表 5 模型的混淆矩阵

预测 / 实际	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	卫生设计
城乡建设	175	6	3	2	2	10	0
环境保护	5	86	0	0	0	1	0
交通运输	4	0	49	0	0	5	0
教育文体	3	0	0	149	0	1	0
劳动和社会保障	6	0	0	0	179	1	4
商贸旅游	5	1	2	2	0	103	2
卫生设计	0	0	0	0	0	0	82

3 问题二：

3.1 问题分析

题目要求对留言进行归类并定义热度评价指标求出热度排名结果，首先要对留言数据归类就必须确定文本归类算法，由于附件 3 数据中没有标签，因此，可以使用无监督聚类方法对留言进行归类，在使用聚类算法之前还需要将文本数据进行预处理。热度排名可以通过建立热度计算模型来求解热度，模型的建立需要选择热度评价指标，可以根据附件 3 中的有关数据，选择对热度有影响的正负指标并确定指标的权重。热度排名前 5 的类需要将每一类的留言问题进行简要概括，得出类内部的问题描述。

3.2 模型建立

3.2.1 热度评价指标

根据题目要求，要解决热点问题，需要定义合理的热度评价指标对热点问题进行排序，则需要定义热度评价指标受热点问题的哪些因素所影响，并定义相对应的较为恰当的权重值，从而得到热度指数。

1、指标的选择

根据实际情况，各类社情民意反应的问题的热度代表一个问题被社区人员反复反应的程度，要计算热度指数，就需要对指标进行选择，指标即代表热度的影响因素，根据附件 4 的数据，选择的指标如下所示：

（1）留言数量：某一留言问题留言的人数和次数越多，则代表该问题影响的程度越大， 对热度的计算影响很大。

（2）时间间隔：对于某一留言问题，跨越的时间越短，说明在单位时间内，该问题不断被提及的次数越多，即也说明了该问题为大多数人集中讨论的话题，对热度的定义有一定的影响。

（3）点赞数：对于某一留言问题，下面点赞的人数越多，说明该问题的情况越属实，对热度定义十分重要。

（4）反对数：对于某一留言问题，下面反对的人数越多，说明该问题可能只有极少数情况会发生，热度不高。

因此，用留言的点赞数、反对数、时间间隔以及人员提及该问题的留言数量基本能衡量一个问题的热度。

2、热度指数计算模型

对于某一条热点问题 A 定义其热度：

$$H_A = \beta_1 T + \beta_2 N_{留言} + \beta_3 N_{赞成} - \beta_4 N_{反对} \quad (15)$$

其中，T 代表留言问题的时间间隔，单位为天数；$N_{留言}$ 代表社区人员对该问题的留言数；$N_{赞成}$ 代表对该问题的赞成数；$N_{反对}$ 代表对该问题的反对数；$\beta_1, \beta_2, \beta_3, \beta_4$ 代表权重因子， $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$。

由于热度指数是反应该问题的在社区被重视的程度，留言数反应了重视对该问题的社区人数，权重呈正值；时间间隔长代表单位时间内问题被提及的次数少，

热度不大，即时间间隔与热度成反比，故权重呈负值；反对数反应社区反对该问题的真实情况，故权重呈负值，赞成数则相反，权重呈正值。

根据各指标的重要程度，并通过多次权重设置测试结果，联系本题数据，得出最优的权重值分别为： $\beta_1=0.1,\beta_2=0.3,\beta_3=0.4,\beta_4=0.2$ 。

综上所述，热度指数计算模型的数学公式为：

$$H_A=-0.1\times T+0.3\times N_{留言}+0.4\times N_{赞成}-0.2\times N_{反对} \quad (16)$$

3.3 模型求解

3.3.1 留言的归类

为了将留言内容进行归类，需要先预处理留言内容，然后再利用聚类算法对留言数据进行聚类，从而达到留言归类的效果。

1、留言内容预处理

由于题目所给的留言数据可能存在一些对反映留言内容没用的词语，因此需要对留言数据进行清洗，清洗过后再分词，从而提取出重要的词进行特征提取。留言内容预处理框架图图 3-1 所示：

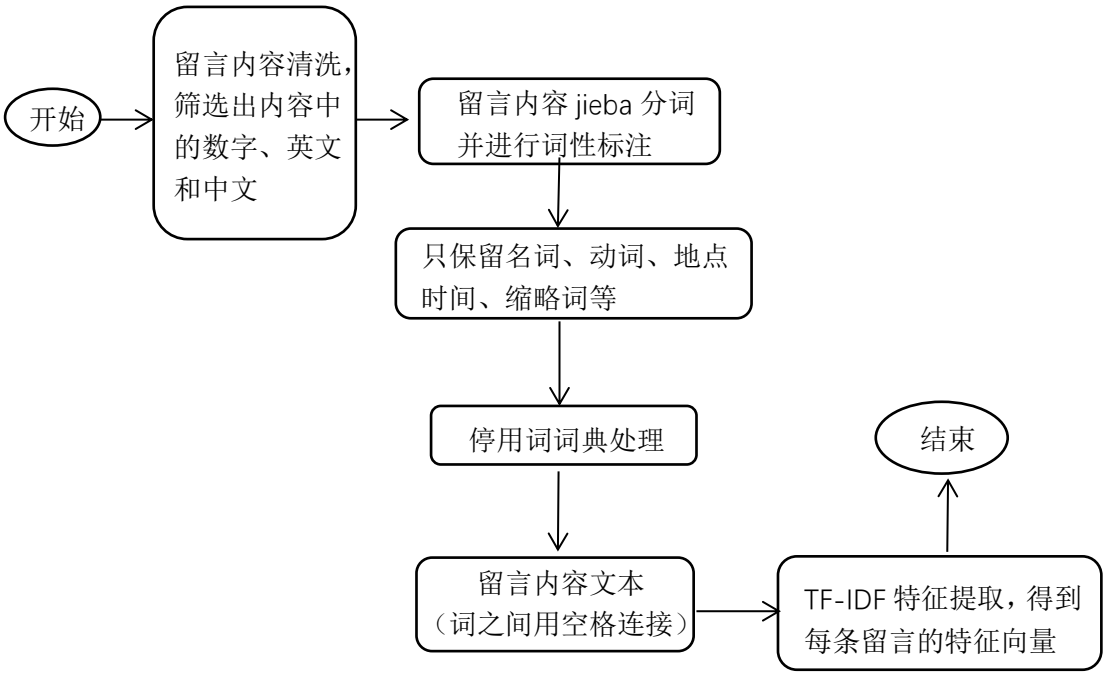


图 3-1 留言内容预处理

由图 4 可知，数据预处理过程分为四部分，分别是：留言数据清洗、分词和词性标注、关键词提取以及特征提取。

(1) 留言内容的清洗

根据附件 3 可以看到一些空白字符，或者会出现用户打错的词语等与留言内容无关的词汇，为了清理这些没用的内容，可以利用正则表达式将其匹配，并替换其为空字符。

对于留言内容的聚类，大部分标点符号也会影响聚类的效果，因此需要将留言详情的标点符号全部删除，从而剩下的中文、英文和数字才能更好地反映留言内容。

(2) jieba 分词及词性标注

留言内容清洗完成之后，需要进行分词操作。 jieba 库是一个简单实用的中文自然语言处理分词库，jieba 分词[3]算法使用了基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能生成词情况所构成的有向无环图(DAG)，再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，使用了 Viterbi 算法。 jieba 分词中提供了词性标注功能，可以标注句子分词后每个词的词性，词性标注集采用北大计算所词性标注集，属于采用基于统计模型的标注方法。

(3) 关键词提取

由于形容词、副词、助词等对留言内容的特征贡献不大，因此在内容特征提取的过程中，只提取名词、动词、地点、时间等词性的词语，以便能更好地分辨这条内容。然而，提取这些词还不能够达到目的，为了聚类效果更好，需要设置停用词词典，在分词之后，把在停用词词典中不需要的词在分词后的词中去除。

(4) TF-IDF 留言内容特征提取

上一步筛选出关键词之后，用这些词进行特征提取，因为其相比那些普通词而言，重要性更大，本文使用的是 TF－IDF 来提取特征。

通过 TF－IDF 特征提取之后，每条留言内容都有一个向量标识。向量上的每一个值都是一个词的 TF－IDF 值。其向量获得方式为首先统计出所有词，然后把每个词当成向量的每一个维度，如果该文档中有词，就在某词的维度上计算它的 TF－IDF 值；如果不存在某词，那么某词的维度上的值为 0。这种方式对留言内容进行特征提取，得到的结果是一个稀疏矩阵。

TF－IDF [4]的计算方法如下：

TF 代表词频，表示某特征词在某文本文件中的出现次数，其定义为：

$$T_F = \frac{n_w}{\sum_k^N n_k} \tag{17}$$

其中，n_w 是特征词 w 在文本文件中的出现次数；N 为该文本文件中特征词的数量；T_F 为衡量特征词在文本中重要程度的指标。

IDF 代表逆向文件概率，其定义为：

$$I_{DF} = \log \frac{D}{I+Q} \tag{18}$$

其中，D 是所有文本文件的总数；Q 是包含特征词的文本文件数量；I_{DF} 为衡量特征词在所有文本文件中重要程度的指标。

将式（1），（2）相乘，可得文本特征词权重

$$f_{TF-IDF} = T_F \times I_{DF} \tag{19}$$

2、留言内容聚类

根据题目要求，需要对特定地点或者人群进行归类，对于 *BIRCH* 算法聚类，数据的插入顺序对聚类特征树的构建影响很大，所以本文通过对初始数据集进行预聚类的方法，即首先使用 K－means 算法对特定地点进行预聚类，接着再使用 *BIRCH* 算法对第一次聚类的子类依次聚类，由此便可将特定地点的不同类别留言内容归类出来。在预聚类阶段采用的算法是 K－means 算法，因为该算法效率高、复杂度较低，不会在初始阶段增加 *BIRCH* 算法的复杂度，从而使两种聚类算法达到了互相优化的作用。

(1) K－means 算法聚类

K－means 算法[5]实际上就是通过计算不同样本间的距离来判断他们的相近关系的，相近的就会放到同一个类别中去。K－means 聚类算法的基本思想如下：

算法流程如下：

输入：簇数目 k，地点特征的文本向量矩阵

①随机选取 k 个点，作为聚类中心；

②计算每个点分别到 k 个聚类中心的聚类，然后将该点分到最近的聚类中心，这样就行成了 k 个簇；

③再重新计算每个簇的质心（均值）；

④重复以上②到④步，直到质心的位置不再发生变化或者达到设定的迭代次数。

输出：簇的集合$\{C_1, C_2 \cdots\cdots C_k\}$

(2) BIRCH 算法聚类

BIRCH 聚类算法[6]是一种使用聚类特征树的多阶段聚类算法，该算法将数据集中的所有数据点逐步加到树形结构中。*BIRCH* 聚类算法中用聚类特征来代表一个类簇，聚类特征树用来表示聚类的层次结构。

聚类特征 *CF* 在 *BIRCH* 聚类算法中用一个三元组 (N, LS, SS) 来表示，其中 N 代表了类簇中所有数据点的个数，LS 代表当前类簇中所有数据点的线性和，SS 代表当前类簇中所有数据点的平方和。依据以上聚类特征信息，类簇的许多统计量可以被很容易推出如类簇的形心和类簇半径等。通过当前的数据集聚类特征，数据点可以快速地找插入树的位置。

算法聚类步骤如下：

输入：使用 K – means 算法预聚类得到簇$\{C_1, C_2 \cdots\cdots C_k\}$

①择第一个簇C_k，并入其中的第一个数据点

②**Repeat (重复)：**

先根据节点中的所有数据点的CF计算新数据点的CF值；

其次从根节点开始插入新的CF值，自上而下选择离该CF值最近的节点，值到到达叶子节点，选择叶子节点，选择叶子节点中离该CF值最近的簇进行预插入；

接着返回检查，若插入后，簇直径大于预先设定的阈值 T，则不进行插入，把新数据作为一个新的簇插入到叶子节点

最后返回到某个节点，若该节点的子节点数目因为插入而增加并大于先设定的阈值 B 或者 L 时，则将该节点分裂，若分裂的是根节点则新增节点数，即树的高度增加 1

③**Untill（直到）**最后的簇C_k中的所有数据都插入到CF树中，每个叶子节点所包含的数据组成一个簇，即为一类。

输出：簇$\begin{Bmatrix} C_{11} & C_{12} & \cdots & C_{1x} \\ C_{21} & C_{22} & \cdots & C_{2x} \\ \vdots & & & \\ C_{k1} & C_{k2} & \cdots & C_{kx} \end{Bmatrix}$

其中，x表示预聚类中每一类被分成的类数，不同的类的x不同。

3.3.2 基于 TextRank 算法的类内部留言问题描述

实现了对留言的聚类和热度排名之后，对于热度指数排名前 5 的留言类别，想要直观的观察留言聚类的效果，还需要进行统计分析，为了能清楚地了解排名前 5 每个类的话题信息，可以使用 TextRank 算法[3]提取出留言内容的关键词，并使用自动文摘方法提取出该留言的关键句，同时对类内部的每条留言的关键句进行排行，得出每个类的留言问题概括描述。

1、TextRank 的自动文摘算法

TextRank 是一种基于图的用于文本的排序算法，基本思想来自于 Google 的 PageRank 算法。TextRank 可提取关键字，也可以进行自动文摘。用于自动文摘的思想是：将每个句子看成 PageRank 图中的一个节点，若两个句子之间的相似度大于设定的阈值，就认为这两个句子之间有相似联系，对应的这两个节点之间便有一条无向有权边，边的权值是相似度，接着利用 PageRank 算法即可得到句子得分，把得分较高的句子作为文章的摘要。

TextRank 算法的主要步骤：

① 预处理：分割原文本中的句子得到一个句子的集合，接着对句子进行分词、去停用词处理，筛选出关键词集。

② 计算句子之间的相似度：

$$\text{句子相似度}=\frac{\text{两个句子都出现的词的数目}}{\log(\text{句子1中词的数目})+\log(\text{句子2 中词的数目})} \quad (20)$$

对于两个句子之间的相似度大于设定的阈值的两个句子节点用边连接起来，设置其边的权重为两个句子的相似度。

③ 计算句子权重：

$$\text{句子1的权重}=(1-\text{阻尼系数})+\text{阻尼系数}\times\sum_{\text{与句子1相连的所有句子}}\frac{\text{句子1和句子2 的相似度}\times\text{句子2 的权重}}{\text{所有与句子2 相连的句子的边的权重和}} \quad (21)$$

由公式（9）可多次迭代计算直至收敛稳定之后可得各句子的权重得分。

④ 形成文摘：将句子按照句子得分进行倒序排序，抽取得分排序最前的几个句子作为选文摘句，再依据字数或句子数量要求筛选出符合条件的句子组成文摘。

2、基于TextRank自动文摘算法的类内部主题排行

为了能使用TextRank算法，将留言内容合并成类似于文章的样子。接着调用TextRank算法进行计算，得到关键词以及关键词的权重，对关键词进行排序，对于权重最高的句子，认为是该类的主要留言问题描述。

利用TextRank算法进行类内部主题排行过程如图 3-2 所示：

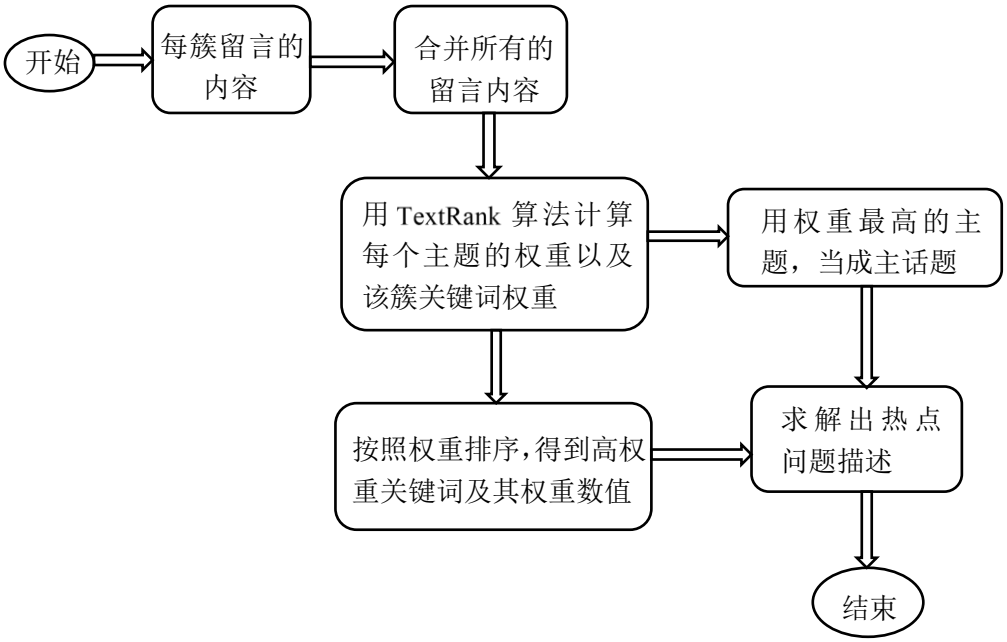


图 3-2 主题重要性排序

3.3.3 热点问题留言明细表的求解

对于热点问题留言明细表的求解，主要是求出每条留言的问题 ID，即将留言进行归类。本文主要是通过无监督聚类算法，识别出相似的类别。根据留言内容预处理，提取出每个留言的地点名词，首先使用K－means聚类算法对特定地点进行预聚类，通过运行主程序文件clf_rank.py聚类模块kmeans预聚类部分代码（其中，数据处理以及特征向量化方法函数分别在preprocessing.py、counter.py代码文件中）一共聚出8类，结果如表 5所示。

表 5 k-means 聚类结果

类别	1	2	3	4	5
留言数（条）	2708	469	258	226	211
类别	6	7	8		
留言数（条）	169	154	131		

接着，将聚出的每个类依次输入 *BIRCH* 聚类算法中，通过运行主程序文件 clf_rank.py 聚类模块 birch 聚类部分代码（其中，数据处理以及特征向量化方法函数分别在 preprocessing.py、counter.py 代码文件中）得出每个类聚出的子类结果如表 6 所示。

表 6BIRCH 聚类结果

类别	1	2	3	4	5
子类数（类）	100	20	10	10	10
类别	6	7	8		
子类数（类）	10	10	10		

从表中可以看出，留言详情一共被分为 180 类，分别定义为问题 ID1 到问题 ID180。最多的子类的留言数可达至 386 条，最少的只有 3 条。接着在附件 3 中给每条留言加上其所属的问题 ID，保存为“**热点问题留言明细表**”，如“作品附件 2”所示。

3.3.4 热点问题表的求解

对于热点问题表的求解，需要求解出每一类留言问题的热度指数，再根据热度指数进行排序，得出排名前 5 的热点问题。同时，确定排名前 5 每一类留言的问题概括，即求出热点问题所发生地点或人群，并对问题进行简洁描述。

1、热度指数的求解

本文使用建立的热度计算模型来求解热度指数，根据热度计算模型，首先需要将指标量化，即根据聚类出的 180 类留言问题，统计出每一类的留言数量、留言跨越的时间间隔、点赞数和反对数，并对数据进行 max-min 标准化，输入到热度计算模型公式（11）中，从而求解出每一类的热度指数。对 180 类的热度指数进行从大到小排序，热度指数越高代表热度越好，即为热点问题。通过运行

clf_rank.py 主程序文件热点问题模块即可求出热度指数排名前五的热点留言问题及其热度指数如表 7 所示：

表 7 热度指数结果

问题 ID	60	2	101	141	3
热度指数	0.29659	0.25562	0.24320	0.22076	0.20584

从表中可以看出问题 ID 为 60 的类的热度指数最高，即最为热点问题。

2、地点/人群及问题描述的求解

本文使用 TextRank 算法求解排名前 5 的热点留言的问题描述。首先将每一类的留言内容依次输入 TextRank 算法中，求出类内部的关键词、关键词权重和关键留言主题，接着并对其进行概括，从而求解问题描述。从关键词中选择关键词权重最高的地点名词作为热点问题的地点，modeling.py 代码文件存有 TextRank 算法以及数据处理方式，通过运行 clf_rank.py 主程序文件热点问题模块可得出热点问题的关键词权重、关键词、关键主题，经过人工统计，可将热度排名前 5 的地点、问题描述列出，如表 8 所示。

表 8 问题描述结果

热度排名	问题 ID	地点/人群	问题描述
1	60	西地省 A 市人民路	小区水利改造、小区车位、公司违纪欺骗拖欠工资
2	2	A7 县 A1,3,4 区楚江、新村、北辰定江洋小区	小区违规，施工、噪音扰民
3	101	A 市	公共设施建议，公司，建设拆迁管理、咨询问题
4	141	A 市 A2 区	小区垃圾，拆迁改造、住房不安全、街道违规行为投诉
5	3	A 市 A3 区	小区晚上夜宵店油烟、晚上店噪音、渣土车噪音扰民

按照题目要求，将热度排名、问题 ID、热度指数、时间范围、地点/人群以及问题描述统计出，保存为“**热点问题表**”，如“作品附件 2”所示。

4 问题三：

4.1 问题分析

该问题是对留言的答复意见进行质量评价，并给出一套评价方案。因此，需要通过建立评价模型来评定高低质量。模型的建立需要选择评价指数，评价指标需要从相关性、完整性、可解释性等角度进行定义，因此需要从留言的答复意见中提取指标对这几个指标进行量化，给出评价指标，设置阈值，超过阈值的答复意见为高质量回复。最后，从留言的答复意见中提取特征利用分类器的对该评价指标的好坏进行评价。

4.2 数据预处理

由于附件 4 中答复意见的部分数据只是一个日期，并未对留言详情进行答复，故将这部分数据进行删除，将 241、515、525、526、527 等 41 个数据删除。由于在进行计算平均句子长度时，需要答复意见有“。”，“！”，“？”，故对没有这几个标点符号的答复意见末端添上“。”，以便于平均句子长度的计算。因此将附件 4 中 1360、2010、2108、2259 等 12 个数据的答复意见末端添上“。”。

4.3 模型建立

4.3.1 指标的选择

1．相关性：

相关性描述的是社区留言所提问题的文本和答复意见文本之间的主题相关性。一般情况下，留言详情与答复意见越相关，则可认为该答复意见的质量评价指数越大，即社区用户留言详情与答复意见的主题相关性与质量评价指数成正相关。主题相关性由留言详情和答复意见间的文本相似度表示。

2.完整性：

完整性描述的是社区留言答复意见的完整程度是否满足某种规范，完整性与文本长度和标点字符占比[8]有关，文本长度太短无法完整回答群众留言问题，太长显得繁杂；同时如果标点符号占比过大，答复意见的有价值信息就相对不足。

3.可解释性：

可解释性描述的是答复意见对留言详情提出的问题是否带有感情色彩。一般情况下，答复意见带有的感情色彩越多时，可以认为该答复意见的可解释性越低。

感情色彩由答复意见的基于词典的感情度量化表示，即感情度越小，该答复意见的质量评价指数越大。则答复意见的感情度与质量评价指数成负相关。

4.时效性：

时效性描述的是社区留言详情与答复意见之间的时间间隔，一般情况下，时间间隔越长，其答复意见对于社区用户来说丧失了使用价值，变成无用答复。[6]用户在提出问题后，必然希望尽快得到答复，因此答复的时效性影响答复意见质量评价指数的高低，可认为时间间隔越短，该答复意见的质量评价指数越大，即时效性与质量评价指数成负相关。

答复意见的时效性的计算如下：

$$T = time_a - time_b \tag{22}$$

其中，$time_a, time_b$ 分别表示答复意见的时间和留言详情的生成时间，T 反映着答复意见的时效性，该值越小表明该答复越及时，即答复意见质量评价指数越大。

4.3.2 质量评价模型

对于某一条用户提出的留言详情的答复意见 A 定义其质量评价指数模型：

$$\text{质量评价指数}=\alpha_1 S - \alpha_2 C_1 + \alpha_3 \cos C_2 - \alpha_4 E - \alpha_5 T \tag{23}$$

其中，S 表示答复意见与留言详情的相关性，即相似度；C_1、C_2 表示答复意见的完整性，即标点字符占比和文本长度；E 表示答复意见的可解释性，即情感度；T 表示答复意见的时效性。$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ 代表权重因子，$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 = 1$。

相关性与质量评价指数成正相关，完整性，可解释性，时效性与质量评价指数成负相关。

故设置相对应的权重值，$\alpha_1 = 0.2, \alpha_2 = 0.3, \alpha_3 = 0.1, \alpha_4 = 0.1, \alpha_5 = 0.3$，则质量评价模型的计算公式为：

$$\text{质量评价指数}=0.2 \times S - 0.3 \times C_1 + 0.1 \times \cos C_2 - 0.1 \times E - 0.3 \times T \tag{24}$$

4.4 模型求解

本文使用建立的质量评价指数计算模型来求解每条答复意见的质量评价指数，根据质量评价指数计算模型，首先需要将指标量化，即用基于最小编辑距离的相似度量化相关性，用文本长度和标点字符占比量化完整性，用情感度量化可解释性，用时间间隔量化时效性，并对数据进行 max-min 标准化，输入到质量评价指数计算模型公式（24）中，从而求解出每一类的质量评价指数。对所有样本的质量评价指数求均值，并将其设置为阈值，可认为高于阈值的答复意见为高质量评价，反之为低质量评价。

4.4.1 各项指标的求解

1、相关性：

主题相关性由留言详情和答复意见间的基于最小编辑距离的文本相似度表示。最小编辑距离的原理是：比较两个字符串，记录一个字符串通过移除，替换，添加操作转换到指定字符串的次数，来确定两个字符串直接的相似度。

运行程序 similari.py，通过比较基于各个距离的相似度，最终选出适用于该问题的基于最小编辑距离的相似度。将运行后的相似度结果保存于文件 similari.xls 中。

2、完整性：

完整性由文本长度和标点字符占比[9]综合表示。文本长度过短，其信息量不足，无法完整回答群众留言问题，文本长度过长，其答复时未挑出重点，显得繁杂，故给文本长度套用 cos( ) 函数，使其的最大值为 cos( ) 函数的峰值 1；如果标点符号占比过大，答复意见的有价值信息就相对不足，故标点字符占比与质量评价指数成负相关。

运行程序 zhuchengxu.py 中的注释为完整性的部分。

3、可解释性：

可解释性由答复意见的基于情感词典的情感得分表示。情感得分越小，该答复意见的质量评价指数越大，则答复意见的情感得分与质量评价指数成负相关。

运行程序 zhuchengxu.py 中的注释为情感得分的注释部分，并把运行后的结果保存于文件 qinggan.xls 中。

4、时效性：

时效性由留言详情与答复意见的时间间隔表示。时间间隔越大，则表明用户提出的问题没有得到及时的答复，对于用户来说，其信息已经失去了价值所在，故可认为时间间隔与质量评价指数成负相关。

运行程序 timelag.py，并把运行后的结果保存于文件 timelag.xls 中。

4.4.2 质量评价指数权重的确立

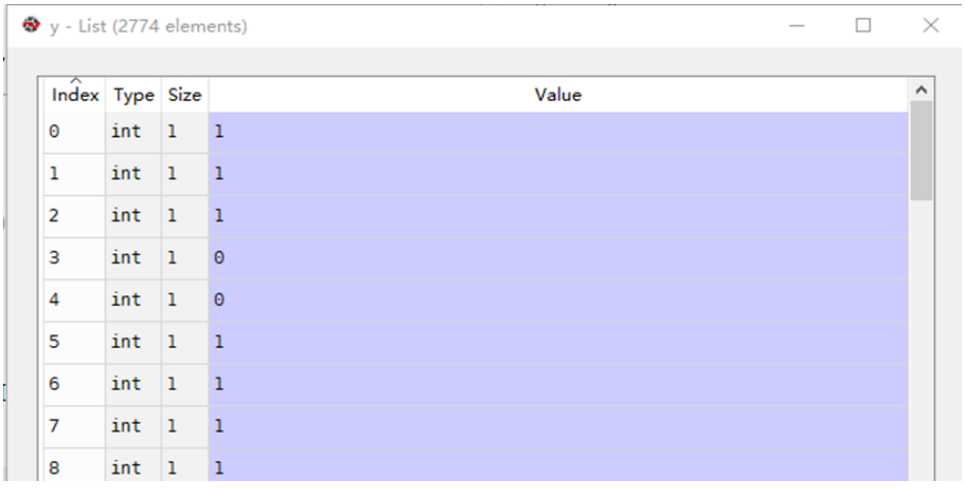
由于相似度、完整性中文本长度的 cos( ) 函数与质量评价指数成正相关，完整性中标点字符占比、用情感得分量化的可解释性、用时间间隔量化的时效性与质量评价指数成负相关。

利用二分类器的训练集准确率和测试集准确率人工调试设置相对应的权重值，$\alpha_1=0.2,\alpha_2=0.3,\alpha_3=0.1,\alpha_4=0.1,\alpha_5=0.3$ 代入公式（24），则质量评价模型的计算公式为：

$$\text{质量评价指数}=0.2\times S-0.3\times C_1+0.1\times\cos C_2-0.1\times E-0.3\times T \qquad (25)$$

其中，S 表示答复意见与留言详情的相似性，即相似度；C_1、C_2 表示答复意见的完整性，即标点字符占比和文本长度；E 表示答复意见的可解释性，即情感得分；T 表示答复意见的时效性，即留言详情与答复意见的时间间隔。

将所有样本的质量评价指数取均值设置为高低质量分类的阈值，质量评价指数大于阈值时则该样本为高质量答复，反之为低质量答复。运行程序 zhuchengxu.py 求出每个样本的高低质量部分分类结果如图 4-1 所示：



Index	Type	Size	Value
0	int	1	1
1	int	1	1
2	int	1	1
3	int	1	0
4	int	1	0
5	int	1	1
6	int	1	1
7	int	1	1
8	int	1	1

图 4-1 每个样本的高低质量部分分类结果

4.1 模型评价

由于质量评价指数计算模型由人工调参给出，故存在分类准确率未知情况。由人工调试可知，样本受时效性的影响最大，但其权重过大时也会造成准确率下下降。选出内容词密度，平均句子长度[7]作为样本特征，以质量评价指数分类后的结果作为标签，将样本划分为训练集和测试集，并用适用于二分类的分类器计算其训练集和测试集准确率。所有算法的训练集以及测试集的准确如表 9 所示：

表 9 算法准确率

算法	准确率	
	训练集	测试集
k-近邻分类器	0.738942	0.580692
逻辑回归分类器	0.633173	0.654179
随机森林分类器	0.969231	0.587896
决策树分类器	1	0.566282
支持向量机分类器	0.609615	0.608069
多项式朴素贝叶斯分类器	0.582212	0.580692
梯度增强决策树分类器	0.767788	0.619597
AdaBoost Classifier	0.650962	0.661383
高斯贝叶斯分类器	0.626442	0.649856
线性判别分析	0.630288	0.654179
二次判别分析	0.627885	0.646974

由表 9 中可以看出，验证集的准确率最高仅可达 65%左右，说明分类器对该质量评价指数的预测准确率达中等程度，即给出新的样本，将其用分类器直接预测其质量的准确率为 65%。

参考文献

[1]朱烨，陈世平.最近邻注意力和卷积神经网络的文本分类模型[R]. 1000-1220（2020）025-06.上海: 上海理工大学光电信息与计算机工程学院.2020.

[2]徐晓璐.基于深度学习的多标签短文本分类方法研究[D].桂林：桂林电子科技大学.2019.

[3]叶建成.利用文本挖掘技术进行新闻热点关注问题分析[D].广州：广州大学计算机科学与教育软件学院.2018.

[4]但宇豪.黄继风.杨琳.高海.基于 TF-IDF 与 word2vec 的台词文本分类研究.上海师范大学学报[J].2020，第 49 卷（1）.90-95.

[5]王磊.基于网格点密度估计的聚类算法研究[D].兰州：兰州大学.2015.

[6]胡鹏辉.基于多模型的问答社区答案质量评价研究[D].南京：南京师范大学大学.2019.

[7]李晨.巢文涵.陈小明.李舟军.中文社区问答中问题答案质量评价和预测.计算机科学[J].2011，第 38 卷（6）.230-236.

[8]王伟.翼宇强.王洪伟.郑丽娟.中文问答社区答案质量的评价研究：以知乎为例.图书情报工作[J].201，第 61 卷（22）.36-44.