

“智慧政务”中的文本挖掘应用

摘要：随着城市智能化的发展，微信、市长信箱、阳光热线等网络问政平台逐步成为了政府了解公众意见、调适公共事务发展方向的重要渠道，各类社情民意相关的文本数量不断攀升，因此，运用自然文本语言处理和文本挖掘的方法对政民互动留言信息的研究具有重大意义。

针对问题一，基于群众留言文本内容，利用中文分词、文本向量化和文本多分类的方法进行文本挖掘，需要建立一级标签识别分类模型，自动对留言文本进行识别分类，以解决人工分类效率低下的问题。为此我们选择了常用分类模型朴素贝叶斯模型、随机森林、支持向量机以及逻辑回归模型分别构造分类器，对处理后的文本数据进行训练，最后基于分类结果的精确性和全面性的综合评估，经调整参数模型优化，根据模型在验证集性能效果表现，我们选择效果最优的 $TF-IDF + MultinomialNB$ 模型做留言文本分类模型，其 F-score 得分可达 0.8549

针对问题二，有两方面的要求，一方面需要将留言按照特定地点人物事件进行归类，首先需要进行数据预处理，除去多余的停用词和标点符号，提取出关于地点人物事件的动词名词，采用 TextRank 算法来对文本相似度进行分析，超过一定的阈值即可归为一类。

另一方面需要建立合理的热度评价体系来判断是否为热点问题。首先，每一个留言本身的热度，都可通过留言问题的反对或者点赞之类的来进行热度的量化，其次在留言归类的基础上，把同一类的留言量化到该相关里，并且热度随着留言的发表时间段的拉长进行衰减，这样就形成了同一类热点问题协同加权。

针对问题三，为了检验职能部门对群众留言的处理质量，需要针对相关部门对留言的答复意见质量给出一套评价方案，这还有利于建立高质量的智能政务问答对知识库。为解决上述问题，本文将答复意见的质量与留言内容进行比对，从答复的相关性、完整性、可解释性、时效性四个维度建立答复意见质量评价指标，通过提取这些信息作为特征集合，训练分类器，用于判断答复文本的质量。

关键词 中文分词；文本挖掘；朴素贝叶斯；TF-IDF；TextRank；协同加权

Application of Text Mining in Intelligent Government Affairs

Abstract: With the development of urban intelligence, online questioning platforms such as WeChat, mayor's mailbox, and sunshine hotline have gradually become an important channel for the government to understand public opinions and adjust the direction of public affairs. , The use of natural text language processing and text mining methods is of great significance to the study of the interactive message of government and citizens.

Aiming at problem 1, based on the content of mass message texts, using Chinese word segmentation, text vectorization and text multi-classification methods for text mining, it is necessary to establish a first-level label recognition classification model to automatically classify the message text to solve the inefficient manual classification The problem. To this end, we chose the common classification model Naive Bayes model, random forest, support vector machine and logistic regression model to construct a classifier, train the processed text data, and finally based on the accuracy and comprehensiveness of the classification results. Evaluation, optimized by adjusting the parameter model, according to the performance of the model in the verification set performance, we choose the best effect TF-IDF + MultinomialNB model as the message text classification model, and its F-score score can reach 0.8549.

For problem two, there are two requirements. On the one hand, you need to classify the messages according to the person events at specific locations. First, you need to perform data preprocessing to remove unnecessary stop words and punctuation marks, and extract verb nouns about the person events at the location , The TextRank algorithm is used to analyze the text similarity. On the other hand, it is necessary to establish a reasonable heat evaluation system to judge whether it is a hot issue. First, the enthusiasm of each message itself can be quantified by objection to the message question or likes. Second, based on the classification of the message, the same type of message can be quantified into the correlation As the posting period of the message lengthens, it decays, thus forming a collaborative weighting of hot issues of the same type.

Keywords: Chinese word segmentation; naive Bayes; TF-IDF; TextRank;

目录

一、问题背景	3
二、问题目标	3
三、文本挖掘技术简介	4
3.1 文本预处理	4
3.1.1 中文分词	4
3.1.2 停用词过滤	4
3.1.3 词性标注	4
3.2 文本特征提取方法	5
3.2.1 词频法	5
3.2.2 文档频数法	5
3.2.3 TF-IDF 特征提取	5
3.3 文本数据挖掘的关键技术概要	6
四、留言分类模型	7
4.1 模型要求解读	7
4.2 数据探索及预处理	7
4.2.1 数据描述	7
4.2.2 数据分布	8
4.2.3 数据预处理	9
4.3 模型选择与建立	11
4.3.1 分类模型	11
4.3.2 模型建立	13
4.4 模型评价与优化	15
4.4.1 模型评价	15
4.4.2 模型优化	19
五、根据文本相似度将留言进行归类	21
5.1 数据预处理流程:	21
5.2 文本相似度的计算	21
5.3 热度评价指标的建立	22
六、答复意见质量的评价方案	23
6.1 问题解读	23
6.2 答复意见质量评价指标体系构建	23
参考文献	25

一、问题背景

随着城市智能化的发展，政民互动问题的解决面临着全新的时代背景，信息时代的“数字化生存”不仅使得公众在公共服务领域表达诉求和获取服务越来越倾向于利用网络，也使得政府部门基于网络信息技术拥有了更多优化和改变公共服务方式的机会。

微信、市长信箱、阳光热线等网络问政平台逐步成为了政府了解公众意见、调适公共事务发展方向的重要渠道，各类社情民意相关的文本数量不断攀升，人工分类的分类标准不统一、分类效率较低的问题也随之暴露，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带来了极大的挑战。因此解决人工分类问题的弊端，快速、准确地分析出群众当前关心的热点事件，及时处理群众的诉求，对于政府的管理有极大的帮助。

借助大数据、云计算、人工智能等技术，可以建立基于自然语言处理技术的智慧政务系统，实现留言文本的智能分类、民生热点挖掘等功能，对提升政府的民生建设管理水平和施政效率具有极大的推动作用。

二、问题目标

本次建模目标是对源自互联网公开来源的群众问政留言记录及相关部门的答复意见进行处理，利用自然语言处理和文本挖掘的方法，达到以下三个目标：

- 1) 群众留言分类。基于群众留言文本内容，利用中文分词和文本多分类的方法进行文本挖掘，建立一级标签识别分类模型，自动对留言文本进行识别分类，以解决人工分类效率低下的问题。
- 2) 热点问题挖掘。利用语义相似度的计算和文本聚类的方法，并定义一个热度评价指标，对某一时段内群众集中反映的问题进行挖掘，以帮助相关部门更有效更准确地了解某时段某地点的热点问题。
- 3) 答复意见的评价。针对附件数据，及相关职能部门对群众留言的答复意见，建立一套包含答复的相关性、完整性、可解释性三个维度的评价方案。

三、文本挖掘技术简介

由于中文文本的特殊性，影响文本分类和文本聚类精确度的关键因素之一就是文本预处理过程。在预处理过程中，需要对中文文本进行分词处理、去停用词等操作，获取文本的关键特征词，以此提高精确度。除了文本预处理之外，在进行文本分类等进一步文本挖掘之前还需要对文本进行特征选择，进一步对分词结果进行筛选，起到文本降维的作用。

3.1 文本预处理

3.1.1 中文分词

在中文里，一句话的含义往往通过一段连续的词组进行表达，词组之间没有一个形式上的分界符将其断开。因此我们在对文本处理时，需要进行文本分词，并按照规定重新合成词序列。目前主要分为基于词典的分词算法和无词典的分词算法两种。本实验采用的是基于词典的分词算法，利用 Python 的中文分词包 jieba 进行分词。

而其算法为正向最大匹配法，这是基于词典的分词中最常用的方法之一，其基本原理为：用 Length 表示词典中词语的最大长度，正向地取长度为 Length 的字符串，在词典中查找该字符串。如果匹配成功就将该字符串作为一个词语进行切分，然后再正向移动长度为 Length 的字符继续进行查找，依次进行下去，直至切分成功。

3.1.2 停用词过滤

在对文本进行分词之后，需要对分词结果进行进一步处理。句子中有一些功能词、语气词等，这些词对文本挖掘毫无意义，称之为停用词。在特征提取过程中，停用词会导致结果出现误差，因此需要在文本预处理阶段将停用词去除。

停用词去除需要载入一个停用词表，停用词表中出现的词，会从分词结果中滤除。在对群众留言文本数据进行去除停用词时，可以手动加入一些专有领域的词，完善停用词表。

3.1.3 词性标注

除了分词和处理停用词，词性标注也是文本预处理时重要的一部分，它有利于分词后关键词的筛选。在对留言文本进行处理时，由于是针对专有领域的文本挖掘，所以得根据文本的特殊性对分词结果进一步筛选。留言文本中的语气词、感叹词等虚词对热点问题的挖掘并没有实际的意义，比较重要的是包括名词、动词、地点名等命名实体的实词，所以在去停用词后可以对分词结果进行词性标注，提取相关词性的词作为结果。

3.2 文本特征提取方法

为了识别一个文本，需要获取文本的特征。有三种常见的提取文本特征方法：词频方法、文档频数法和 TF-IDF。

3.2.1 词频法

词频即一个词在文档中出现的频率。实际上，词频较小的词其重要程度很多时候往往高于词频较大的词，所以仅仅统计词频的方法在实际中有一定的缺陷。

3.2.2 文档频数法

词的文档频数即指在所有的文档集中包含该词的文档数，不过在特征提取的时候，往往需要去除文档频数达到某一阈值或者小于某一阈值的词的文档数。文档频数的方法速度快，但是存在缺陷。假设某一少见的词在某一类文档中出现，但是由于这类文档的数量较小，因此会被去除其文档频数，导致其特征丢失。

3.2.3 TF-IDF 特征提取

为了平衡权重，有了 TF-IDF，用以评估一个词汇对于一个语料库中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 是指词频，对于在某一文档 d_j 里的词语 t_i 来说， t_i 的词频可表示为：

$$TF_{i,j} = \frac{\text{词 } t_i \text{ 在文档 } d_j \text{ 中出现的次数}}{\text{文档 } d_j \text{ 的总词数}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

其中 $n_{i,j}$ 是词语 t_i 在文档 d_j 中出现的次数，分母则是在文件 d_j 中所有词语的出现次数之和。

IDF 是指逆向文件频率，其主要思想为如果包含词语 t 的文档越少，则 IDF 越大，说明词语 t 在整个文档集层面上具有很好的类别区分能力。对于某一特定词语的 IDF，可以由总文件数除以包含该词语的文件数，再将得到的商取对数得到：

$$IDF_{i,j} = \log \left(\frac{\text{语料库的文档总数}}{\text{包含词 } t_i \text{ 的文档数}} \right) = \log \frac{|D|}{|\{j: t_i \in d_j\}|}$$

其中 $|D|$ 是语料库中所有文档总数，分母是包含词语 t_i 的所有文档数。如果该词语不在语料库中，就会导致分母为零，因此一般情况下使用以下公式

$$IDF_i = \log \frac{|D|}{1 + |\{d \in D: t \in d\}|}$$

TF-IDF 的值为：

$$TF - IDF_{i,j} = TF_{i,j} \times IDF_{i,j}$$

实际分析得出 TF-IDF 值越大，该词在文本中的重要性越高。计算文本中每个词的 TF-IDF 值，进行排序，排在前列的即为要提取的文本关键词

3.3 文本数据挖掘的关键技术概要

文本数据挖掘是指从文本数据中抽取有价值的信息和知识的过程，其关键技术主要包括以下几点：

1) 文本分类 文本分类是利用计算机对文本集合按照预先定义的分类体系或标准进行自动分类标记。文本分类是采用基于文本主题对文档按主题进行自动归类。留言文本分类模型是基于主题的应用。

2) 文本聚类 文本聚类是基于“同类的文本集相似度较大，而不同类的文本集相似度较小”理论，假设对文本集合进行有效地组织、摘要和导航，会方便人们从文本集中发现相关的信息。

3) 关联规则 关联规则是描述一个文本中某些属性同时出现的规律和模式。它的核心是将各种信息载体中的共现信息定量化的分析方法，以揭示信息的内容关联和特征项所隐含的寓意，借此可以挖掘隐含的或潜在的有用的信息。

四、留言分类模型

4.1 模型要求解读

在处理网络问政平台的群众留言时，工作人员需要先按照划分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、效率低下，差错率高等问题。故需要建立关于留言内容的分类模型，自动对留言文本进行识别分类，以解决人工分类的问题。

根据模型要求，要建立足够大的语料库以保证模型的预测分类效果，因此相较简短的留言主题，留言详情中有更多的关键词，在本节中的文本分类会主要针对留言详情上的留言文本。留言本文分类模型建立的操作流程如下图所示

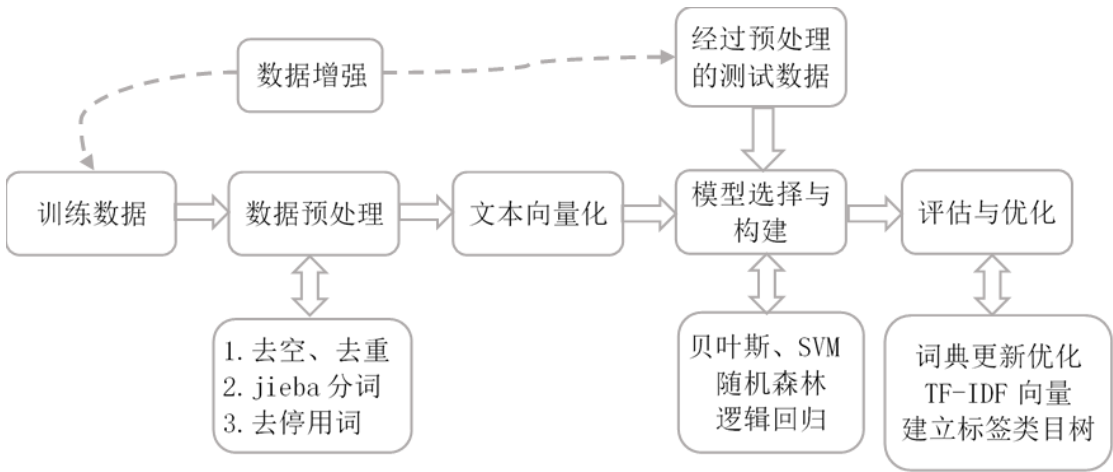


图 4-1 留言分类模型建立的操作流程

4.2 数据探索及预处理

4.2.1 数据描述

附件 1 和附件 2 都是 xlsx 文件。

附件 1 为对群众留言内容分类的三级标签体系，一级标签共有 15 个类别，分别为“城乡建设”、“党务政务”、“国土资源”、“环境保护”、“纪检监察”、“交通运输”、“教育文体”、“经济管理”、“科技与信息产业”、“劳动和社会保障”、“民政”、“农村农业”、“商贸旅游”、“卫生计生”、“政法”。三级标签之间是从属和包含的关系。附件 1 总共有 517 条记录，为文本格式，其数据结构如下：

	A	B	C
1	一级分类	二级分类	三级分类
2	城乡建设	安全生产	事故处理
3	城乡建设	安全生产	安全生产管理
4	城乡建设	安全生产	安全隐患
5	城乡建设	城市建设和市政管理	园林绿化环卫
6	城乡建设	城市建设和市政管理	城管执法
7	城乡建设	城市建设和市政管理	居民服务设施
8	城乡建设	城市建设和市政管理	城市公共设施
9	城乡建设	城市建设和市政管理	其他
10	城乡建设	城市建设和市政管理	公共汽车

图 4-2 附件 1 的数据结构示意图

附件 2 中记录的是留言内容的样本，分为六个属性：留言编号、留言用户、留言主题、留言时间、留言详情和所属的类别的一级标签。样本记录共有 9210 条，其数据结构如下所示：

	A	B	C	D	E	F
1	留言编号	留言用户	留言主题	留言时间	留言详情	一级标签
2	24	A00074011	市西湖建筑集团占道施工有安全隐患	2020/1/6 12:09:38	目施工围墙内。每天尤其上下班	城乡建设
3	37	U0008473	水一方大厦人为烂尾多年，安全隐患	2020/1/4 11:17:46	户栏围着，不但占用人行道，而	城乡建设
4	83	A00063999	投诉A市A1区苑物业违规收停车费	2019/12/30 17:06:14	业主已多次向物业和社区提出诉	城乡建设
5	303	U0007137	蔡锷南路A2区华庭楼顶水箱长年	2019/12/6 14:40:14	的用品，霉是一种强致癌物，我	城乡建设
6	319	U0007137	A1区A2区华庭自来水好大一股霉味	2019/12/5 11:17:22	的用品，霉是一种强致癌物，我	城乡建设
7	379	A00016773	投诉A市盛世耀凯小区物业无故停水	2019/11/28 9:08:38	至于物业不是为业主服务的，而	城乡建设
8	382	U0005806	咨询A市楼盘集中供暖一事	2019/11/27 17:14:11	地处月亮岛片区近年规划有楚江	城乡建设
9	445	A00019209	梓坡西路可可小城长期停水得不	2019/11/19 22:39:36	位寻求帮助至今没有找到具体停	城乡建设

图 4-3 附件 2 的数据结构示意图

4.2.2 数据分布

模型要求根据附件 2 给出的数据，建立关于留言内容的一级标签分类模型，是关于文本多分类的问题。首先对附件 2 中的 9210 条原始数据进行数据探索，发现数据中并未存在空值，进一步查看各类别标签留言文本的分布情况。

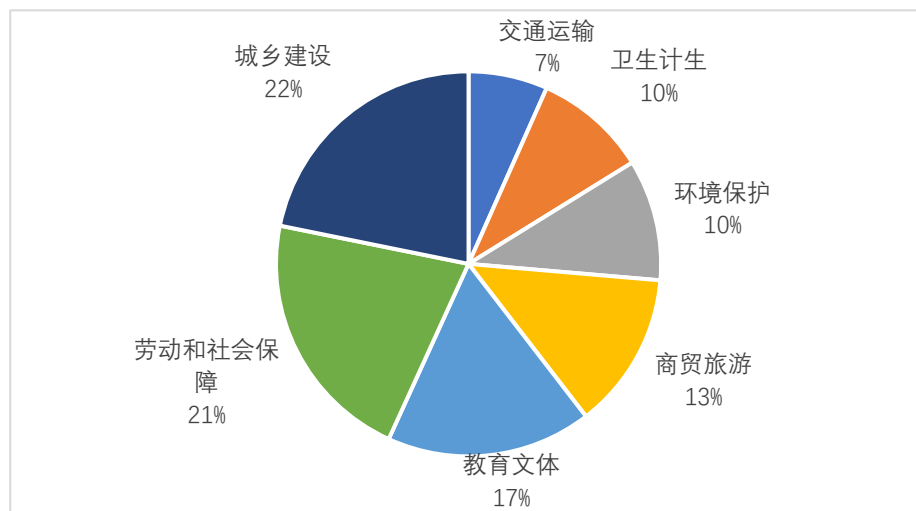


图 4-4 各类别标签留言的分布情况

附件 2 中的一级标签均属于附件 1 中的标签体系,但附件 2 中的数据并未包含全部一级标签的样本。

4.2.3 数据预处理

(1) 去空、去重

导入数据到 Python 中，查看数据的存储格式。

	留言编号	留言用户	留言主题	留言时间	留言详情	一级标签
0	24	A00074011	A市西湖建筑集团占道施工有安全隐患	2020/1/6 12:09:38	\n\n\n\n\n\n\n\n\n\nA3区大道西行便道，未管所路口至加油站路段，...	城乡建设
1	37	U0008473	A市在水一方大厦人为烂尾多年，安全隐患严重	2020/1/4 11:17:46	\n\n\n\n\n\n\n\n\n\n位于书院路主干道的在水一方大厦一楼至四楼人为...	城乡建设
2	83	A00063999	投诉A市A1区苑物业违规收停车费	2019/12/30 17:06:14	\n\n\n\n\n\n\n\n\n\n尊敬的领导：A1区苑小区位于A1区火炬路，小...	城乡建设
3	303	U0007137	A1区蔡锷南路A2区华庭楼顶水箱长年不洗	2019/12/6 14:40:14	\n\n\n\n\n\n\n\n\n\nA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设
4	319	U0007137	A1区A2区华庭自来水好大一股霉味	2019/12/5 11:17:22	\n\n\n\n\n\n\n\n\n\nA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设

图 4-4 数据集的数据结构

查看数据，可以看到留言详情一系列的文本数据存在许多空白符和换行符，并且存在重复文本内容，为了清理这些无关内容，我们进行了如下操作：

- 利用 `dropna` 函数去缺失值和空值。并无缺失值和空值，处理后仍有 9210 条数据；
- 抽取留言详情列的文本数据进行去重处理，利用 `drop_duplicates` 函数，处理后为 9052 条数据。
- 利用 `strip` 和 `translate` 函数去除文本中首尾以及存在的空白符和换行符，如“`\n`”、“`\t`”以及“`\xa0`”、“`\u3000`”的字符串。

处理后的数据如下所示:

0 A3区大道西行便道，未管所路口至加油站路段，人行道包括路灯杆，被圈西湖建筑集团燕子山安置房项...
1 位于书院路主干道的在水一方大厦一至四楼用于拆除水、电等设施后，烂尾多年，用护栏围着，不但占...
2 尊敬的领导：A1区苑小区位于A1区火炬路，小区物业A市程明物业服务有限公司，未经小区业主同意...
3 A1区A2区华庭小区高层为二次供水，楼顶水箱长年不洗，现在自来水龙头的水严重霉味，大家都闻道...
5 我在2015年购买了盛世耀凯小区17栋3楼，4楼两层共计2千平方，一直以来我们按时足额缴纳物...
Name: 留言详情, dtype: object

图 4-5 去空、去重处理后的数据

(2) jieba 分词

对于清理后的留言文本，需要对其进行分词操作，在分词前考虑分词工具分词会将某些专有名词错误划分，所以本文通过观察文本分词后的结果，自定义一个用户词词典，将分词工具的未登录词加入用户词词典中，可以先让分词工具对用户词词典进行分析，再使用分词工具自带词典进行分词，减少出现错误分词的现象。

```
content = df_mes.content.values.tolist()    #将留言内容取出，存为列表
jieba.load_userdict('.../userdict.txt')    #将自定义的词典加入
content_S = []                             #指定一个空列表，准备存储分词结果
for line in content:
    current_segment = jieba.lcut(line)      #对每句话进行分词，默认为精确模式
    if len(current_segment) > 1 and current_segment != '\r\n':
        content_S.append(current_segment)
#如果分词结果长度大于1，且不是一个换行符，那么添加到 content_S 列表中
df_content=pd.DataFrame({'content_S':content_S})    #列表转为 DataFrame 格式
```

分词过程即为上述代码所示，首先提取出所有的留言内容，以列表形式存储。并建立一个空列表，用来存储分词后的结果。建立循环，取出留言列表中的每句话，分别做分词操作，当分词得到的结果长度大于1，且不属于换行符，即存为一个成功分词的结果。最后将每一条留言的分词结果存入 DataFrame 格式。部分分词结果如下：

	content_S
0	[A3, 区, 大道, 西行, 便, 道, , , 未管, 所, 路口, 至, 加油站, 路段...
1	[位于, 书院, 路, 主干道, 的, 在水一方, 大厦, 一楼, 至, 四楼, 人为, 拆...
2	[尊敬, 的, 领导, :, A1, 区苑, 小区, 位于, A1, 区, 火炬, 路, , ...
3	[A1, 区, A2, 区华庭, 小区, 高层, 为, 二次, 供水, , , 楼顶, 水箱, ...
4	[我, 在, 2015, 年, 购买, 了, 盛世, 耀凯, 小区, 17, 栋, 3, 楼...

图 4-6 分词结果示意图

(3) 去停用词

上述分词结果中可以看到，有大量的标点及无意义的字词，对后续的分析会造成很大的影响，因此需要进行停用词过滤。这里我们使用四川大学机器智能实验室发布的含 976 个中文停用词的词库。

```
def drop_stopwords(contents, stopwords):
    contents_clean = []    #存储清理后的词
    for line in contents:
        line_clean = []
```

```

for word in line:
    if word in stopwords:    #出现在停用词，则不处理
        continue
    line_clean.append(word)    #需要保留的词按列加入到 line_clean
contents_clean.append(line_clean)
return contents_clean

```

上述为停用词过滤的代码。对分好词的列表进行遍历，如果该词出现在停用词表中，就删除该词，但是该操作会有点繁杂，而上述代码则是通过提取没有出现在停用词表中的词语，从而达到将停用词删除的效果。列表 `contents_clean` 即为清理后的非停用词记录。转为 Dataframe 格式查看，效果如下

	contents_clean
0	[A3, 区, 大道, 西行, 道, 未管, 路口, 加油站, 路段, 人行道, 包括, 路...
1	[位于, 书院, 路, 主干道, 在水一方, 大厦, 一楼, 四楼, 人为, 拆除, 水, ...
2	[尊敬, 领导, A1, 区苑, 小区, 位于, A1, 区, 火炬, 路, 小区, 物业, ...
3	[A1, 区, A2, 区华庭, 小区, 高层, 二次, 供水, 楼顶, 水箱, 长年, 不...
4	[2015, 购买, 盛世, 耀凯, 小区, 17, 栋, 楼, 楼, 两层, 共计, 平方...

图 4-7 去停用词结果展示

4.3 模型选择与建立

4.3.1 分类模型

传统的分类算法有随机森林、支持向量机、逻辑回归、朴素贝叶斯等。而他们应用于文本分类中时，其基本原理如下：

• 随机森林

随机森林就是指通过多个不同的决策树进行预测，最后取多数的预测结果为最终结果。随机森林中的决策树必须是不同的，为了实现这个不同决策树的生成，需要每棵决策树的训练样本是在训练集中有放回抽样产生的，一般训练集大小为 n ，即有放回抽样 n 次；假设训练集样本特征数为 M ，会从中选取 $m(m < M)$ 个作为单一决策树的特征集合，使不同的决策树根据不同的特征进行分类，且每棵决策树一般不进行剪枝，以保证不同决策树间的差异性。

由于随机森林每棵树的生成具有随机选择性，且各树的生成相互独立，具有高并行性，因而随机森林算法在处理高维度、大数据量的数据时具有优势。

• 支持向量机

把各个样本当作是点画在空间内，以二维空间为例，SVM 其实就是画一条线将两类点分隔开。这条线就叫决策面，SVM 其实就是选定和优化选定决策面的算法。考虑到要在两类数据间划分出足够的空隙、容量，具有“最大间隔”的决策面就是 SVM 要寻找的最优解。多用于二分类问题且 SVM 算法做文本分类较为耗时。

• 逻辑回归

常规的回归算法是对于文本的特征值 $\omega_1, \omega_2, \dots, \omega_n$ ，拟合出一个多项式：

$$f(\omega) = c_0 + c_1\omega_1 + c_2\omega_2 + \dots + c_n\omega_n$$

使得可以根据 $f(\omega)$ 的值来确定样本分类。

而逻辑回归是对其的一种优化，逻辑回归利用 sigmoid 函数的特殊性质，使最终拟合出的函数值可以直接表示某样本从属于某一类的概率，且保证函数值域为 $[0, 1]$ 。sigmoid 函数为：

$$\text{sig}(\omega) = \frac{1}{1 + e^{-\omega}}$$

假定样本为分类 c_1 的概率为 $p(\omega)$ ，根据 $\text{sig}(\omega) = \ln(p(\omega)/(1 - p(\omega)))$ ，两式联立可以解得：

$$p(\omega) = \frac{1}{1 + e^{-(c_0 + c_1\omega_1 + c_2\omega_2 + \dots + c_n\omega_n)}}$$

所以逻辑回归算法的训练就是拟合函数的过程。值得注意的是，由于 $p(\omega)$ 表示的是文本属于某一类的概率，所以对于样本分类数量为 n 的数据集，需要 $n - 1$ 个 $p(\omega)$ ，取 $p(\omega)$ 最大的分类作为结果。在二分类问题中，也可以表示为若 $p(\omega = 1) > 0.5$ ，则分类为 1，否则为测试文本分类为 0。

• 朴素贝叶斯分类

朴素贝叶斯是假设特征词间是相互独立的，并对每个样本计算出出现在各个类别中的概率，概率最大的就判别成该文的类别。设定类别集合为 $C = \{c_1, c_2, c_3, \dots, c_m\}$ ，样本 $d = \{\omega_1, \omega_2, \dots, \omega_n\}$ ， n 和 m 分别表示样本的特征词数和类别个数，则对于样本 d_i ，属于类别 c_j 的条件概率如下：

$$p(c_j|d_i) = \frac{p(d_i|c_j)p(c_j)}{p(d_i)}$$

根据假设，各特征词项相互独立， $p(d_i)$ 可以表示如下：

$$p(d_i|c_j) = p(\omega_{i1}|c_j) \times \dots \times p(\omega_{in}|c_j)$$

其中 ω_{ik} 表示文档 d_i 中第 k 个特征项，所以 $p(c_j|d_i)$ 可以表示如下：

$$p(c_j|d_i) = \frac{p(c_j) \sum_{k=1}^n p(\omega_{ik}|c_j)}{p(d_i)}$$

贝叶斯分类的优点是简单且性能较好，分类的过程效率高，通过朴素贝叶斯算法训练的分类器还可以直接用于新样本分类，在实时分类中效率较高。

4.3.2 模型建立

(1) 数据集划分

对数据集进行划分，分为训练集和测试集，调用 sklearn 库中的 `train_test_split` 函数。定义 `random_state` 为 1，以验证模型的精度。

但是经预处理后的数据中各类别标签样本数分布如下：

城乡建设	1986
劳动和社会保障	1957
教育文体	1566
商贸旅游	1158
环境保护	920
卫生计生	872
交通运输	593
Name: 一级标签, dtype: int64	

图 4-8 预处理后各类别标签样本数

常规的机器学习算法能处理类别均匀分布的数据，但是当我们将其应用到不均衡分布的数据，常规机器学习的结果会产生偏差。这种不均衡数据，会出现的问题叫做欠抽样和过抽样，在我们的优化中，为了对少数标签类别也能预测准确，我们需要对样本较多的标签类别信息量瘦身，这时可以进行数据抽样以保证各类标签样本量相对均衡。交通运输只有 593 条，我们以此为标准，对训练数据进行数据平衡化处理，从每类标签中抽取 500 个样本进行训练。

```
n=500          #100 抽样个数做对比
a=df_train[df_train['label']==1].sample(n)
b=df_train[df_train['label']==2].sample(n)
c=df_train[df_train['label']==3].sample(n)
d=df_train[df_train['label']==4].sample(n)
e=df_train[df_train['label']==5].sample(n)
f=df_train[df_train['label']==6].sample(n)
g=df_train[df_train['label']==7].sample(n)
df_train=pd.concat([a,b,c,d,e,f,g],axis=0)#拼接，列拼接
```

或者，直接定义

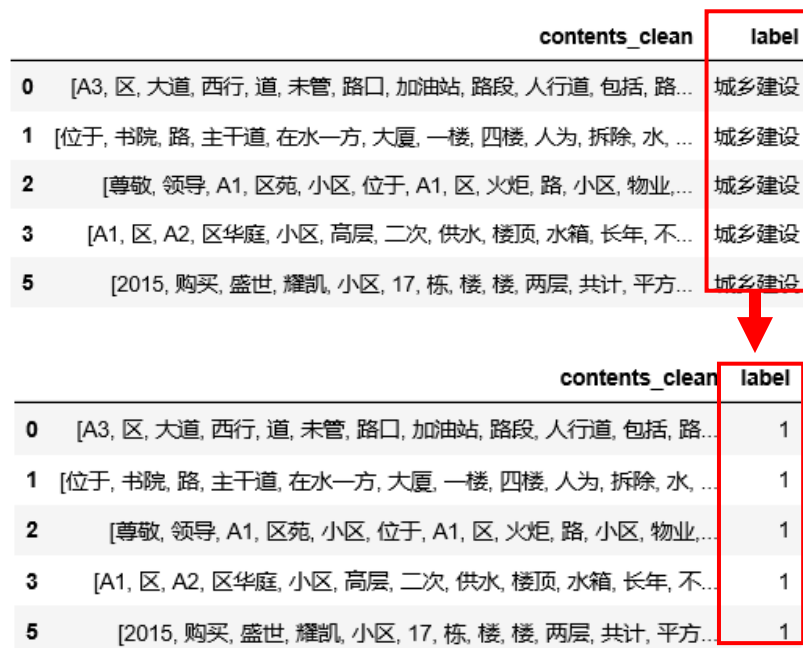
```
train_test_split(x_train, y_train, random_state = 1, stratify = y_train)
```

这里的 `stratify = y_train` 则是保证了训练集和测试集中各分类的比例一致，从而达到数据平衡的效果。

而本节通过模型评价后，选择采用第一种抽样方法，其模型效果更佳。

(2) 文本向量化

重新导入经预处理的数据，发现一共有 7 个分类，即要建立一个七分类的分类器。而机器学习算法中需要对本文做向量化，把类别标签映射成数字。



	contents_clean	label
0	[A3, 区, 大道, 西行, 道, 未管, 路口, 加油站, 路段, 人行道, 包括, 路...	城乡建设
1	[位于, 书院, 路, 主干道, 在水一方, 大厦, 一楼, 四楼, 人为, 拆除, 水, ...	城乡建设
2	[尊敬, 领导, A1, 区苑, 小区, 位于, A1, 区, 火炬, 路, 小区, 物业, ...	城乡建设
3	[A1, 区, A2, 区华庭, 小区, 高层, 二次, 供水, 楼顶, 水箱, 长年, 不...	城乡建设
5	[2015, 购买, 盛世, 耀凯, 小区, 17, 栋, 楼, 楼, 两层, 共计, 平方...	城乡建设

	contents_clean	label
0	[A3, 区, 大道, 西行, 道, 未管, 路口, 加油站, 路段, 人行道, 包括, 路...	1
1	[位于, 书院, 路, 主干道, 在水一方, 大厦, 一楼, 四楼, 人为, 拆除, 水, ...	1
2	[尊敬, 领导, A1, 区苑, 小区, 位于, A1, 区, 火炬, 路, 小区, 物业, ...	1
3	[A1, 区, A2, 区华庭, 小区, 高层, 二次, 供水, 楼顶, 水箱, 长年, 不...	1
5	[2015, 购买, 盛世, 耀凯, 小区, 17, 栋, 楼, 楼, 两层, 共计, 平方...	1

图 4-9 文本向量化的效果

而对于留言文本的向量化，则是通过将前面数据集的 list of list 格式转换为符合 sklearn 向量化的格式要求，再进行文本向量化

```
from sklearn.feature_extraction.text import CountVectorizer
words = []
for line_index in range(len(x_train)):
    try:
        words.append(' '.join(x_train[line_index]))
    except:
        print (line_index, word_index)
vec = CountVectorizer(analyzer='word', max_features=4000, lowercase=False)
vec.fit(words)
```

sklearn 中的 CountVectorizer 是一个通过词汇计数来将一个文档转换为向量方法。其原理为把所有词库中的词当成列，并初始化向量为 $[0, 0, \dots, 0, 0]$ ，长度由词库的词个数决定，如果文本中出现某一词，那么某一词的数就+1，最后得到词频的向量。Countvectorizer 中可根据语料库中的词频排序选出前 max_feature 的词，只得到 max_feature 维的特征。

(3) 模型训练并测试

从 sklearn 中导入多项式朴素贝叶斯算法包 (MultinomialNB)、SVM 算法、随机森林、逻辑回归算法包分别构造分类器，对训练数据进行训练。然后对测试数据做同样的向量化处理，进行预测结果，查看精度，对模型进行评价。

```
models = [  
    RandomForestClassifier(),  
    LinearSVC(),  
    MultinomialNB(),  
    LogisticRegression(random_state=0)]  
CV = 5  
entries = []  
for model in models:  
    #模型的名字  
    model_name = model.__class__.__name__  
    accuracies = cross_val_score(model, X,Y, scoring='f1_macro', cv=CV)  
    for fold_idx, accuracy in enumerate(accuracies):  
        entries.append((model_name, fold_idx, accuracy))
```

4.4 模型评价与优化

4.4.1 模型评价

对于二分类问题，常用评价指标为查准率及查全率，可将样例根据其真实类别和分类器预测类别划分为：

真正例 (TP)：真实类别为正例，预测类别为正例；

假正例 (FP)：真实类别为负例，预测类别为正例；

假负例 (FN)：真实类别为正例，预测类别为负例；

真负例 (TN)：真实类别为负例，预测类别为负例；

构建的混淆矩阵如下所示

表 4-1 二分类的混淆矩阵

真实\预测	0	1
0	TN	FP
1	FN	TP

- 准确率，又称查准率

$$P = \frac{TP}{TP + FP}$$

- 召回率，又称查全率

$$R = \frac{TP}{TP + FN}$$

- $F - Score$

$$F_1 = \frac{2 * P * R}{P + R}$$

查准率表示的是预测类别为负例的样本中真实类别为负例的样本所占的比例，查准率越高，模型负例的分类效果越好；查全率表示被正确分类的负例的比例，查全率越高，表示模型将负例误分为正例的模型概率越低，模型效果越好。而 $F - Score$ 则综合考虑查准率和查全率，是它们的加权调和平均。

而当我们在评价多分类问题的性能时，可以转换为 n 个二分类混淆矩阵，这时候的综合考察评价指标就会用到宏平均和微平均。宏平均(macro-average)和微平均(micro-average)是衡量文本分类器的指标。题目要求的即为宏平均指标，这里只介绍宏平均：

宏平均是先对每一个类统计指标值，然后再对所有类求算术平均值。在本节中以宏平均中的 $F - Score$ 指标来评价留言文本分类模型的性能：

- 查准率

$$P_i = \frac{TP_i}{TP_i + FP_i} = \frac{\text{类别 } i \text{ 识别正确的样本数}}{\text{被识别成类别 } i \text{ 的总样本数}}$$

$$Macro_P = \frac{1}{n} \sum_{i=1}^n P_i$$

- 查全率

$$R_i = \frac{\text{类别 } i \text{ 识别正确的样本数}}{\text{实际为类别 } i \text{ 的总样本数}}$$

$$Macro_R = \frac{1}{n} \sum_{i=1}^n R_i$$

- $F - Score$

$$Macro_F = \frac{2 * Macro_P * Macro_R}{Macro_P + Macro_R}$$

$$Macro_F = \frac{1}{n} \sum_{i=1}^n \frac{2 * P_i * R_i}{P_i + R_i}$$

当模型能够综合地考虑分类结果的精确性和全面性时，即能识别并准确分类各个标签的留言样本，模型的 $F - Score$ 才会越高。

在实现层面上，可以直接调用 sklearn 库中的 metrics，选择 f1_score 函数可以直接计算所需的 F_1 值，对于多分类的宏平均指标计算而言，需要设置参数 average="macro"，才能准确的计算多分类模型的 F_1 值。

经过上面将数据变为均衡数据、提取文本特征，我们接下来使用随机森林、支持向量机、逻辑回归、朴素贝叶斯这四种模型分别训练，并打印四种模型的表现：

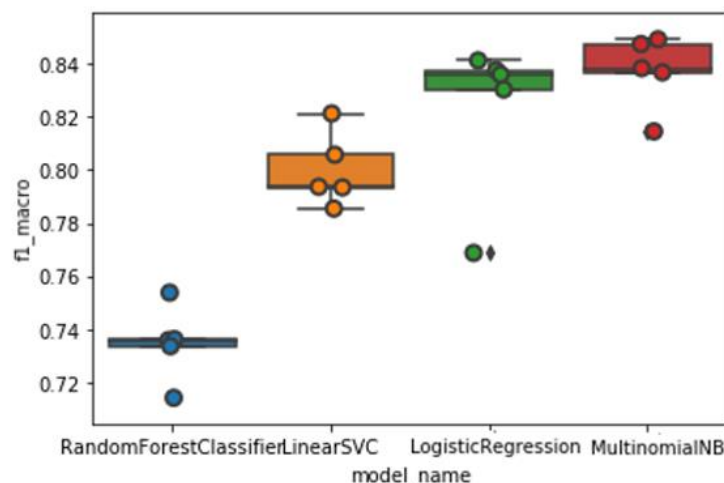


图 4-10 四种模型的 f1_score 值对比图

	model_name	fold_idx	f1_macro		model_name	fold_idx	f1_macro
0	RandomForestClassifier	0	0.735774	10	MultinomialNB	0	0.841070
1	RandomForestClassifier	1	0.735156	11	MultinomialNB	1	0.768727
2	RandomForestClassifier	2	0.739823	12	MultinomialNB	2	0.837372
3	RandomForestClassifier	3	0.742155	13	MultinomialNB	3	0.835796
4	RandomForestClassifier	4	0.742470	14	MultinomialNB	4	0.830160
5	LinearSVC	0	0.793333	15	LogisticRegression	0	0.836600
6	LinearSVC	1	0.785393	16	LogisticRegression	1	0.814396
7	LinearSVC	2	0.821134	17	LogisticRegression	2	0.847240
8	LinearSVC	3	0.793566	18	LogisticRegression	3	0.838154
9	LinearSVC	4	0.805647	19	LogisticRegression	4	0.848999

图 4-11 四种模型的 f1_score 值

model_name	
LinearSVC	0.799814
MultinomialNB	0.837078
LogisticRegression	0.822625
RandomForestClassifier	0.739076
Name: fl_macro, dtype: float64	

图 4-12 四种模型的 f1_score 平均值

经过比对，朴素贝叶斯算法是非常适合应用到多分类文本分析之中的。进一步对朴素贝叶斯模型的分类结果进行可视化：

模型得分:83.43				
	precision	recall	f1-score	support
城乡建设	0.69	0.69	0.69	127
环境保护	0.86	0.94	0.90	127
交通运输	0.82	0.82	0.82	137
教育文体	0.88	0.87	0.88	127
劳动和社会保障	0.85	0.82	0.84	124
商贸旅游	0.85	0.77	0.81	113
卫生计生	0.89	0.92	0.91	120
accuracy			0.83	875
macro avg	0.84	0.83	0.83	875
weighted avg	0.83	0.83	0.83	875

图 4-13 朴素贝叶斯模型的得分报告

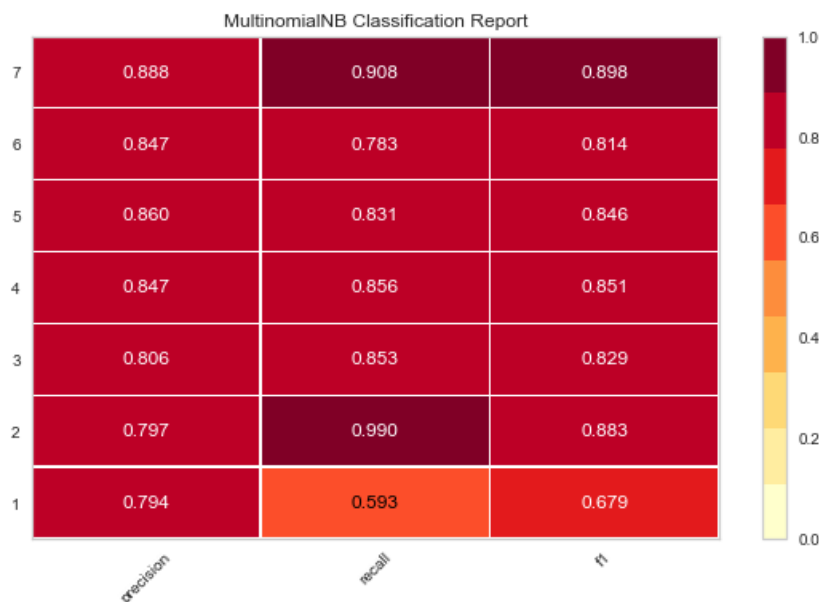


图 4-14 各分类中的模型得分

经过比对，基于多项式贝叶斯算法的文本分类模型其模型效果是四种模型中较好的，

它建立在查全率以及查准率两个指标的基础之上，综合地考虑了分类结果的精确性和全面性，可以选做留言文本的一级标签分类模型。

4.4.2 模型优化

上述模型的 F_1 值达到了 0.83,得到了较好的模型效果,但为了进一步提高模型精度,达到更好的查全率和查准率,除了更换算法之外,我们进行了如下优化:

(1) 词典更新

从附件 1 中抽取出附件 2 中涉及的 7 个一级标签类所含的二三级标签导入到用户词典中,附件 1 中的二三级标签大多数为专有名词,以此作为划分,减少出现错误分词的现象,增加分词的准确率。

userdict	
248	劳动环境
249	安全防护
250	劳动合同纠纷
251	非法用工
252	农民工权益
253	劳务派遣纠纷

图 4-15 自定义词典

(2) 建立标签类目树

本节中建立的模型只是基于附件 2 中出现的七个标签类别。但如果是对于附件 1 中的划分体系,其类目标签数目较多的话,一般会将类目标签按照一定的层次关系,建立类目树。那么接下来就可以利用层次分类器来做分类,先对第一层节点训练一个分类器,再对第二层训练 n 个分类器(n 为第一层的节点个数),依次类推。利用层次类目树,一方面单个模型更简单也更准确,另一方面可以避免类目标签之间的交叉影响,但如果上层分类有误差,误差将会向下传导。

可以先用某种无监督的聚类方法,将训练文本划分到某些 clusters,建立这些 clusters 与附件 1 类目体系的对应关系,然后人工处理这些 clusters,切分或者合并 cluster,提炼 name,再然后根据知识体系,建立层级的分类系统。

(3) 引入 TF-IDF 向量

在文本特征提取方面,上述模型建立过程中只使用了 CountVectorizer 方法,对于每一个训练文本,它只考虑每种词汇在该训练文本中出现的频率,将文本中的词语转换为词频矩阵,直接通过 fit_transform 函数计算各个词语出现的次数。但实际上,词频较小的词其重要程度很多时候往往高于词频较大的词,所以仅仅统计词频的方法在实际中有一定的缺陷。

为了提升模型的效果，可以更换文本向量的表示方式，选用 TF-IDF 向量。TF-IDF 的主要思想为：如果某个词比较少见，但是它在这篇文章中多次出现，那么它很可能就反映了这篇文章的特性，可能是我们所需要的关键词；当两个词在该篇文章中出现的次数相同，则常见词在该文章中的重要性要低于另外一个词，即常见词在该文章的特征词提取中的排序要低于另外一个词。

在前面留言内容文本向量化的步骤中，使用 `TfidfVectorizer` 方法代替 `CountVectorizer` 方法。即文本向量化的步骤改为：

```
from sklearn.feature_extraction.text import TfidfVectorizer
words = []
for line_index in range(len(x_train)):
    try:
        words.append(' '.join(x_train[line_index]))
    except:
        print (line_index,word_index)
vectorizer=TfidfVectorizer(analyzer='word',max_features=4000, lowercase = False)
vectorizer.fit(words)
```

这是 sklearn 的 `TfidfVectorizer`，可以将原始文档的集合转换为 TF-IDF 特性的矩阵，该类将 `CountVectorizer` 和 `TfidfTransformer` 类封装在一起，相当于 `CountVectorizer` 配合 `TfidfTransformer` 使用的效果。

经词典更新以及 TF-IDF 向量化后，再对测试数据集进行预测，评估模型效果，最终模型得分 F_1 值为 0.8549，提升了两个百分点。

模型得分:85.49				
	precision	recall	f1-score	support
城乡建设	0.82	0.69	0.75	127
环境保护	0.85	0.98	0.91	127
交通运输	0.83	0.89	0.86	137
教育文体	0.91	0.91	0.91	127
劳动和社会保障	0.84	0.83	0.83	124
商贸旅游	0.86	0.79	0.82	113
卫生计生	0.88	0.89	0.88	120
accuracy			0.85	875
macro avg	0.86	0.85	0.85	875
weighted avg	0.85	0.85	0.85	875

图 4-16 TF-IDF+MultinomialNB 模型得分

经上述比较，并通过调整参数训练模型，基于分类结果的精确性和全面性的综合评估，根据模型在验证集性能效果表现，我们选择效果最优的 $TF-IDF + MultinomialNB$ 模型做留言文本分类模型，将模型保存，便于后续的预测。

五、根据文本相似度将留言进行归类

留言的热度分析方法，包括以下步骤：

- (1) 对所有的留言进行聚类分析，分类成不同的事件；
- (2) 根据事件热点指数计算模型计算每个话题事件的热点指数。

5.1 数据预处理流程：

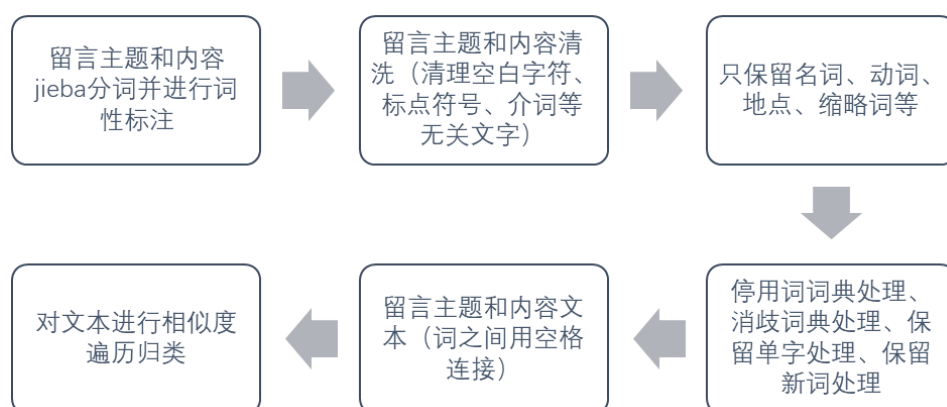


图 5-1 文本聚类预处理过程

上一节说明了对文本的分词以及清理的原理，但是要想在没有测试数据以及分类体系的情况下对文本进行聚类，则需要对文本的相似度进行遍历计算。

5.2 文本相似度的计算

基于 TextRank 的自动文摘算法，主要步骤如下：

- (1) 预处理：对留言进行分词以及去停用词处理，筛选出候选关键词集。
- (2) 计算句子间的相似度：采用如下公式进行计算句子 1 和句子 2 的相似度：

$$\text{句子的相似度} = \frac{\text{两个句子都出现的词的数目}}{\log(\text{句子 1 中的词数目}) + \log(\text{句子 2 中的词数目})}$$

若两个句子之间的相似度大于设定的阈值，则认为这两个句子之间有相似联系，，则可看成相连，在经过对所有的句子进行遍历计算后，可以将留言进行分类。

- (3) 计算句子的权重：

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

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。权重高者可作为该分类的摘要。

5.3 热度评价指标的建立

令类的集合 $Z = \{X_1, \dots, X_N\}$ ，第 i 类的数据集为 $X_i = \{x_{i1}, \dots, x_{in}\}$ ， x_{ip} 为 X_i 已知道的热度值最高的一条留言，

留言 x_{ik} 的热度值为

$$y_{ik} = 10 + 0.5 * n_{ik} + m_{ik}$$

该类热点事件的热度值为

$$Y_i = \sum_{k=n}^{k=1} y_{ik} \times \log_{1/e} |t_{ik} - t_{ip}|$$

t_{ik} ：距离 x_{i1} 的时间长度

n_{ik} ：反对数

得到结果如下

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	1776.88	2019/1/8至2019/10/25	西地省A市	58车贷非法经营诈骗
2	2	1126.24	2019/5/5至2019/9/19	A5区五矿万境K9县	房屋质量普遍出现问题
3	3	687.57	2019/8/23至2019/9/6	绿地海外滩小区	附近的长赣高铁噪音大
4	4	352.33	2019/6/25至2020/1/26	A2区丽发新城的居民	附近修建搅拌站严重扰民
5	5	243.35	2019/3/26至2019/4/15	A6区月亮岛片区	月亮岛路沿线架设110kv高压电线杆

图 5-2 热度问题表

六、答复意见质量的评价方案

6.1 问题解读

为了检验职能部门对群众留言的处理质量，需要针对相关部门对留言的答复意见质量给出一套评价方案，这还有利于建立高质量的智能政务问答对知识库。为解决上述问题，本文将答复意见的质量与留言内容进行比对，从答复的相关性、完整性、可解释性、时效性等角度建立答复意见质量评价指标，通过提取这些信息作为特征集合，训练分类器，用于判断答复文本的质量。

6.2 答复意见质量评价指标体系构建

- 相关性：答复意见的内容是否与问题相关，可以通过语义相似度的计算进行量化；
- 完整性：是否满足某种规范；
- 可解释性：答复意见中是否给出内容的相关解释；
- 时效性：职能部门的回复效率也影响着答复的质量

其中，相关性、完整性和可解释性属于文本特征维度。

表 6-1 评价特征选择表

维度	指标	解释及说明
相关性	词重叠	两个文本词汇重合比例
	词向量相似度	两个文本中词向量的余弦相似度在各区间的词向量组合数量在总组合数量中占的比例
	文本语义相似度	将文本表示成可计算的语义向量，从而计算两个向量之间的相似度（LDA 和 Doc2Vec）
	文本熵	包括基于词和字符的熵值
	类别距离	高质量的答复应该和问题从属于同一类别
	留言与答复耦合度	留言文本与答复之间的重叠部分，文本长度之比
完整性	文本长度	答复文本包含的字符数。答复文本的长度越长，答复越丰富和完整
	关键词数量	答复文本中包含的关键词数量
	停用词	答复文本中包含的停用词数量，停用词数量越少，质量越高

可解释性	外部引用数量	答复文本中包含的引用文本的数量，如法律文件、政府工作报告等具有
时效性	答复与留言生成间隔时间	答复时间与留言时间的间距

(1) 文本熵:

包括基于词和字符的熵值。文本熵值往往用于检验文本的多样性。对于一段 λ 长度的文本，包含 n 个不同单词的文本熵定义如下

$$E_r(p_1, \dots, p_n) = \frac{1}{\lambda} \sum_{i=1}^n p_i [\log_{10}(\lambda) - \log_{10}(p_i)]$$

其中， $p_i, i = 1, 2, \dots, n$ 是单词 i 在文本中的频率。

(2) 类别距离:

高质量的答复应该和问题从属于同一类别，例如交通运输类留言中如果出险城乡建设类别的答复，则不可能是较高的质量。类别距离的计算可以基于朴素贝叶斯算法，以一级标签分类下的留言文本作为训练语料，得到答复文本向量和留言所属的一级标签向量的距离得分。文档 d 和类别 C_ω 的距离定义如下：

$$Dist(d, C_\omega) = \frac{1}{P(d, |C_\omega)}$$

$$P(d, C_\omega) = P(d)P(C_\omega|d)$$

(3) 词向量相似度

对附件 4 的语料分析后发现，存在答复文本内容篇幅较小，且留言文本中的词汇在答复文本中出现频率很高，简单的使用词重叠特征很难区分这类答复文本。基于这种情况，我们基于 Word2Vec 的方法挖掘词汇的语义相似性，而不是简单的根据词汇的重叠来判断答复与留言的相关性。将留言和答复文本中的词汇表示成词向量，分别得到两个词向量的集合 A、B。然后在这两个词向量集合中分别取词向量得到 a_i 、 b_i ，并计算两个词向量的余弦相似度，计算公式如下

$$\cos = \frac{a_i \cdot b_i}{||a_i|| \times ||b_i||}$$

计算得到 A 和 B 的所有词向量组合的相似度后，设置多个相似度阈值进行统计，抽取的特征为，相似度在区间 0-0.1, 0.1-0.2, ..., 0.9-1 的词向量组合数量在总组合数量中占的比例。

参考文献

- [1]薛彬,陶海军,王加强.针对民生热线文本的热点挖掘系统设计[J].中国计量大学学报,2017,28(03):371-379.
- [2]冯振华.基于DBSCAN聚类算法的研究与应用[D].江南大学,2016.
- [3]龙银杏.基于GIS的移动通信网络投诉热点核查系统的分析与设计[D].武汉邮电科学研究院,2016.
- [4]朱瑞峰.基于Hadoop和R语言的网络自媒体热点挖掘系统的设计与实现[D].电子科技大学,2015.
- [5]曹彬,顾怡立,谢珍真,陈震.一种基于大数据技术的舆情监控系统[J].信息安全,2014(12):32-36.
- [6]黄禹忠.基于文本分类技术的客户投诉智能分析系统的设计与实现[D].湖南大学,2013.
- [7]夏海峰,陈军华.基于文本挖掘的投诉热点智能分类[J].上海师范大学学报(自然科学版),2013,42(05):470-475.
- [8]陈阳,凌俊民,蒙圣光.投诉数据智能挖掘分类管理系统[J].数字技术与应用,2011(06):146-149.
- [9]周启海,黄涛,张元新,吴红玉.同构化信息温度与热点发现应用初探[J].计算机科学,2007(11):113-117.