智慧政务中的评论文本分析

摘要

近年来，随着微博以及留言信箱的普及，互联网逐渐成为政府了解民意的重要渠道。越来越多的人通过网络留言的方式向政府提出意见建议，随着留言数量的增多，如何快速高效地处理这些留言信息成了政府相关部门的首要难题。本文主要基于自然语言处理（NLP）技术对留言内容进行分类，并构建综合评价体系对相关部门的回复内容进行评价。

针对问题一，首先对文本内容进行预处理，利用 jieba 中文分词工具对留言主题进行文本分词处理，构建词频向量，同时通过自定义的停用词表对词频向量进行停用词过滤；接着，依据留言主题文本的 TF-IDF 对文本进行词向量化；最后，利用 Stacking 算法对留言进行分类。经过反复筛选，本文的初级学习器采用朴素贝叶斯、XGBoost、随机森林、支持向量机（SVM）以及多层感知器（MLP），次级学习器采用逻辑回归，验证集最终的 F1-Score 为 0.876。

针对问题二，对留言主题进行中文分词处理，构建词频向量，去除停用词。接着，计算得到 TF-IDF，并基于 K-means 聚类算法对文本进行聚类。由于初步分类结果不理想，通过绘制词云，观察留言详情具体文本内容，对聚类后的各个类别根据不同的关键词重新进行筛选，剔除无关的留言。最后，根据分类后的文本内容计算热度，热度计算公式为“点赞数＋反对数＋1”，得到热度前五的留言内容。

针对问题三，首先对留言主题、留言详情和回复内容分别进行文本预处理，包括分词、构建词频向量、去除停用词等。接着，采用 TextRank 算法分别计算留言主题、留言详情和回复内容之间的文本相似度，两者取较高的作为留言内容和回复内容的相似度。根据计算结果得到相关留言的答复相关性指标。同时，根据留言回复时间和留言上传时间，相减得到留言回复效率指标。最后，从回复效率指标和答复相关性指标出发，对这两个指标等额赋权，构建回复质量的综合评价体系，将留言分为低质量、一般质量和高质量三类。

关键词：TF-IDF；Stacking；词云；K-means 聚类；文本相似度；TextRank 算法；

Abstract

In recent years, with the popularity of microblog and message mailbox, the Internet has gradually become an important channel for the government to understand public opinion. More and more people put forward opinions and suggestions to the government through the way of network message. With the increase of the number of message, how to deal with these message information quickly and efficiently has become the primary problem of the relevant departments of the government. This paper mainly classifies the message content based on NLP technology, and constructs a comprehensive evaluation system to evaluate the reply content of relevant departments.

To solve the first problem, firstly, preprocess the text content, use the Chinese word segmentation tool of Jieba to do the text word segmentation on the message subject, construct the word frequency vector, and filter the word frequency vector through the customized stop word list; then, quantify the text according to TF-IDF of the message subject text; finally, use the stacking algorithm to classify the message. After repeated screening, the primary learners in this paper use naive Bayes, xgboost, random forest, support vector machine (SVM) and multi-layer perceptron (MLP), the secondary learners use logical regression, and the final f1-score of the verification set is 0.876.

In order to solve the second problem, Chinese word segmentation is applied to the message topic, and word frequency vector is constructed to remove the stop words. Then, TF-IDF is calculated and text is clustered based on K-means clustering algorithm. Because the preliminary classification results are not ideal, by drawing the word cloud, observing the specific text content of the message details, each category after clustering is re screened according to different keywords, and irrelevant messages are removed. Finally, the heat is calculated according to the classified text content, and the heat calculation formula is "likes + anti logarithm + 1", and the top five heat message content is obtained.

In view of the third problem, firstly, text preprocessing is carried out for the message subject, message details and reply content respectively, including word segmentation, construction of word frequency vector, removal of stop words, etc. Then, the textrank algorithm is used to calculate the text similarity between the message subject, message details and reply content respectively, and the higher of the two is taken as the similarity between the message content and reply content. According to the calculation results, we can get the response correlation index of the related messages. At the same time, according to the message response time and message upload time, we get the message response efficiency index by subtracting. Finally, starting from the response

efficiency index and response correlation index, the two indexes are given equal weight, and a comprehensive evaluation system of response quality is constructed. The messages are divided into three categories: low quality, general quality and high quality.

Key words： TF-IDF algorithm; stacking; word cloud; K-means clustering; text similarity; textrank algorithm

目录

1 引言

1.1 背景

2016 年政府工作报告中提到政府作为城市建设的基石，要大力推进“互联网＋政务服务”。由于人工智能技术的不断进步和发展，如何快速又高效地处理政府工作是相关部门的首要难题，智慧政务应运而生，具体包括医疗、物流、食品健康等方面。将人工智能应用于政府工作中，可以免去很多不必要的人力物力，例如，在主要路口安装具有人脸识别功能的监视器，就能够自动识别在逃犯等。“互联网＋政务服务”模式不仅提高了相关部门的工作效率，同时缩减了居民和企业办事的时间，打造服务型政府。

处理社情民意的留言内容是政府工作的重要内容，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。

1.2 研究内容

本文主要利用提供的留言内容，采用自然语言处理和文本挖掘的方法解决以下问题：

（1）建立关于留言内容的一级标签分类模型。

（2）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。

（3）针对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。

2 问题分析

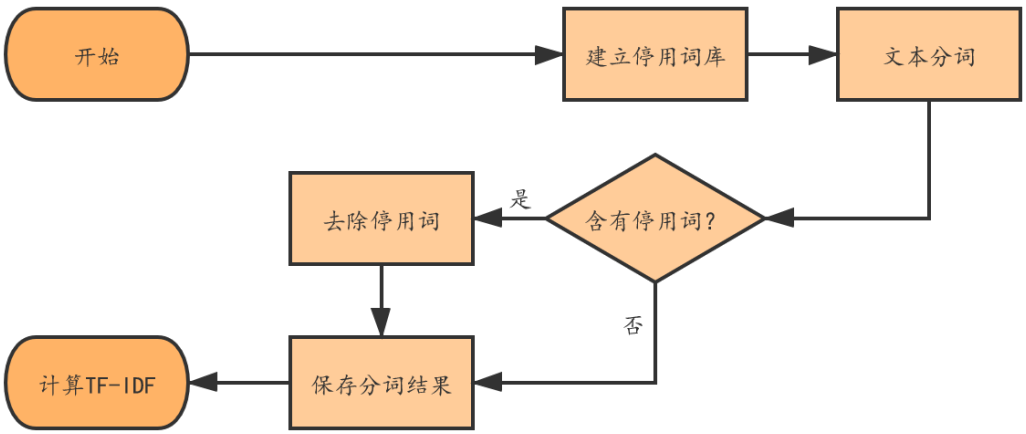

图 1 文本预处理流程图

为了实现研究内容，首先需要对所有文本内容进行预处理。先对文本内容进行分词处理，

构建词频向量，计算得到 TF-IDF，同时去除停用词，文本预处理流程图见图 1。

2.1 问题 1 的分析

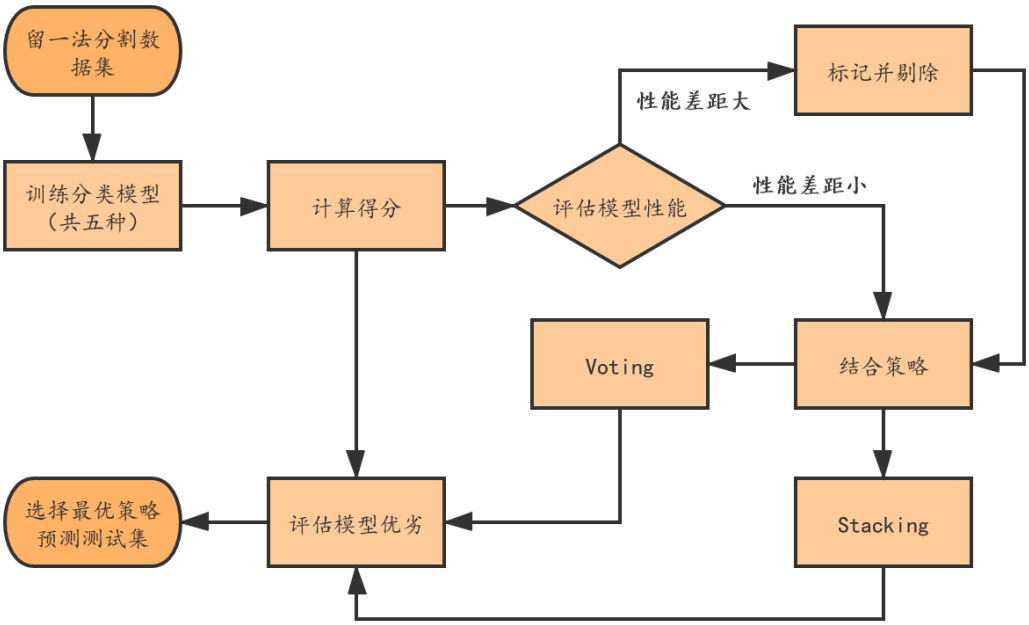

图 2 问题一流程图

附件二给出了留言内容的详细数据，包括留言编号、留言主题和留言详情等。对文本进行图 2 所示的预处理，得到能够代表文本内容的 TF-IDF 词空间向量。根据得到的词空间向量，构建文本分类模型。由于分类模型众多，在选取模型时要考虑不同模型建模的效果。

2.2 问题 2 的分析

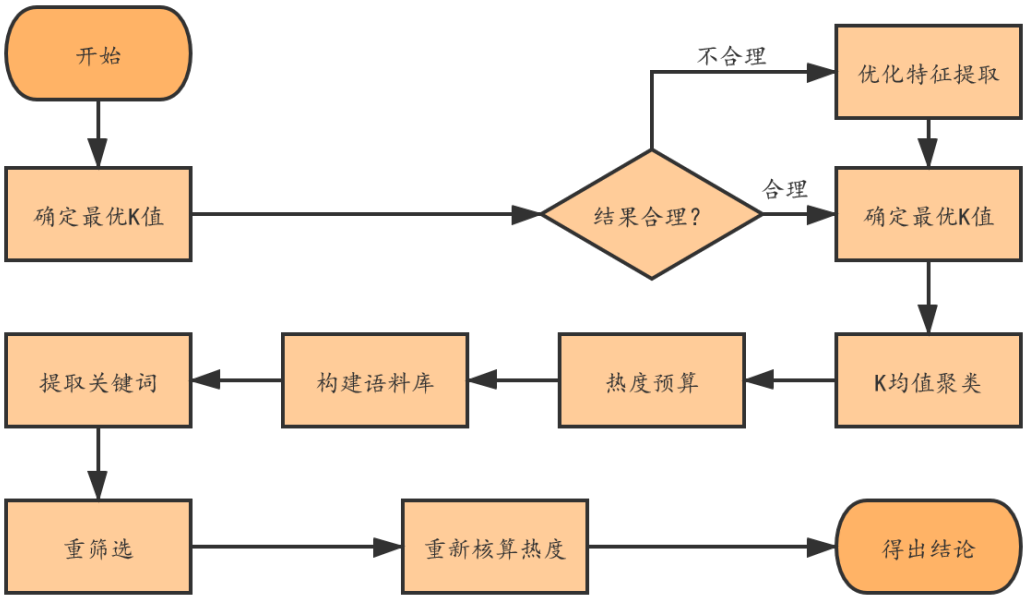

图 3 问题二流程图

附件三给出了某一时段特定地点和特定人群的留言主题、留言详情、相应的点赞数和反对数。首先对附件三留言详情的文本内容进行预处理，具体处理步骤包括分词、构建词向量、计算 TF-IDF 和去除停用词等。该问题的难点在于如何构建合理的热度评价指标，为此，需要明确，负面效应大，同样是热度高的体现。

2.3 问题 3 的分析

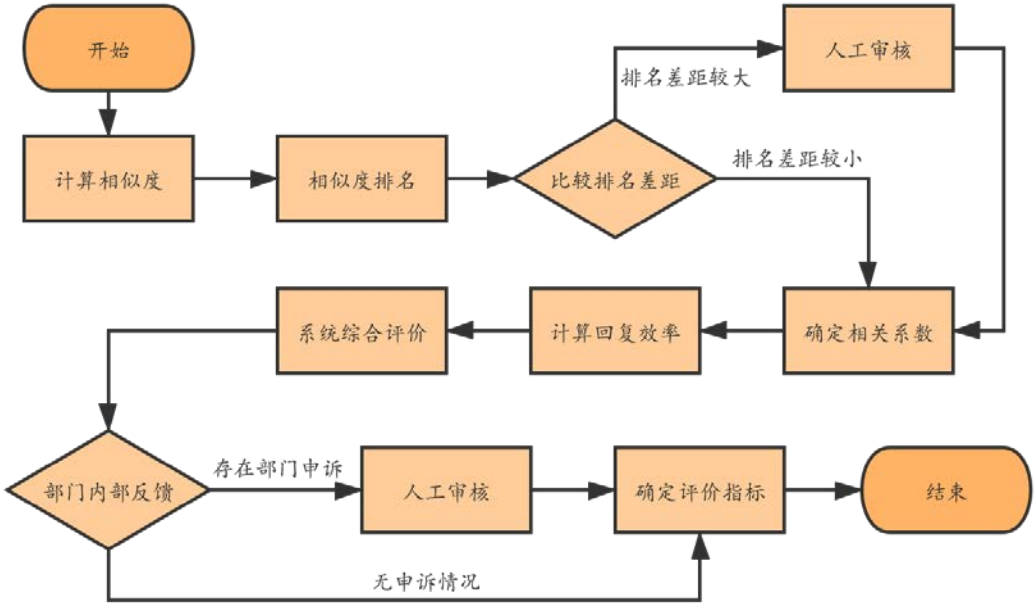

图 4 问题三流程图

附件四给出了相关部分对每条留言内容具体的答复意见，需要据此对答复意见的质量给出一套评价方案。首先需要对文本内容进行分词、构建词向量，接着根据留言内容和答复意见构建语料库，计算相应的文本相似度。回复效率也是体现答复质量的重要指标，得到相关数据之后，需要根据这些指标建立相应的评价体系。

3 符号说明

表 1 符号及说明

符号	含义
Similarity	文本相似度
TF	词频
IDF	逆文档频率指数
P_{i}	精确率
R_{i}	召回率

4 研究方案及实施

4.1 问题一

首先对附件二不同留言分类的留言条数进行统计，结果见表 2 所示。

表 2 不同留言分类的留言条数

一级分类	留言条数
城乡建设	1606
环境保护	749
交通运输	482
商贸旅游	949
卫生计生	695
教育文体	1293
劳动和社会保障	1594

由表 2 可知，主题为城乡建设的留言条数最多，主题为交通运输的留言条数最小。

4.1.1 文本评论分词

在对文本进行建模之前，首先需要对文本进行分词处理。中文分词指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。

jieba 分词支持三种分词模式：

（1）精确模式：试图将句子最精确地切开，适合文本分析；

（2）全模式：把句子中所有的可以成词的词语都扫描出来, 速度非常快，但是不能解决歧义；

（3）搜索引擎模式：在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

由于留言内容属于长文本，留言主题属于短文本，可以高度概括反映的内容，因此群众留言分类数据采用留言主题。本文使用 python 提供的 jieba 库，采用精确分词模式，分别对验证集和训练集实现了中文文本分词。分词结果示例：

分词前：A 市体育运动场地何时可以真正开放给市民用
分词后：A 市 体育运动 场地 何时 可以 真正 开放 给 市民 用

从上述结果可以看出，已经将留言主题文本切分成一个个的词语。

4.1.2 构建词频向量

在对自然语言进行分词处理后，需要构建词表，里面包含出现在任何留言主题中的所有

词，对于每条留言主题，计算词表中每个单词在该文档中的出现频次，但是以计数特征文本向量化存在明显的不足之处，以部分文本内容举例，具体留言主题内容如表 3 所示：

表 3 部分示例留言主题分词结果

留言编码	留言主题
1309	A6 区润 和 紫 郡 用电 的 问题 能 不能 解决
118617	反映 K6 县 公交车 监控 的 有关 问题
111335	K4 县 农村信用 合作 联社 208 户 合伙 建房 工程 招投标 有 问题
46322	请求 政府 加强 整顿 C4 市金 薮 乡镇 区 问题

对选取的四条文本数据进行词频统计，词频统计结果见表 4。

表 4 部分示例文本的词频向量

语料库：['208', 'a6', 'c4', 'k4', 'k6', '不能', '乡镇', '公交车', '农村信用', '加强', '区润', '反映', '合伙', '合作', '工程', '市金', '建房', '招投标', '政府', '整顿', '有关', '用电', '监控', '联社', '解决', '请求', '问题']

词频向量：[[0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1]

[0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1]

[1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 1 0 0 1]

[0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 1]]

如果以词频统计结果得到每条留言的特征向量，以此进行留言内容分类，存在一个明显的问题。比如留言编号为 118617 的文本内容，“监控”、“公交车”各出现 1 次，而“问题”出现了 1 次。单从词频统计结果来看，“监控”、“公交车”和“问题”三个特征对该留言主题的重要程度相同。但观察四条留言主题的特征词，可以发现“问题”是一个非常普遍的词，在 4 条留言主题中都出现了，因此它在四个短文本区分度最低。如果采用以计数为特征的向量化无法反应这一点。因此需要进一步的预处理来反应文本的这个特征，即 TF-IDF 预处理。

4.1.3 计算 TF-IDF

如果一个单词在某个特定文档中经常出现，但在许多文档中却不常出现，那么这个单词很可能是对文档内容的很好描述。IDF 反应了一个词在所有文本中出现的频率，如果一个词在很多的文本中出现，那么它的 IDF 值应该低，比如上文中的“问题”。而反过来如果一个词在比较少的文本中出现，那么它的 IDF 值应该高。从上述定义可以看出：当一个词在文档频率越高并且普遍度低，其 TF-IDF 值越高。其计算公式如下所示：

$$TF = \frac{\text{某个词在段落中的出现次数}}{\text{文章出现次数最多的词语的次数}}$$

$$IDF = \log(\frac{\text{语料库中段落总数}}{\text{包含该词的段落词}})$$

计算得出：

$$TFIDF = TF * IDF$$

其中 N 是训练集中的文档数量，N_w 是训练集中出现单词 w 的文档数量，tf 是单词 w 在查询文档 d 中出现的次数。IDF 计算过程用到的语料库采用 python 自带的语料库，部分语料库内的词语如图 5 所示。TF-IDF 预处理结果见表 5。

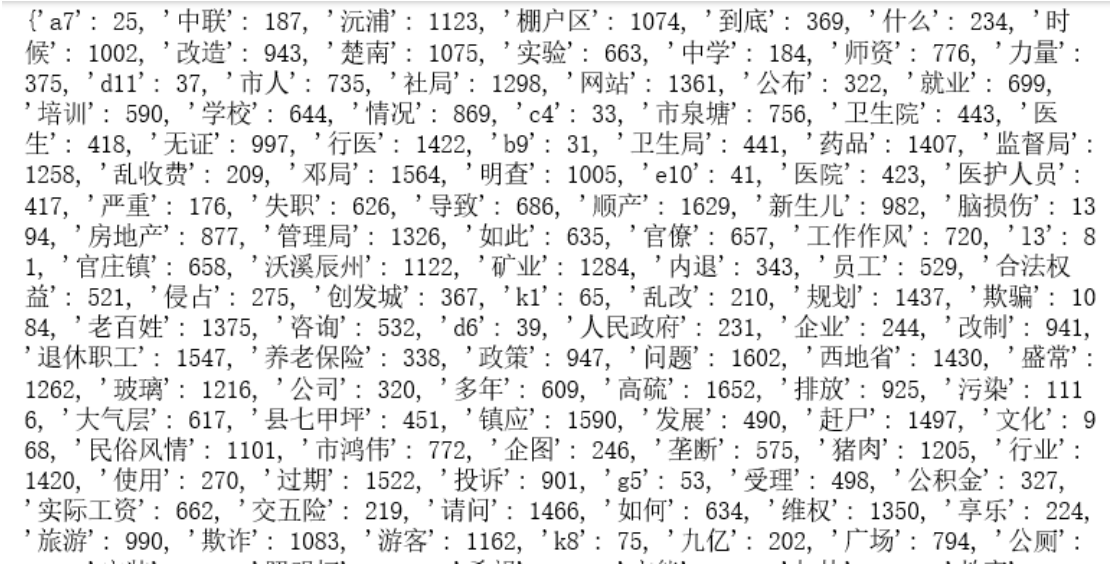

图 5 部分语料库

表 5 TF-IDF 空间向量

(0，10510)	0.16898487492378259
(0，10382)	0.36640532773767853
(0，10001)	0.26301683181610896
(0，8332)	0.3832753488628407
(0，6831)	0.19446624108866506
(0，6773)	0.2816576134332746
(0，4773)	0.3832753488628407
(0，4744)	0.3832753488628407
(0，4666)	0.20680819920246402
(0，2439)	0.2816576134332746
(0，977)	0.2967568740664334
(1，10698)	0.45902293283006723
(1，10509)	0.29572478569087335
(1，9114)	0.45902293283006723
(1，8429)	0.4042797445424842
(1，8228)	0.42448382653406663
(1，6727)	0.3840756625509019
(2，10588)	0.3081470498732718
(2，9825)	0.6027225586666108
(2，8503)	0.5355933982128067

部分 TF-IDF 向量结果见表 5，(0，10510)表示词向量中第 10510 个单词在第一行文本中的权重占比为 0.16898，(0，10382)表示词向量中第 10382 个单词在第一行文本中的权重占比同样为 0.3664，其余结果类似。

4.1.4 停用词过滤

停止词是分词过程中，我们不需要作为结果的词，比如表 4 中 A6、K4 和 C4 等词。这些词在统计词频的时候意义不大，且会增加噪音，需要删除没有信息量的单词，舍弃出现次数太多以至于没有信息量的词。本文使用特定语言的停用词列表，分别对训练集和验证集进行停用词过滤。本文在计算 TF-IDF 时已进行了停用词过滤，过程和代码不再重复给出。

4.1.5 集成模型建模

为了防止数据泄露（Data Leakage），本文采用“留一法”分割训练集。一般来说，训练集用来估计模型中的参数，使模型能够反映现实，进而预测未来或其他未知的信息，而验证集用来评估模型的预测性能。本文将附件二数据按 8:2 的比例划分训练集和验证集，其中随机抽取训练集留言条数为 7368 条，验证集为 1842 条。构建词向量空间后需要对其建模，最终对留言主题进行分类。常见的属于有监督算法的分类模型有 KNN 算法、决策树、随机森林、SVM 和 XGBoost 等。

由于单个机器学习模型所能解决的问题有限，泛化能力差，为了进一步提高模型的泛化能力，本文采用 Stacking 算法进行集成模型。Stacking 指将多种分类器组合在一起来提高预测性能的一种集成学习框架。流程见图 6。

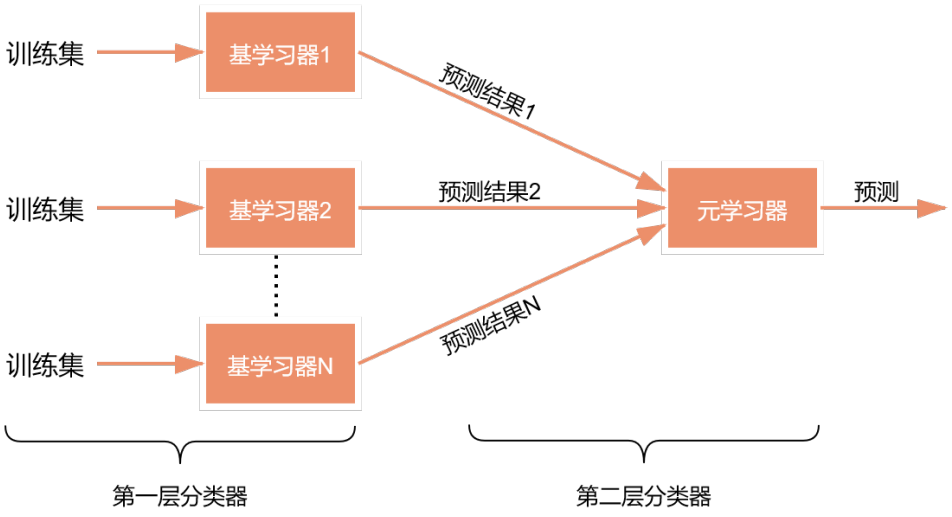

图 6　Stacking 流程图

本文中样本数据为 7368 条，预测数据为 1842 条。将样本数据均分为 5 份，每一份之间都是不重复的，其中 1 份当做验证数据集，其中 4 份当做训练数据集。

本文初级学习器采用朴素贝叶斯、逻辑回归、随机森林、SVM 支持向量机、XGBoost 等

个体学习器，现在以朴素贝叶斯模型为例详细说明 Stacing 算法第一层的过程。首先针对 10 折的训练数据分别进行学习，利用对应的验证集数据进行预测，得到 10 份预测结果。10 折数据集全部学习完毕后，每个训练样本都对应有一个结果，将这个结果序列定义为训练数据集的特征。接着对预测数据进行预测，由于每个预测样本均要进行 10 次预测，因此对这 10 次预测结果取平均值，得到的序列定义为预测数据集的特征。

5 个模型分别按照上述步骤进行预测，得到训练数据集的新的 5 个特征。同样的，预测数据集得到了同样的 5 个特征，以及初级学习器下不同模型总的精准率、召回率和 F1-Score，具体见图 7。其中，精准度为正确的正例与所有正例之比，召回率为正确的正例与正确的正例和错误的负例和的比。F1-Score 计算公式为：

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

其中 P_i 为第 i 类的查准率，R_i 为第 i 类的查全率。

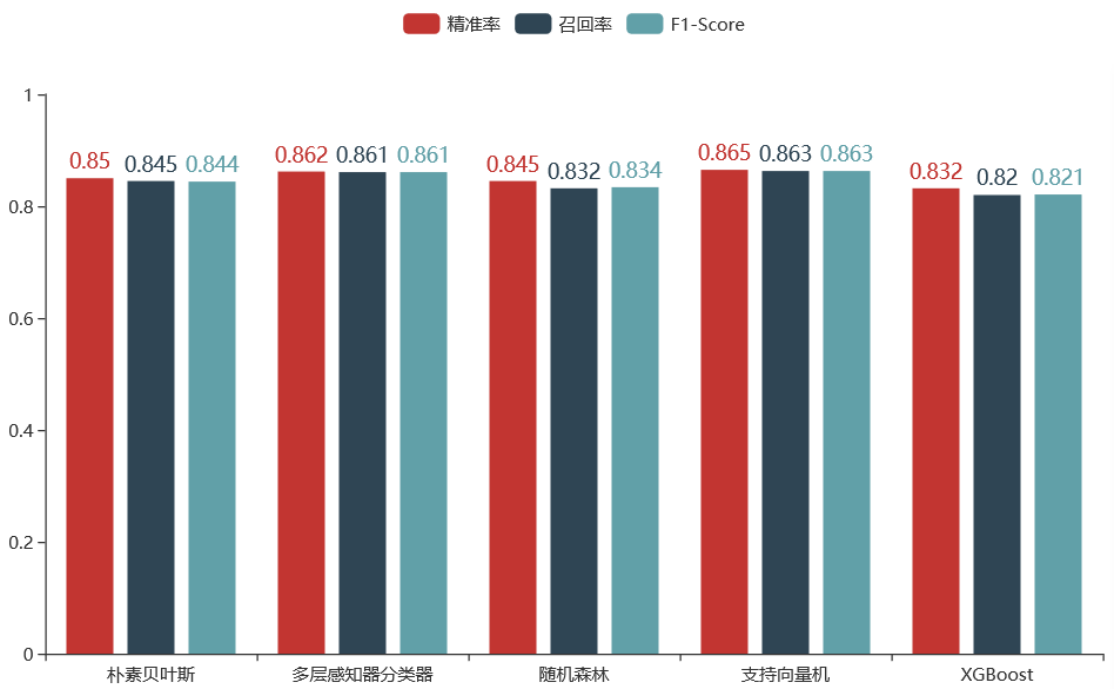

图 7 不同初级学习器下的模型效果

由图 7 可以看出，支持向量机模型下 F1-Score 值最高，为 0.863，精准率和召回率分别为 0.865 和 0.863。

次级学习器，即用于结合的学习器，选取 Logistic 模型。将上一层得到的新的特征作为训练数据集的输入，进行学习。接着，对由新特征形成的预测数据集进行预测，从而得到最终的结果。留言类别分别为“交通运输”、“劳动和社会保障”、“卫生计生”、“商贸旅游”、“城乡建设”、“教育文体”和“环境保护”七类。

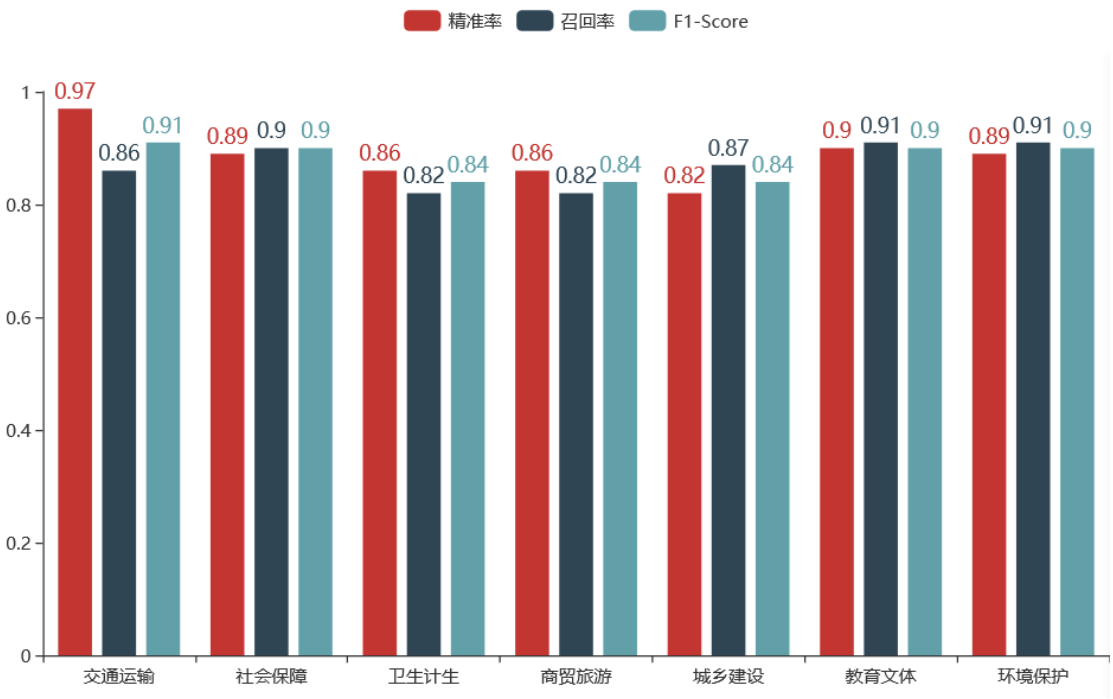

图 8 不同类别的预测结果

不同类别的准确率、召回率和 F1-Score 值见图 8，交通运输类的 F1-Score 值最高，达到了 0.91。模型总体 F1-Score 为 0.876，可以看到 Stacking 模型预测效果较好。

4.2 问题二

4.2.1 文本聚类

在对留言内容进行分类之前，首先需要对附件三的留言详情内容进行分词，将分词结果构建生成词向量，过滤停用词，最终得到 39278 个特征词。同时对特征词进行 TF-IDF 预处理，得到特征词的空间坐标和相应权重。

针对附件三的词向量空间采用 K-means 方法对学生的消费习惯划分类型，主要流程如下：

（1）确定好样本空间

（2）选择 K 个中心点（随机选择 K 行）；

（3）把每个数据点分配到离它最近的中心点；

（4）重新计算每类中的点到该类中心点距离的平均值；

（5）分配每个数据到它最近的中心点；

（6）重复步骤；

（7）直到所有的观测值不再被分配或是达到最大的迭代次数。

根据得到的 TF-IDF 向量空间模型进行 K-means 聚类，首先根据手肘法确定聚类个数，选取曲率最高时的 K 值。运行结果见图 9。

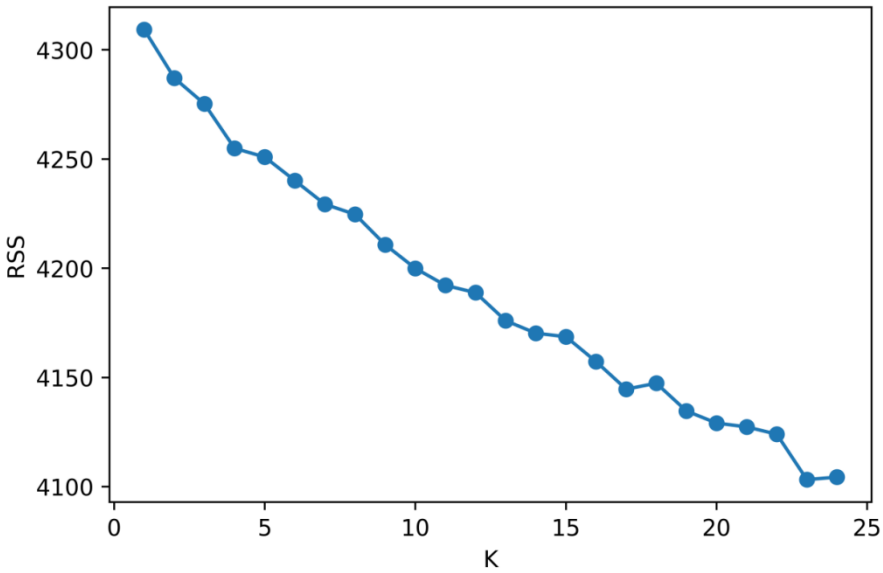

图 9 手肘法计算最佳 K 值

当聚类个数小于 20 时，SSE 下降速度非常快，K 值超过 20 后再增加 K 所得到的 SSE 趋于平缓，即 K 值为 20 时曲率最高，说明 K 为 20 时聚类效果最佳，因此聚类个数设定为 20。最终得到 20 类留言主题，相应的留言条数如表 20 所示：

表 6　K-Means 聚类结果

分类编号	留言条数	分类编号	留言条数	分类编号	留言条数	分类编号	留言条数
1	50	6	123	11	190	16	41
2	2739	7	55	12	54	17	114
3	54	8	118	13	77	18	111
4	60	9	144	14	65	19	45
5	111	10	50	15	77	20	56

4.2.2 计算热度

（1）热度预算

我们需要从这些留言内容中根据一些指标得到留言的热点问题，本文根据点赞数和反对数设定热度指标，为防止出现点赞数和反对数均为 0，热度指标为 0 的情况，每条留言根据“热度=点赞数＋反对数＋1”的公式计算热度。举例来说，如果反对数和赞数为 0，那么该评论热度为 1。利用 Python 计算每组类别下的热度指标，具体见表 7。

表 7 不同类别下的热度预算

分类编号	热度	分类编号	热度	分类编号	热度	分类编号	热度
1	76	6	147	11	325	16	57
2	13553	7	110	12	212	17	207

3	117	8	493	13	374	18	212
4	130	9	281	14	131	19	169
5	275	10	109	15	145	20	225

本文根据 20 个类别的留言内容制作词云，以期得到不同类别留言内容的关键词，更清晰快速地掌握主要内容。“词云”是通过形成“关键词云层”或“关键词渲染”，对网络文本中出现频率较高的“关键词”的视觉上的突出。本文根据得到的 20 类文本分别绘制词云。此处给出热度前四的词云结果。

图 10 第 2 类留言主题词云

图 11 第 8 类留言主题词云

图 12 第 13 类留言主题词云结果

图 13 第 11 类留言主题词云

通过词云图，可以过滤掉大量重要性比较低的文本信息。由图 10 可以看出热度第一留言主题出现频率较高的词为“业主”、“物业”和“开发商”等；热度第二的出现频率较高的词为“公园”、“项目”和“规划”等，说明第 8 类留言内容主要是围绕公园内的项目规划情况；热度第三的出现频率较高的词为“设置”、“不合理”和“红绿灯”等，表示第 13 类留言主要是针对不合理政策提出的相应的意见，且关于路口红绿灯错乱问题反馈较多；热度第四的出现频率较高的词为“教师”、“教育局”和“招聘”等，说明第 11 类留言主要

是关于教师的一些建议，比如适当提高教师收入或补贴、教师招聘考试的相关通知等。

通过上文，对初步 K-means 聚类分类结果有了基本了解，但由表 7 可知，第 2 类样本个数过多，远远大于其他几类。通过观察相关词云和留言详情，可以发现第 2 类留言内容反映的现象较为混杂，分类效果较差，因此需要对分类结果作进一步处理。

（2）提取关键词文本

根据 20 类的词云结果和相关留言详情，选取 20 类文本中最重要的 2-5 个关键词，对每类的留言内容进行筛选，表 8 为 20 类留言选取的关键词。

表 8 20 类留言选取的关键词

第一类留言：车贷 58
第二类留言：五矿 配套 金毛
第三类留言：国际 违规 保利 开发商 中海
第四类留言：规划 公园
第五类留言：新城 搅拌站 丽发
第六类留言：噪音 油烟 麻将馆 魅力
第七类留言：二期 宣传 诈骗
第八类留言：拖欠 工资 噪音 污染
第九类留言：茶场 拆迁 国道 西湖
第十类留言：中学 补课 收费
第十一类留言：旧城 提质 凉塘路
第十二类留言：时代 国家 溪湖 城市
第十三类留言：有限公司 拖欠
第十四类留言：大道 红绿灯 路口
第十五类留言：施工 工地 夜间 噪音
第十六类留言：公交车 线路 增加
第十七类留言：人才 购房 补贴
第十八类留言：安置 拆迁
第十九类留言：滨河 景园 车位 捆绑
第二十类留言：碧桂园 壹号

根据表 8 所示的关键词分别对每类文本内容进行筛选，如果该分类下某条留言主题内容文本分词结果包含这些特定关键词，则保留，反之删除。

（3）重新计算热度

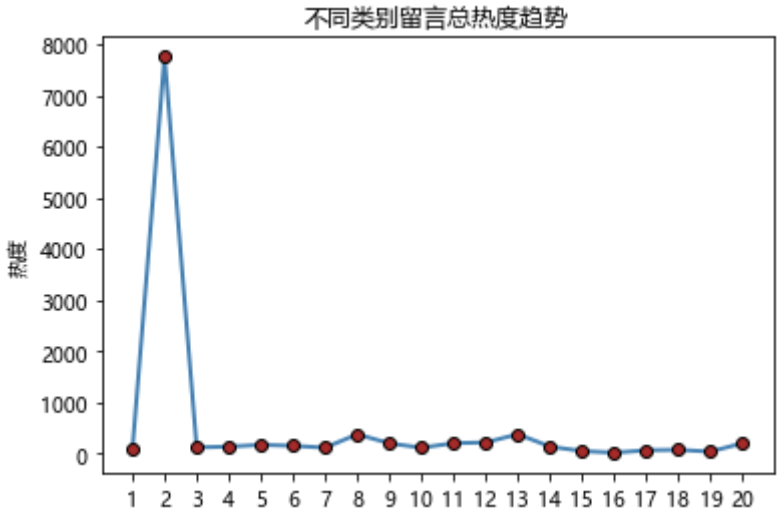

图 14 不同类别留言总热度趋势

第二类留言热度最高，达到了 7758，第 16 类留言热度仅为 12，其余类别留言热度波动幅度不大。根据每类热度值大小进行排序，保留热度前五的留言文本内容。

（4）调整分类结果

重新计算每类留言条数，预分类和重新分类留言条数对比折线图结果见图 15。

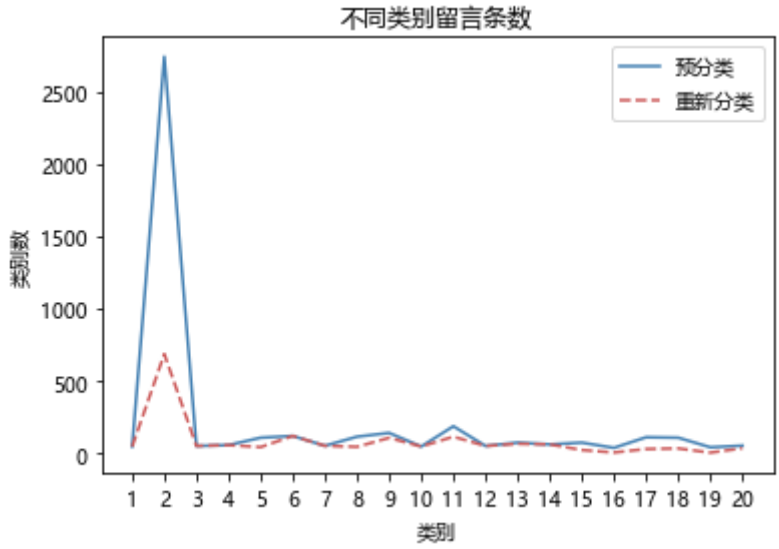

图 15 不同类别留言条数

由折线图可以更直观地看到经过调整后的分类结果有了较大改善。第二类留言条数仍然是 20 类留言类别中条数最多的，但通过观察每一类的具体留言内容，每类留言反映的主题基本一致。

4.3 问题三

针对附件 4 相关部门对留言的答复意见，本文将从答复的相关性、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，主要从回复内容与留言内容的相关性、回复效率两方面对答复的质量进行评估，建立综合评价体系，将留言内容划分为高质量回复、中质

量回复和低质量回复。

4.3.1 答复相关性

首先对文本内容进行预处理，对每条留言主题、留言详情和答复意见分别进行分词，去除停用词。示例结果见表 9。

表 9 部分示例分词结果

留言编号：	2549
答复意见分词：	现将 网友 平台 问政 西地省 栏目 胡华衡 书记 留言 A2 区景蓉 花苑 物业管理 有 问题 调查核实 情况 网友 答复 您好 感谢您 信任 支持 平台 栏目 胡华衡 书记 留言 A2 区景蓉 花苑 物业管理 有 问题 情况 收悉 现将 情况 答复 经 来信 小区 停车 收费 问题 景蓉华苑 业委会 2019 10 日至 27 日以 意见 收集 方式 业主大会 经 业委会 统计 超过 三分之二 业主 同意 收取 停车 管理费 业主大会 结束 业委会 业主 提出 意见 建议 疏理 归纳 反馈 业委会 制定 停车 收费 标准 高于 周边 小区 价格 来信 物业公司 去留 问题 辖区 桂花 坪 街道 牵头 组织 社区 物业公司 业主 委员会 业主 代表 会议 区 住房 城乡 建设局 参加 会议 综合 意见 辖区 桂花 坪 街道 区 住房 城乡 建设局 业委会 依法 依规 业主大会 业主大会 表决 执行 程序 感谢 您 我区 理解 关心 2019
留言主题分词：	A2 区景蓉华苑 物业管理 有 问题
留言详情分词：	2019 位于 A 市 A2 区 桂花 坪 街道 A2 区 公安分局 宿舍区 景蓉华苑 乱象 小区 物业公司 美顺 物业 扬言 退出 小区 小区 水电 改造 物业公司 高昂 水电费 收取 原 水电 小区 买 水 4.23 一吨 电 0.64 一度 征收 小区 停车费 增加收入 小区 业委会 不知 处于 何种 理由 物业公司 一再 挽留 业主 提出 新 应聘 物业公司 以交 20 万 保证金 提高 收费 苛刻 条件 拒之门外 业委会 未 业主大会 情况 制定 高昂 收费 方案 业主 投票 投票 采用 投票箱 制定 表格 物业公司 人员 这一 利害关系 机构 负责 组织 投票 业主 隐私权 保护 投 反对票 业主 领导 做 方式 改变 同意 票 投票 何来 公平 公正 公开 面对 公安干警 采用 方式 投票 合法性

对留言主题、留言详情和回复内容进行分词后，将分词后的短语，映射到向量空间，形成文本中文字和向量数据的映射关系。由于相关部门针对每条民众留言均给出了详尽的答复意见，本文根据有关部门的留言回复情况，分别计算每条留言的回复内容和留言主题、留言详情之间的文本相似度。常见的文本相似度有余弦相似度、TextRank 算法和 Jaccard 相似系

数。

首先根据词频统计向量，利用余弦相似度方法计算文本之间的相关系数。假定 a 和 b 时 n 维向量，向量 a 和向量 b 的余弦计算公式如下：

$$\cos(\theta)=\frac{a \bullet b}{\|a\| \times \|b\|}=\frac{\sum_{\mathrm{i}=1}^{n}(x_1, y_1)}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}}$$

两个句子的相似度计算的步骤是：

（1）通过中文分词，把完整的句子根据分词算法分为独立的词集合

（2）求出两个词集合的并集

（3）计算各自词集的词频并把词频向量化

（4）带入向量计算模型求出文本相似度

示例结果见表 10。

表 10 部分示例的词频向量

留言编号：	2549
语料库：	['A2 区景蓉华苑 物业管理 有 问题 ', '现将 网友 平台 问政 西地省 栏目 胡华衡 书记 留言 A2 区景蓉 花苑 物业管理 有 问题 调查核实 情况 网友 答复 您好 感谢您 信任 支持 平台 栏目 胡华衡 书记 留言 A2 区景蓉 花苑 物业管理 有 问题 情况 收悉 现将 情况 答复 经 来信 小区 停车 收费 问题 景蓉华苑 业委会 2019 10 日至 27 日以 意见 收集 方式 业主大会 经 业委会 统计 超过 三分之二 业主 同意 收取 停车 管理费 业主大会 结束 业委会 业主 提出 意见 建议 疏理 归纳 反馈 业委会 制定 停车 收费 标准 高于 周边 小区 价格 来信 物业公司 去留 问题 辖区 桂花 坪 街道 牵头 组织 社区 物业公司 业主 委员会 业主 代表 会议 区 住房 城乡 建设局 参加 会议 综合 意见 辖区 桂花 坪 街道 区 住房 城乡 建设局 业委会 依法 依规 业主大会 业主大会 表决 执行 程序 感谢您 我区 理解 关心 2019 ']
留言主题词频向量：	[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
回复内容词频向量：	[1 2 1 2 1 4 4 5 2 1 1 2 2 1 1 1 3 1 1 2 2 0 1 1 1 1 1 2 2 1 2 2 1 2 1 1 3 3 2 1 1 1 1 1 1 2 1 1 1 1 1 2 2 1 2 2 2 2 1 2 1 2 1 1 1 2 1 1 2 1 1 1 2 2 2 2 1 1 1 1 2 1 4 1]

根据留言编号为 2549 的留言主题词频向量和回复内容的词频向量计算两个短文本之间的相似程度，计算过程如下：

$$\cos(\theta)=\frac{1\times2+1\times2+1\times2+1\times4}{\sqrt{1^2+1^2+1^2+1^2+1^2}\times\sqrt{1^2+2^2+1^2+\mathrm{L}\ +1^2+4^2+1^2}}=0.2758$$

但无论是利用词频向量还是 TF-IDF 计算余弦相似度，存在同样的问题。如果两段文本很长，文本的特征向量会特别的多，用余弦来计算会导致计算量很大，时间成本太高。由于留言详情和回复详情文本较长，且 TextRank 计算公式中分母使用对数可以抵消长句子在相似度计算上的优势(长句子包含相同单词的可能性更高)。

因此本文根据 TextRank 算法来计算句子之间的文本相似度。计算公式如下：

$$Simi\mathit{larity}(S_i,S_j)=\frac{|\{w_k\mid w_k\in S_i\ \&\ w_k\in S_j\}|}{\log(|S_{\mathrm{i}}|)+\log(|S_{\mathrm{j}}|)}$$

其中，S_{i} 表示第 i 个句子，W_{k} 表示句子中第 k 个单词，$|S_i|$ 表示句子中单词的个数，$|\{w_k\mid w_k\in S_i\ \&\ w_k\in S_j\}|$ 代表着同时在 S_{i} 和 S_{j} 中出现的单词。

同样以表 11 为例，留言编号为 2549 的留言主题和回复内容的相似系数计算过程如下：

表 11 部分示例留言主题和回复内容的相同词语

留言编号：2549
相同词：A2 物业管理 有 问题

$$Simi\mathit{larity}(S_1,S_2)=\frac{5}{\log(1+1+1+1+1)\times\log(1+2+1+\mathrm{L}\ +1+4+1)}=0.6168$$

利用 TextRank 算法得到所有留言主题、留言详情与回复内容之间的文本的相似度，部分计算结果见表 12：

表 12 部分示例 TextRank 算法相似度计算结果

留言编号	留言主题与回复内容的相似度	留言详情与回复内容的相似度
2549	0.616842714300822	3.96093426658643
2554	0.94553164707606	0.356971829673739
2555	1.15980606099561	1.96060967262366
2557	0.757434586	1.3135717790212
2574	0.540252446307395	1.92119674711881

可以看出对于大部分留言文本，相比于留言主题与回复内容的相似度，留言详情和回复内容之间的文本相似程度更高，一部分留言文本前者的相似度更高，本文在“留言主题与回复内容的相似度”和“留言详情与回复内容的相似度”这两个变量之间选取较高的值作为与回

复内容之间的相似度值，同时根据该变量的值进行排序。通过观察每条回复内容的相似度排序，留言编号为 454 的相似度最高，有 248 条回复内容相似度为 0。

表 13 部分相似度为 0 的留言内容

留言编号：12274
留言主题：请求解决蒋垄家火车噪音问题
留言详情：尊敬的易书记 及市委主管部门：最近以来，特别是晚上，进入蒋垄家的火车大肆鸣笛，严重的影响群众的休息。晚上 22 点以后，拉笛的时间又长有多。为此，请求市委各级部门协调地方与铁路关系，解决好这一老大难问题。以前有居民反映过相关噪音问题，但都无具体解决方案。或者推卸责任，认为这是铁路问题，无法解决。为此全体居民请求市委主管部门认真践行群众路线，下决心解决这一关系群众身心的问题。 A4 区蒋垄家全体居民 2014/5/25
答复意见：网友：您好！留言已收悉

对相似度为 0 的 248 条留言的答复意见进行分析，发现大部分回复内容均和编号为 12274 的留言答复意见相似，只表达了相关部门看到了该留言，但并未为阐述如何展开具体的解决措施、何时解决该问题，没有从本质上解决网民的需求。

接着根据文本相似度指标构建答复相似度指标：相似度小于 1，认为答复的相关性非常低；相似度在 1 到 3 之间，认为答复的相关性较低；相似度在 3 到 5 之间，认为答复的相关性一般；相似度在 5 到 10 之间，认为答复的相关性较高；相似度大于 10，认为答复的相关性非常高。

4.3.2 留言回复效率

在对答复情况做评价时，回复效率同样是非常重要的。本文根据每条留言的留言时间和答复时间，两者相减，得到留言回复间隔时间率以体现回复效率的快慢。可以得到平均每条留言的处理时间为 20 天，其中留言编号为 159285 的留言处理时间最长，为 1160 天，长达三个月多的时间，有 132 条留言在相关部分在当天就对其进行回复。同样根据留言回复间隔时间构建留言回复效率指标：间隔时间小于 3 天，认为回复效率非常高；间隔时间在 3 天到 7 天之间，认为回复效率一般；间隔时间在 7 天到 30 天之间，认为回复效率一般；间隔时间在 30 天到 90 天之间，认为回复效率较低；间隔时间超过 90 天以上，认为回复效率非常低。图 16 为不同回复相似度和不同回复效率下的留言比例图。

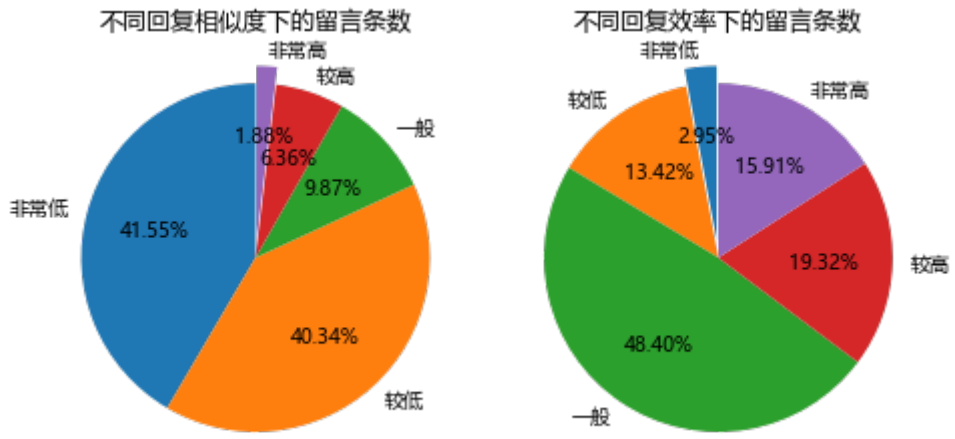

图 16 不同回复相似度、回复效率下的留言条数

从图 16 看，答复的总体相似度水平不高，大部分回复内容的文本相关度为较低或非常低。结合表 13 反映出的问题，相关部门在对留言内容进行回复时，应尽可能地保证答复意见解决了网民反映的意见或建议，而不是仅仅单方面为了提高工作效率。

从回复效率看，有 448 条留言内容在 3 天内就得到了有关部门的回复，仅有 83 条留言处理时间长达 3 个月。由于通过网民在网上留言，反映一些城乡建设或交通方面的意见或建议，这些建议具有时效性。因此在回复效率方面，有关部门需要继续加快处理留言内容的速度，以便更及时地了解民生民意。

由于计算回复效率和答复相似系数过程均通过文本分析来实现，在等级评定过程中可能会出现偏差，因此，可以抽取部分留言，对评定系统进行抽查，也可以通过内部反馈。如果抽查结果或内部反馈发现部分留言的回复效率和答复相似系数的等级评价存在问题，则对这些留言进行人工审核，人工校准。

4.3.3 构建评价体系

在得到每条留言文本内容的回复效率和文本相似度之后，需要对回复内容建立总的评价体系。本文对这两个变量等额赋权，如果回复效率或答复相关性非常低，则赋值 1，较低赋值为 2，一般赋值为 3，较高赋值为 4，非常高赋值为 5。两者相加得到总分，如果分值小于等于 4，则认为留言质量低，分值大于等于 7，则质量高，反之分值一般。计算得到回复质量低的留言条数为 1824 条，质量一般的为 544 条，质量高的为 448 条。从答复情况的总体质量来看，大部分回复的质量较差。相关部门在回复网民的留言时，不仅要做到及时回复留言，而且要对网民反映的问题有切实的解决方案。

参考文献

[1] 李航.统计学习方法[M], 清华大学出版社, 2012,3.

[2] Steven B, Ewan K & Edward L.Natural Language Processing with Python[M],2009,6

[3] 姜维.文本分析与文本挖掘[M],科学出版社,2018.

[4] GEHRKE J.Classfication and regression tress[J].Wiley Interdisciplinary Reviews:Data Mining &Knowlege Discovery,2005,1(1):14-23.

[5] 张俊博, 李健, 张宏宇. 潜在语义分析中主题数的确定方法[J].信息技术, 2016(7): 96-100.

[6] 沈斌.基于分词的中文文本相似度计算研究[D].天津天津财经大学.2010:8-10.

[7] Jan Leeuw,Sandra Pruzansky.A new computational method to fit the weighted euclidean distance model[J].Psychometrika.1978,43(4):479:490.

[8] 汪前进,施琚文档相似度量算法的研究与应用淮海工学院学报(自然科学版).2007,16(3):28-31.

[9] Christopher D,Hinrich S.Foundations of Statistical Natural Language Processing[M].The MIT Press,1996,6.

[10] 谢丽星,基于 SVM 的中文微博情感分析的研究[D].北京,11-13.