“智慧政务”中的文本挖掘应用

摘要

随着网络问政平台的多样化，各类社情民意相关的文本数据量不断攀升，依靠人工来进行留言划分和热点整理难以满足需求，为此我们利用自然语言处理和文本挖掘的方法对该问题进行解决。

首先我们对留言文本进行预处理，为建模做准备：首先对非文本字符进行剔除和替换处理，其次使用pycorrector工具纠正错别字，消除错别字影响；然后使用jieba分词、自定义停用词、去停用词，消除虚词等干扰；最后基于TF-IDE构建文本特征向量。

针对问题一，我们通过构建基于高斯朴素贝叶斯分类器模型对留言数据中的主题和详情文本特征进行分类。首先选取预处理后留言数据集的80%进行模型训练，20%数据进行模型验证。其次采取“留言主题：留言详情：留言主题+留言详情=2：3：5”的方式计算分类结果。最后通过测试全部数据对模型进行评价，得到该模型的评价得分为0.88。

针对问题二，我们首先通过留言信息构建的特征向量，构建基于Cosine计算相似度的归类模型，根据归类后留言的点赞数、反对数、频率定义热度评价指标。其次根据归类结果、热点评价指标大小，创建热点问题留言明细表，同时提取前5个热点问题(表6-2)的时间范围、描述等创建热点问题表。详细内容见附件“热点问题表.csv”、“热点问题留言明.csv”。

针对问题三，我们创建基于得分和评语集的答复评价体系对留言的答复进行评价。首先量化答复相关性、完整性、可解释性三个指标并对其进行评审。接着，根据评审结果输出此条答复评分、评语，例如，“80分 内容全面 答复迅速”，详细内容见附件“答复评价表.csv”。

最后，本文对实验流程进行分析和总结，对模型的优点和不足进行了深刻的探讨。

关键词：TF-IDE；高斯朴素贝叶斯；Cosine相似度，F1-Score，自然语言处理

目录

1 挖掘目标

1.1 挖掘背景

近年来，我国“互联网政治”渐入佳境。领导人上网了解民情、政府上网公开政务、公民上网表达意见，互联网已经成为政府和人民互动的良好平台。“互联网政治”是政府面向社会的窗口，也是公众与政府互动的渠道，对于促进政务公开、推进依法行政、接受公众监督、改进行政管理、全面履行政府职能具有重要意义。

然而在微信、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、凝聚民气重要渠道的同时，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。

此题需要根据附件给出的收集自互联网公开来源的群众问政留言记录，及相关部门对部分群众留言的答复意见，分析留言记录和答复意见运用数据挖掘方法，建立有效的数学模型对留言进行分类和热点排序，对回复进行综合性评价。

随着“互联网政治”与人们生活的不断融合，如何通过自然语言处理和文本挖掘的方法对海量的留言和回复进行分类显得至关重要，更加提高留言划分和热点整理的相关部门的工作效率，从而提高国人的幸福感。建立利用自然语言处理和文本挖掘的方法对留言进行分类和评价综合评价模型对留言划分和热点整理的相关部门有着重要的意义。

1.2 挖掘目标

附件给出了收集自互联网公开来源的群众问政留言记录，及相关部门对部分群众留言的答复意见，利用自然语言处理和文本挖掘的方法解决下面的问题。

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。并且使用F-Score对分类方法进行评价。

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”。

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

2.1 问题一的分析

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，对群众的留言进行分类。分析附件二中给出的数据，着眼于留言主题和留言详情这两个数据维度。首先，对数据进行清洗，包括清除空白符等无意义符合、文本纠错；接着，对清洗好的数据进行分词并删除停用词，完成数据预处理；然后，利用TF-IDF选取文本特征，构建特征向量；在算法选择上，使用高斯朴素贝叶斯分类算法，通过对分类模型训练，构建关于留言内容的一级标签分类模型。最后使用F-Score对分类模型进行评价。

2.2 问题二的分析

热点问题挖掘，对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，并将排名前5的热点问题保存为文件“热点问题表.xls”，详细信息按相应格式保存为“热点问题留言明细表.xls”。采用聚类分析的方法根据留言标题将留言进行聚类，结合每条留言的点赞数和反对数得到热点问题的排名情况。

针对问题二，通过问题聚类合并相似问题，继而通过定义热度指数对热点问题进行排名，整理数据得到“热点问题表.xls”和“热点问题留言明细表.xls”。本文采用TF-IDF构建特征向量，采用贝叶斯理论计算留言问题的相似度，进而对相似问题归类。热点指数考虑两方面因素，问题的频数与持续时间的比值和该问题的赞成数与反对数之和，通过对两因素结果加权求和量化热点指数。

2.3 问题三的分析

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。

针对问题三，本文从答复的相关性、完整性、可解释性三个角度出发建立答复意见的评价方案。其中，答复的相关性由问题与答复内容的切合度决定；完整性由答复的格式、内容决定；可解释性由答复的时间延期决定。通过量化指标，并使用熵权法对各个指标加权得到最终评分，并输出评语如“答复及时”，“内容全面”，“格式残缺”。

3 模型假设与符号说明

3.1 模型假设

- 假设所给的数据真实，可靠。
- 假设问题某段时间内的热度指数跟它在这段时间内的点赞数和反对数和频数有关。
- 假设答复意见的质量评价由相关性、完整性、可解释性来体现。

3.2 符号说明

表 1 符号说明

符号	意义
t_i	留言i的特征项
w_{ij}	t_i在文本j中的权重
N_i	属于类别的文本数目
c_i	训练集的文本总数。
t_j	文本特征集合中的第j个特征词
I_i	第i个问题留言的热点指数
F_i	归类后第i个问题的留言频率（个/天）
K_i	第i个问题的赞成数与反对数
P_i	第i条答复的评价得分
R_i	第i条答复的相关性量化
C_i	第i条答复的完整性量化
E_i	第i条答复的可解释性量化

本表未涉及的符号在文中有具体介绍。

4 数据预处理

文本预处理是指将非结构化文本信息转换为计算机可以处理的形式。过滤噪声数据和提高文本数据的质量是文本预处理技术的主要功能。在文本分类过程中，它是一系列操作的通用术语，例如文本标记，分词，词干提取和停用词删除。经过预处理后，可以提高文本表示的质量，并减少了不利于分类的噪音，本文进行数据预处理的流程图如下所示：

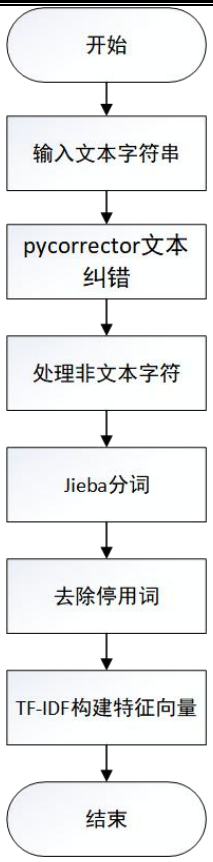


图 4-1 数据预处理流程图

4.1 输入文本字符串进行清洗

文本中有一些标记，例如数字，标点符号等。这种类型的标记通常不切实际，也不能用作区分文本类别的基础，在某些文本分类问题中，它们的存在会代替对文本进行分类它带来了干扰，例如，在垃圾邮件分类中，经常在垃圾邮件中混入特殊符号，从而逃避了分类器的识别。因此，需要删除它们以避免干扰分类器的性能并减轻分类负担。

4.2 pycorrector文本纠错

中文文本的自动校对分为两个方面：错误检测和纠正。其中，纠错是为检测到的错误提供纠正建议。在纠错方面，一般来说，根据其采用的策略，可以将其大致分为以下三类。

（1）采用模式匹配方法

对长词进行纠错处理，但该方法没有充分利用出错字串的特征，算法计算量大。

（2）采用替换字表结合主词典

一种纠错算法，通过添加单词和替换单词来为检测到的错误字符串提供错误纠正建

议。然而，该算法的局限性在于，纠错建议仅限于纠错字表而没有考虑上下文启发式信息，主要只考虑了纠错的纠错类型。单个单词错误仅由替换单词表中的候选单词替换。诸如遗漏单词，多个单词，易位和多个单词替换之类的错误类型的错误纠正能力很弱。

（3）采用混淆集

检查错误时，根据算法，针对出现在文本中的单词字符串y，从y的混淆集中选择最适合上下文的单词字符串w。如果w不等于y，则将其视为错误，并使用w作为错误纠正建议。这样，该算法将校对问题转换为消歧问题。但是，该算法复杂且计算量大。

4.3 文本分词

分词是指按照相关的分词算法将连续的文本数据划分成词序列的处理过程。对汉语来说，它的词与词之间通常没有明确的分隔符，因此这类语言的词的获取就必须要通过某种算法来实现，这是一个非常复杂的处理过程。在汉语自然语言的处理过程中中文分词技术是第一步，更是中文文本信息处理的关键与基础。下面主要介绍基于词库匹配、基于知识理解和基于词频度统计的三类主流中文分词算法。

（1）基于词库匹配的分词算法

通常，基于词库匹配的算法的基本思想是根据规定的扫描顺序和匹配原理，将待分割的中文文本数据与现有大型词库中的单词进行比较。在同义词库中找到相应的同义词库后，立即将字符串切出。尽管这种方法在理论上具有简单的概念和高效率的分词，但是在复杂的中文系统中，统一规则的制定和词典的完整性使它难以在实践中广泛使用。

（2）基于知识理解的分词算法

基于知识理解的分词是让计算机智能模拟人类对句子的理解过程，以达到文本分词的目的。基本思想是提高分词的准确性，并在分词过程中综合考虑语义和语法信息，以适当地解决歧义问题。分词系统通常可以分为三个子系统：通用控制系统，分词子系统和句法语义子系统。同时，由于整个中文系统非常复杂，如何将中文信息完整，准确地翻译成机器可以直接读取，处理的形式仍然是一个难以突破的问题。因此，当前基于中文知识理解的文本分割系统没有实际的应用意义。

（3）基于词频度统计的分词算法

基于词频统计的词分割是指使用概率统计算法提取词的相关特征。基于词频统计的词分割方法的优点在于，它基本上不需要先验的文本数据知识，也不需要构建任何词典。

但是，在使用这种方法提取信息之后，特征向量空间的维数过高，导致分类系统的效率低下。

论文中我们采用基于词频度统计的分词算法进行文本分词产生的云词图如下所示：

4.4 删除停用词

停用词是文本集合中的无用词，对确定文本类别没有积极影响，但出现频率很高，从而增加了分类系统的负担。因此，在文本分类之前，需要引入停用词列表以过滤掉停用词。

4-2 环境保护分词词云图

4-3 商贸旅行分词词云图

4.5 TF-IDF构建特征向量

在众多文本相似度量的方法中，TF-IDF（词频-逆文档频率，Term Frequency–Inverse Document Frequency）无疑是最为典型的一种，该方法最初应用于信息检索领域，现在也常见于文本数据挖掘中，由于其计算方法简单易用，在业界得到了普遍认可。对于一个文件集或某个语料库的某个文件来说，TF-IDF方法是一种用于评价某一字词对于它们的重要程度的有效方法。TF-IDF的方法假设：如果一个词在当前文档中出现的频率高，

而在其他文档中很少出现，则它更能代表该文档的主题。

Step 1 **选取文本特征**

本文采用基于TF-IDF +特征融合的方法来抽取文档关键词，尽可能少的依赖文档外部信息——所引用的外部信息越少，则提取关键词越不受领域约束。为了保证方法的通用性，本文只考虑文档自身能提供的基本特征：词性特征，词长度和词位置。

根据文献［6］所示，标题词和首尾句子更能代表文档的主题思想，然而，对于留言本来而言，文档本身没有首尾段落，本身首尾句的重要程度与其他句子没有明显的区别，故为了关键词抽取的通用性考虑，仅考虑标题位置信息。对于标题词，设定权重为α，采用与文献［7］相同的标题词权值计算公式，公式如下：

$$title(word)=\begin{cases}\partial & word \in Title\\ 1 & word \notin Title\end{cases} \tag{4-1}$$

其中，Title为标题词集合，文献［6］已经对α的取值进行充分的实验分析，本文采用与其相同的参数值1.3作为实验数据。通常，长词信息量要比短词信息量大，本文添加了词长信息。为了降低长词的影响，利用文档最长词的长度除以候选词词长作为词长得分，计算公式采用文献［6］的公式：

$$weight(len)=word_{len}/\max_{len} \tag{4-2}$$

其中 $word_{len}$ 表示当前词长， $\max_{len}$ 表示文档中最长词长，词性不同，作为关键词的重要度也有所不同，故本文根据不同的词性，对候选词进行不同的词性打分。经过实验，设定动词词性得分为0.5，字符串0.7，名词0.8，SW和MWE都视为未登录词，设定得分为1.0， 专有名词设为1.2。由于中文词性标注中，有一定的兼类词，在同一文本集合中，既作为动词出现又作为名词出现。例如“建议”，在“建议改善生活”中标记为动词，在“提出建议”中则标记为名词，一个文档中， 若“建议”作为不同词性出现N次，它的重要程度应该比同频率的名词得分低，动词得分高。本文针对该现象，考虑词性信息，对于不同词性标注的相同词分别计算其频率，改进基本TF-IDF公式如下：

$$Score(word)=(\sum_{i=0}^{n-1}weight(POS_i)\times fre(POS_i))\times IDE\times(title(word)+weight(len)) \tag{4-3}$$

其中， $Score(word)$ 为当前候选词 $word$ 的最终得分， POS_i 为候选词 $word$ 的词性标注集合中的第i个词性， $weight(POS_i)$ 表示词的第i个词性的得分， $fre(POS_i)$ 表示word词性标注为 POS_i 时的出现频率.

Step 2 文本表示

其实文本表示的本质就是将文本数据使用某种方法转换成一个数值或者向量，这样计算机才能对其进行处理。对文本进行数值化或者向量化是因为计算机无法直接处理文本中的字符串数据，因此需要对文本中的字符串数据进行数值化或者向量化。

向量空间模型（Vector Space Model，简称VSM）是由Salton和McGill[18]于1969年提出的，因其概念简单，就是用向量空间中的运算来简化代替文本的内容，这时语义的相似度是由空间上的相似度表现的。因此广泛应用于信息检索(International Retrieval，简称IR)。从信息检索中引进的向量空间模型这种方法主要应用于文本的自动分类当中。向量空间模型的基本思想是：将特征空间坐标系内的每个维度都表示成每个词项，每一个向量都代表一篇文档，文本间相似度的大小就用向量间的夹角余弦值衡量，在向量空间模型中，每一组规范化的正交词条向量张成的向量空间的一个点都用来表示一个文本，即为n维空间中的向量，所以我们可以将文本抽象为：

$$V(d_j)=((t_1,w_{1j}),\cdots,(t_i,w_{ij}),\cdots,(t_n,w_{nj})),i=1,\cdots,n \qquad (4\text{-}4)$$

其中，t_i是特征项，w_{ij}是t_i在文本d_j中的权重。向量空间模型的构造过程及文本表示如图所示：

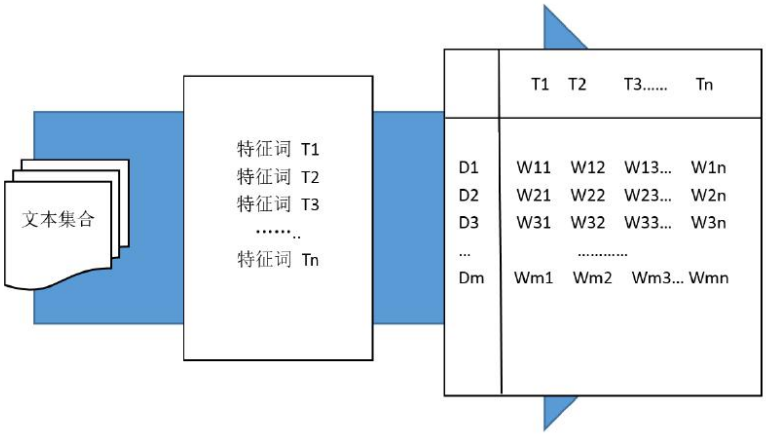


图 4-4 向量空间模型的构造过程及文本表示

在向量空间模型中，我们可以发现，一篇文本的类别标签与这些词在文本中出现的次序是没有关系的，而是与某些特征词条在该文本中出现的频率有关。事实上，向量空间模型就是一种词袋(Bag of Words)文本表示模型，它不需要考虑词与词之间的上下位关系，出现的位置顺序和文本长度。文本在这个模型中被视为一组词条所构成的集合，这些词条之间彼此相互独立且没有关系。将文本内的每个词项的词频信息都充分利用起来的过程就是文本分类的主要过程。这也就很好的说明了为什么可计算性和可操作性能够被向量空间模型所具备。然而对于向量空间模型的特征项的权重以及文本相似度计算而

言，模型只提供了一个理论框架基础，并未做明确统一的规定。可根据具体的实际情况对不同的分类系统的权重和相似度的计算采用不同的方法，基于这一特性，使得向量空间模型具有广泛的适应性，可以更好的应用于各种分类系统。同时，由于机器学习方法和向量空间模型的结合越来越紧密，而前者又处于不断壮大发展规模的阶段，这使得主流文本表示模型的地位一直被向量空间模型所占据。随着文本分类技术逐渐进步，有许多新的文本表示方法被提出，但这些新的方法大都基于向量空间模型，在其基础上进行的改进和扩展。本文的文本表示方法也将采用向量空间模型。

4.6 预处理结果示例

以编号为165845的留言为例：

“ 张厅长： 您好！我是一名医务工作者。就目前的医患关系，尤以市，县级医院突出。每当有患者在治疗效果欠佳，或医治无效时。其家属基本上采取”闹“，“打”，“砸”甚至侵害医护人员安全。根本听不进任何解释。可我们也是上有老，下有妻儿的百姓。谁来保障我们医务人员的人身安全呢。我爱人也是同行。每天上班心中未塌实过。见了穿工作服的就打让人心寒。在我们行使“救死扶伤”的职责时,谁给予了医务者的安全。是您吗？

输入：“每天上班心中未塌实过。”

表 4-1 文本分类步骤示例

文本纠错	使用pycorrector文本纠错工具进行纠错 输出：‘每天上班心中未踏实过’，[‘塌实’，‘踏实’，7，8]
非文本处理	替换非文本字符如:’ ’,’:’,’ 。’等 输出：“每天上班心中未踏实过”
分词	使用jieba进行分词 输出： [‘每天’，‘上班’，‘心中’，‘未’，‘踏实’，‘过’]
删除停用词	自定义停用词表：[‘与’，‘且’，‘之’...] 消除中文常见词和没有预测能力的虚词影响 输出：[‘每天’，‘上班’，‘心中’，‘未’，‘踏实’]

5 问题一群众留言分类

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，对群众的留言进行分类。为此我们定义了文本分类系统，其主要功能模块简述如下：

预处理：为了提高文本表示的质量，方便后续的处理，对于原始的文本语料，需要进行格式化等预处理操作。 文本表示：文本表示需要解决的问题包括：一是选择什么样的语言元素作为文本特征，大多选择词或者词组；二是选择什么样的模型量化文本对象。

特征降维：为了实现文本分类，需要从文本中选择出最能反映文档主题的特征，要求特征数量尽可能少且准确，并给出文本特征的权重度量方法。

分类模型：如何设计文本分类器是文本分类方法的主要研究内容，首先选择能够代表分类系统中每个类别的文本作为训练集，从训练集中学习得到分类器，并实现新对象的分类。

性能评价：此步骤目的在于评价分类方法和系统性能的优劣。对于不同的分类问题可以采用不同的评价参数，如单标签分类和多标签分类问题就会使用不同的参数。文本分类器性能评价方法包括召回率、准确率、F值、微平均和宏平均等，从而改进分类系统性能。

该分类系统的流程图如图5-1所示：

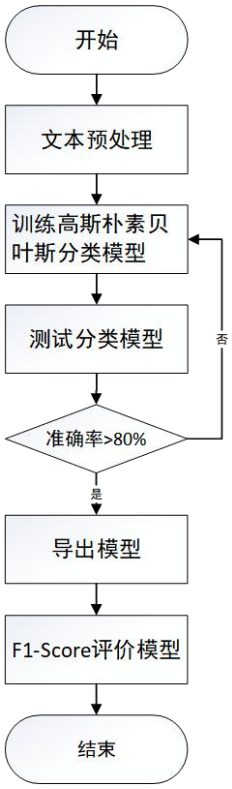


图 5-1　问题一流程图

5.1　高斯朴素贝叶斯分类模型

文本分类一般分有两种方法：基于训练集的文本分类方法；基于分类的词汇的文本分类方法。两种分类方法的研究角度不同，训练集的方法更多的是应用于计算机及人工

智能领域；词表分类法则常见情报学领域。本文将采用基于分类的词汇的文本分类方法对已经处理好的文本进行分类。

贝叶斯分类方法以贝叶斯定理为理论基础，采用了概率推理方法。贝叶斯分类的原理就是通过计算给定样本在各个类别上的后验概率，然后把该样本判定为最大后验概率所对应的类别。而在计算后验概率的过程中，需要知道数据集中每个类别的先验概率，以及属性的条件概率。类别的先验概率可以通过统计的手段预先知道，而属性的条件概率也可以通过统计的方法或者假定的分布模型来估计。

朴素贝叶斯模型的数学基础坚实，分类效率稳定。该算法可应用到大规模文本集合中，具有方法简单、速度快、分类准确率高等优点。理论上，由于朴素贝叶斯分类算法所基于的假设太过于严格，因此其分类效果要普遍由于其他分类算法。但是在实际应用中并不能完全符合理论中的假设条件，则该分类算法的准确率就会有一定程度的下降。在类别数目比较多或者是类别之间相关性较小的情况下，该模型的分类性能才能达到最佳状态。

5.1.1 训练模型

假设训练集中存在的j个类别，类别集合表示为$C=(c_1,c_2,\cdots,c_j)$，文本特征词集合表示为力$T=(t_1,t_2,\cdots,t_j)$，各个文本特征对给定文本类别的影响是相互独立的。那么，类别的先验概率为：

$$P(c_i)=\frac{N_i}{N},i=1,2,3\cdots j \tag{4-1}$$

其中，N_i表示属于类别的文本数目，c_i表示训练集的文本总数。

设t_j为表示文本特征集合中的第j个特征词，表示特征词在所有属于类别的文档集中出现的概率。则未知类别文本d属于文本类别c_i的条件概率$P(d\mid c_i)$为：

$$P(d\mid c_i)=P((t_1,t_2,\cdots,t_j)\mid c_i)=\prod_{i=1}^{j}P(t_j\mid c_i) \tag{4-2}$$

根据贝叶斯定理，文本类别c_i的后验概率$P(c_i\mid d)$为：

$$P(c_i\mid d)=\frac{P(d\mid c_i)P(c_i)}{P(d)} \tag{4-3}$$

$$P(d)=\sum_{i=1}^{j}P(c_i)P(d\mid c_i) \tag{4-4}$$

其中，公式里的$P(d)$代表d文本中的所有特征词在整个文本集合中出现的概率，对于各个文本类别来说都是常数，因此可以舍去，简化公式后如下：

$$P(c_i \mid d) = P(d \mid c_i)P(c_i) \quad (4\text{-}5)$$

综上所述，结合公式（4-6）和（4-9）和可得：

$$P(c_i \mid d) = P(c_i)\prod_{i=1}^{j} P(t_j \mid c_i) \quad (4\text{-}6)$$

利用式（4-6）计算出每个类别对于文档d的后验概率值，然后将文档d判定到概率值最大的那个文本类别中去。

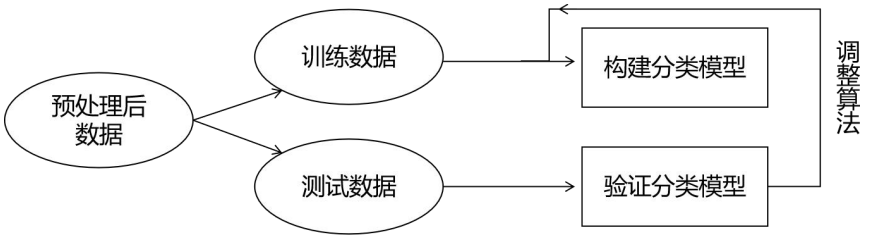
5.2 模型评价

5.2.1 评价方法

对于构建的分类器，为了验证该分类器性能的优劣，需要进行测试和评价。根据所使用的测试数据的不同，分为封闭性和开放性两种。其中，前者在训练过程中使用的数据，也应用于测试；而后者所采用的测试数据仅用于测试过程，在训练过程中并不会用到。由于封闭性测试中训练阶段和测试阶段中使用了相同的数据，所以很可能出现过拟合现象，不利于新文本的分类。所以，在测试分类器性能时，经常使用开放性测试。 分割数据集时，要确保各个类别的数据比例适当，以免影响各类间的先验概率。根据将数据集划分为训练集和测试集方法的不同，通常可以采用以下两种方法[107]：

（1）保持法

保持法的主要思路就是，把收集整理的文档样本数据随机的划分成训练集和测试集，并且这两个子集要保持相对的独立。在训练过程中，按照特定的规则公式利用训练集数据进行计算，并归纳建立分类模型；在测试过程中，利用测试集数据根据分类模型进行文本类别判定，并且对分类模型的准确率进行评估。保持法的缺点在于，想要保证分类算法的准确度通常需要较大规模的文本训练集合。因此，在文本自动分类过程中，利用保持法的训练集至少要包含三分之二的文本数据，而其余的数据就作为测试集使用。由于该方法只使用部分的文本数据用于文本分类，因此保持法是一种保守的评估方法。保持法的原理如下图所示：



5-2 保持法原理

（2）K 折交叉验证

在 K 折交叉验证（K-fold cross-validation）中，数据集被划分为k个大小相等的子集Si，i=1,2,⋯,k，每个子集之间互不相交。选取其中的k-1个子集用于训练，另外一个子集用于测试，进行k次这样的训练和测试。分类器的准确率表示为k次测试中被正确分类的文本总数与数据集的总文本数之间的比值。在实际应用中，k的值通常取10。

5.2.2 评价模型

本模型采用的评价指标主要包括准确率以及F1-Score。

首先我们可以计算准确率(Accuracy)，其定义是: 对于给定的测试数据集，分类器正确分类的样本数与总样本数之比。也就是损失函数是0-1损失时测试数据集上的准确率 。

准确率（Accuracy）即对于给定的测试数据集，模型正确分类的样本数与总样本数之比。准确率在某些场合下确实可以有效地评价一个模型的好坏，然而在一些极端情况下，却显得不是那么重要了。例如：某个地区的总人数为100万人，而人群中患有某种病的人数只有100人。一个负责检测行人是否患有该病的模型只需要持续将行人归为“不患病”一类，即可获得超过99%的概率。

$$Accuracy = \frac{\text{“预测正确”的样本数}}{\text{总样本数}} \tag{4-7}$$

“预测正确”的样本数显然单纯使用准确率是不够的，我们引入F1-Score来解决上述现象，作为我们判断模型好坏的另一个指标。在进行评价之前，我们需要先定义TP，FN，FP，TN四种分类情况。

表 4-1 F1-Score分类情况

	相关（正类）	无关（负类）
被检索到	TP	FP
未被检索到	FN	TN

TP（true positives）：正类判定为正类

FN（false negatives）：正类判定为负类

FP（false positives）：负类判定为正类

TN（true negatives）：负类判定为负类

对于这道题我们需要所有分类完毕的文本中寻找分类正确的文本

查准率(precision)是所有被检索到的item（TP+FP）中，“应该被检索到的item（TP）”占的比例，其公式为：

$$P = \frac{TP}{TP + FP} \tag{4-8}$$

查全率(recall)是所有检索到的item（TP）占所有“应该被检索到的item（TP+FN）”的比例，其公式为：

$$R = \frac{TP}{TP + FN} \tag{4-9}$$

F1-Score是精确率（precision）和召回率（recall）的调和函数。为了符合赛题评分标准，作出如下修改：

$$F1 - score = \frac{\text{“预测标签为1且真是标签也为1的”样本数}}{\text{“标签为1”的样本数} + \text{“真实标签为1”的样本数}} \tag{4-10}$$

考虑到该问要分类的文本有很多种，整个模型最终的评价结果，为每一类文本分类的F1-Score模型的值的总和求平均值。所以对公式进行了如下修改：

$$\begin{aligned} F &= \frac{1}{n} \times (\frac{2P_1R_1}{P_1 + R_1} + \frac{2P_2R_2}{P_2 + R_2} + \cdots + \frac{2P_nR_n}{P_n + R_n}) \\ &= \frac{1}{n} \sum_{i=1}^{n} \frac{2P_iR_i}{P_i + R_i} \end{aligned} \tag{4-11}$$

其中 P_i 为第i类的查准率， R_i 为第i类的查全率。

5.2.3 评价结果

（1）实验环境。

在模型验证过程中，我们主要基 win10 的操作系统，实验环境为 8 G 的内存容量，128G 的固态硬盘容量，Inter i5 的 GPU，主要以 Python 为开发语言。

（2）模型采用F-score进行评价

$$F_1 = \frac{1}{n} \times (\frac{2P_1R_1}{P_1 + R_1} + \frac{2P_2R_2}{P_2 + R_2} + ... \frac{2P_nR_n}{P_n + R_n}) = \frac{1}{n} \sum_{i=1}^{n} \frac{2P_iR_i}{P_i + R_i} = \frac{1}{n} \sum_{i=1}^{n} F_{1i} \tag{5-12}$$

其中：

$$F_{1i} = \frac{2P_iR_i}{P_i + R_i}, P_i = \frac{TP_i}{TP_i + FP_i}, R_i = \frac{TP_i}{TP_i + TF_i} \tag{5-13}$$

TP_i 第i类分类正确的数量。

FP_i 其他类被误认为第i类的数量。

TF_i 第i类被误认为其它类的数量。

全部数据可分为城乡建设、环境保护、交通运输、教育文体、劳动和社会保障、商贸旅游、卫生计生共7个一级分类，每次通过80%数据的进行训练，20%的数据进行测试其结果如下。

表 5-2 城乡建设测试数据

城乡建设TP	FP	FN	P	R	F1-score
80	13	25	0.86	0.76	0.79
81	19	17	0.81	0.83	0.82
87	11	5	0.83	0.88	0.85
81	12	17	0.87	0.83	0.85
92	13	3	0.88	0.97	0.92
89	6	5	0.94	0.95	0.94
80	0	0	1	1	1

综上模型的F1-score的值为：

$$F1\text{-}score = \frac{0.76+0.83+0.85+0.83+0.92+1}{7} = 0.88 \quad (5\text{-}14)$$

此处可以有图，展示这7个类别的分类情况，即某一类正确分类的数量，被认为其它类的数量。

6 问题二热点问题挖掘

热点问题挖掘，对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，并将排名前5的热点问题保存为文件“热点问题表.xls”，详细信息按相应格式保存为“热点问题留言明细表.xls”。采用聚类分析的方法根据留言标题将留言进行聚类，结合每条留言的点赞数和反对数得到热点问题的排名情况。

针对问题二，通过问题聚类合并相似问题，继而通过定义热度指数对热点问题进行排名，整理数据得到”热点问题表.xls”和”热点问题留言明细表.xls”。本文采用TF-IDF构建特征向量，采用贝叶斯理论计算留言问题的相似度，进而对相似问题归类。热点指数考虑两方面因素，问题的频数与持续时间的比值和该问题的赞成数与反对数之和，通过对两因素结果加权求和量化热点指数，该问的求解流程图如图6-1所示。

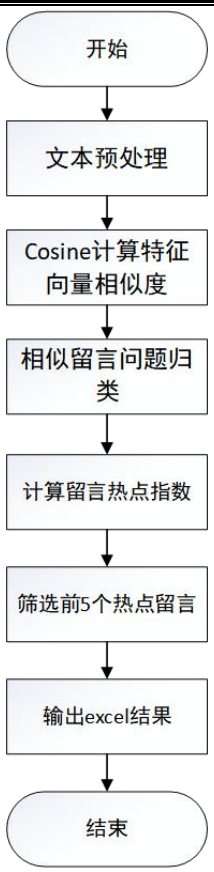


图 6-1 问题二流程图

6.1 理论基础

6.1.1 定义热点指数

热点（hot spot）指的是比较受广大群众关注，或者欢迎的新闻或者信息，或指某时期引人注目的地方或问题。如“社会热点”、“绵山成为旅游的热点”等。

第二问中所说的热点就是人们在某一时间段内最关注的话题，这里的关注不能只从留言的数量来看，还要考虑该条留言的赞成数和反对数来看。

人们对事物的理解是多面的，每一个话题不同人有不同的看法，所以无论是赞成和反对，都代表了人们对这个话题的关注情况，所以我们在第二问中热度指数定义为留言主题、赞成数和反对数的总和。

6.1.2 Cosine相似度

这里我们用相似文本聚类来体现该话题的热度，接下来对本文采用的余弦相似度进行介绍。

余弦距离，也称为余弦相似度，是用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小的度量。余弦值越接近1，就表明夹角越接近0度，也就是两个向量

越相似，这就叫"余弦相似性"。

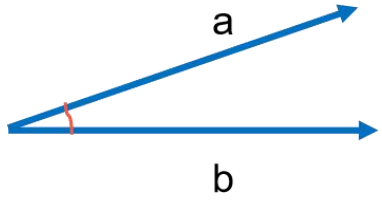


图 6-2 ab相似度较高

图6-2两个向量a,b的夹角很小可以说a向量和b向量有很高的的相似性，极端情况下，a和b向量完全重合。如下图：

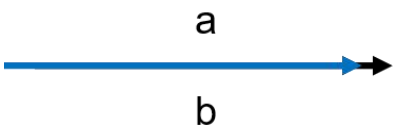


图 6-3 ab完全相似

如上图6-3：可以认为a和b向量是相等的，也即a，b向量代表的文本是完全相似的，或者说是相等的。如果a和b向量夹角较大，或者反方向。如下图

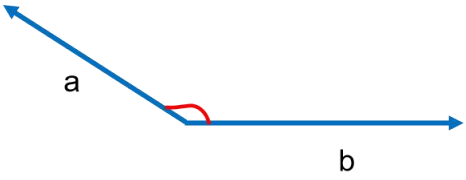


图 6-4 ab相似度低

如上图6-4: 两个向量a,b的夹角很大可以说a向量和b向量有很低的的相似性，或者说a和b向量代表的文本基本不相似。

在向量表示的三角形中，假设a向量是（x1, y1），b向量是(x2, y2)，那么可以将余弦定理改写成下面的形式：

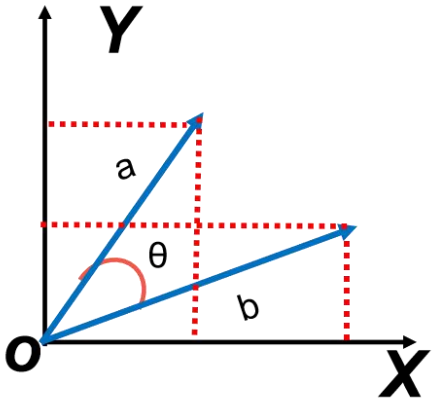


图 6-5 改进余弦定理

向量a和向量b的夹角的余弦计算如下：

$$\cos(\theta) = \frac{a \cdot b}{\|a\| \times \|b\|} = \frac{(x_1, y_2) \cdot (x_2, y_2)}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}} \tag{6-1}$$

扩展，如果向量a和b不是二维而是n维，上述余弦的计算法仍然正确。假定a和b是两个n维向量，a是，b是，则a与b的夹角的余弦等于：

$$\cos(\theta) = \frac{\sum_{i=1}^{n}(X_i \times Y_i)}{\sqrt{\sum_{i=1}^{n}(X_i)^2} \times \sqrt{\sum_{i=1}^{n}(Y_i)^2}} = \frac{a \cdot b}{\|a\| \times \|b\|} \tag{6-2}$$

余弦值越接近1，就表明夹角越接近0度，也就是两个向量越相似，夹角等于0，即两个向量相等，这就叫"余弦相似性"。

另外：余弦距离使用两个向量夹角的余弦值作为衡量两个个体间差异的大小。相比欧氏距离，余弦距离更加注重两个向量在方向上的差异。

借助三维坐标系来看下欧氏距离和余弦距离的区别：

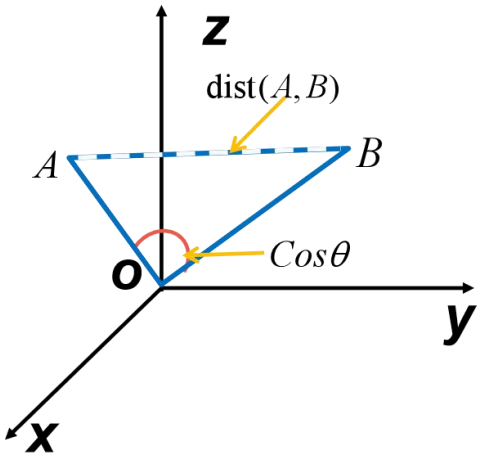


图 6-6 欧式距离和余弦距离比较

6.1.3 计算文本相似度举例

举一个例子来说明，用上述理论计算文本的相似性。为了简单起见，先从句子着手。

句子A：A5 区魅力之城小区一楼被搞成商业门面，噪音扰 民严重。

句子B：A 市魅力之城小区底层商铺营业到凌晨，各种噪音好痛苦，

怎样计算上面两句话的相似程度？

基本思路是：如果这两句话的用词越相似，它们的内容就应该越相似。因此，可以从词频入手，计算它们的相似程度。

表 6-1 文本相似度举例

1、分词、去停用词	句子A：A 5区 魅力之城 小区 一楼 商业 噪音 扰民 句子B：A 市 A市 魅力之城 小区 底层 商铺 营业 凌晨 噪音
2、列出所有的词	A 5区 魅力之城 小区 一楼 商业 噪音 扰民 底层 商铺 营业 凌晨 噪音
3、计算TF，IDF构建文本特征向量	句子A：(1/8*log(2/3),0,1/8*log(2/3),1/8*log(2/3)...) 句子B：(1/10*log(2/3),0,1/10*log(2/3),1/10*log(2/3)...)

表 6-2TF-IDF计算过程表

	A	5区	魅力之城	小区	噪音	…
A TF	1/8	1/8	1/8	1/8	1/8	…
B TF	1/10	0	1/10	1/10	1/10	…
DF	2	1	2	2	2	…
IDF	Log(2/3)	Log(2/2)	Log(2/3)	Log(2/3)	Log(2/3)	…
A TF-IDF	1/8* log(2/3)	0	1/8* log(2/3)	1/8* log(2/3)	1/8* log(2/3)	…
B TF-IDF	1/10* log(2/3)	0	1/10* log(2/3)	1/10* log(2/3)	1/10* log(2/3)	…

每篇文章各取出若干个关键词，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。

到这里，问题就变成了如何计算这两个向量的相似程度。我们可以把它们想象成空间中的两条线段，都是从原点（[0, 0, ...]）出发，指向不同的方向。两条线段之间形成一个夹角，如果夹角为0度，意味着方向相同、线段重合,这是表示两个向量代表的文本完全相等；如果夹角为90度，意味着形成直角，方向完全不相似；如果夹角为180度，意味着方向正好相反。因此，我们可以通过夹角的大小，来判断向量的相似程度。夹角越小，就代表越相似。

$$\cos(\theta)=\frac{\sum_{i=1}^{n}(X_i\times Y_i)}{\sqrt{\sum_{i=1}^{n}(X_i)^2}\times\sqrt{\sum_{i=1}^{n}(Y_i)^2}} \tag{6-3}$$

计算两个句子向量：

句子A：(1/8*log(2/3),0,1/8*log(2/3),1/8*log(2/3)...)和句子B：(1/10*log(2/3),0,1/10*log(2/3),1/10*log(2/3)...)的向量余弦值来确定两个句子的相似度，带入公式求得夹角的余弦值为0.81。

6.2 模型建立

6.2.1 留言相似度计算

留言相似度的计算不不是单纯的文本相似度，为了避免极端现象的出现，这里通过三个指标来判断留言的最终相似度，计算公式如下：

$$\text{Similarity} = \text{TT} \times 0.2 + \text{SS} \times 0.3 + \text{TS} \times 0.5 \quad (6\text{-}4)$$

其中TT代表标题和标题之间的Cosion相似度，标题的字词比较少，有可能出现全部相同，或全部不相同的情况，为了避免极端现象的发生影响模型整体的准确度，所以标题和标题相似度的占比为20%。

SS带表内容和内容Cosion相似度占最终相似度的30%。

TS代表每条留言标题内容并集的再求Cosion相关度占比最终结果的50%。

当最终相关度达到0.4则说明这两条留言是相关的。

6.2.2 留言聚类

聚类，也被称为无监督分类，在没有任何先验知识的前提下，把对象集划分为有意义的集群，称之为簇，下面介绍几种基本的聚类方法。

（1）基于划分的方法

基于划分的聚类方法是最为常用的聚类方法之一，其中最为知名的要数k-means算法了。该算法按照样本与簇中心的平均最近距离把样本集分为k个簇，对新得到的各个簇重新计算簇中心从而进行不断迭代逼近最优。其中，求簇中心点的算法可以很简单地使用各个向量维度的平均值，也可以使用Minkowski Distance公式、Euclidean Distance公式、CityBlock Distance公式。点与点之间的距离可以使用欧式聚类、曼哈顿聚类或向量夹角余弦等。k-means聚类算法如下：

输入：簇的数目k和包含n个对象的数据库。

输出：k个簇，使平方误差最小。

算法步骤：

a. 为每个聚类确定一个初始聚类中心，这样就有k个初始聚类中心。

b. 将样本集中的样本按照最小距离原则分配到最邻近聚类。

c. 使用每个聚类中的样本均值作为新的聚类中心。

d. 重复步骤2-3直到聚类中心不再变化。

e. 结束，得到k个聚类。

但是，k-means算法存在几方面的不足：

首先，算法通常只能得到局部最优的聚类，而不是全局最优的聚类。另外，初始聚类中心的选择会对算法的结果产生很大的影响，而初始中心的选择并没有好的方法。在使用k-means算法时，通常会多次进行运算，使用不同的随机生成的聚类中心点计算聚类结果，然后对各自结果进行评估，选择最优的一个。

k-means算法需要指定初始值k，但是对于没有先验知识的数据集来说，k值是很难进行测定的。算法对异常点敏感。离中心点较远的异常点往往会使计算得到的簇中心点偏了实际簇中心点。

（2）基于层次的方法

层次聚类算法[9]根据其聚类方向分为凝聚式层次聚类与分裂式层次聚类。凝聚式层次聚类起初把每一个数据对象作为一个簇，计算簇之间的两两相似度，选择相似度最高的两个簇进行合并，并重新计算新合成的簇与其他簇之间的相似度，如此迭代进行直到聚为一类或者最大的相似度小于设定的阈值。分裂式层次聚类算法正好是凝聚式算法的逆过程。层次聚类的结果形成一颗层次树。在层次聚类过程中，簇间的每次合并和分裂都需要计算簇之间的距离，目前主要有四种度量方法：平均距离、平均值距离、最大距离、最小距离。平均距离是簇A中所有点与簇B中所有点分别计算距离，求其平均值；平均值距离是计算簇A的中心点a和簇B的中心点b，然后求a和b间的距离；最大距离是取簇A中所有点与簇B中所有点间距离的最大值；最小距离是取簇A中所有点与簇B中所有点间距离的最小值。

BIRCH、CURE、ROCK和CHAMELEON算法在层次聚类中比较流行。BIRCH算法1的使用提出的聚类特征构建类似于B-树的聚类特征树；CURE算法[]是Guha等人提出的，传统方法采用一个簇中心点代替簇进行接下来的计算，该算法在簇集中选取多个点代表这些点所在的簇集，这使得该算法可以对非球形数据集进行聚类，从而对异常点可以进行过滤，较好的屏蔽噪声；ROCK算法[]是基于数据点链接进行相似度计算的聚类算法，适用于不同类别属的文本数据；CHAMELEON算法采用动态建模技术，首先建立稀疏图，然后对进行图分隔和图合并，最终形成簇集。

（3）基于密度的方法

基于密度的方法把数据看成是分布在空间中的点，点密度比较稠密的部分趋于形成

类簇，而点密度比较稀疏的部分则形成簇集之间的分界线。这种方法不仅能发现凸形的簇，也能应用于任意形状的聚类簇。

比较常见的方法有DBSCAN（Density-based Spatial Clustering of Applications with Noise）[10]，这种算法把高密度联通区域作为一个簇集，对异常点及噪声数据不敏感，并且能发现任意形状的类簇。DBSCAN对簇的定义基于“密度可达性”概念。其中，如果点q与点p的距离不超过给定的s，p点周围有足够多的点（也称p为核心点对象），则定义两点之间“直接密度可达”；如果有序列$<p_1,p_2,\cdots,p_n>$，其中有p_i与p_{i+1}直接密度可达，$p=p_1$、$p_0=q$，则定义p到q“密度可达”。

需要注意的是，密度可达关系并不是对称的。点q可能处于聚类簇的边缘位置，而不能成为核心点对象。因此，可以找到一条以非核心点对象为终点的路径。由于这种非对称性，导出“密度相连”概念：如果点o对点p和点q都是密度可达的，则说明q、p密度相连。密度相连的概念是对称的。一个簇需要满足两点：

簇中所有的点都是相互密度相连的。

如果一个点和簇中任何一个点事密度相连的，则它属于该簇。

DBSCAN通过循环迭代执行区域查询，获取密度可达对象，形成簇集。除此之外，比较常见的基于密度的聚类算法好包括DENCLUEU和采用顺序聚类的OPTICS算法。

6.2.3 计算留言热度指数

留言的热度指数具有相对性，某一个时间段内的热点是相对于其他留言而言的，这里我们通过对主题进行评分来代表该主题的热度指数。

将某一时间段内留言平率最高的话题题为基准，热度指数的计算公式如下

$$I_i=100\times(\lambda\times\frac{F_i}{F_{\max}}+(1-\lambda)\times\frac{K_i}{K_{\max}})\quad 0\le\lambda\le1 \tag{5-5}$$

其中$F_{\max}$为留言聚类结果最高的话题，λ用来确定第i条留言的聚类结果和$F_{\max}$的比重，这里我们取λ的值为0.9。

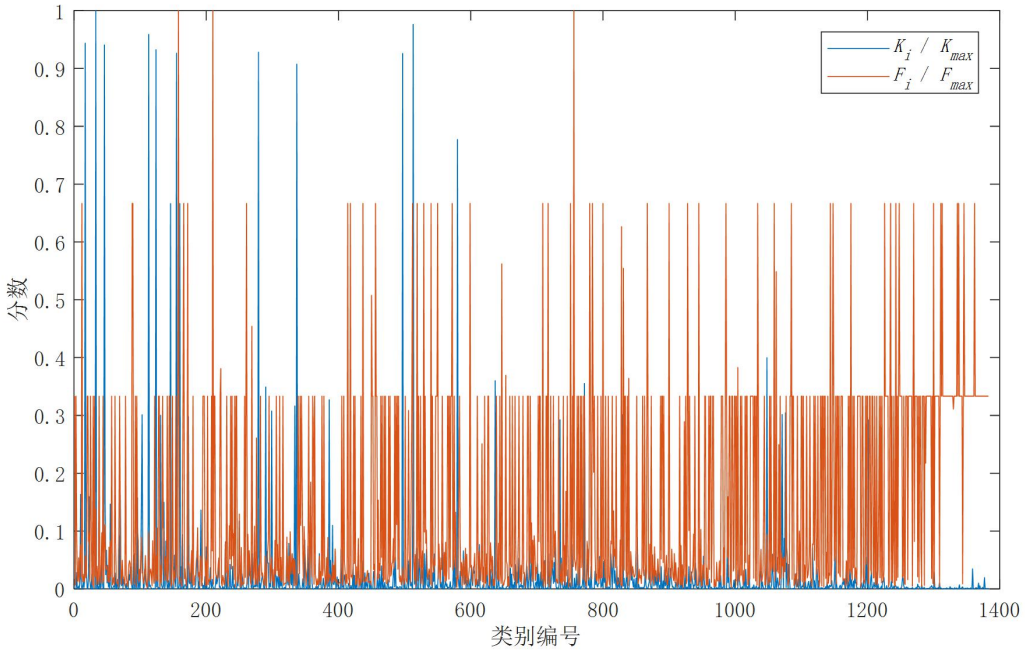


图 6-7 第i个问题的留言指数

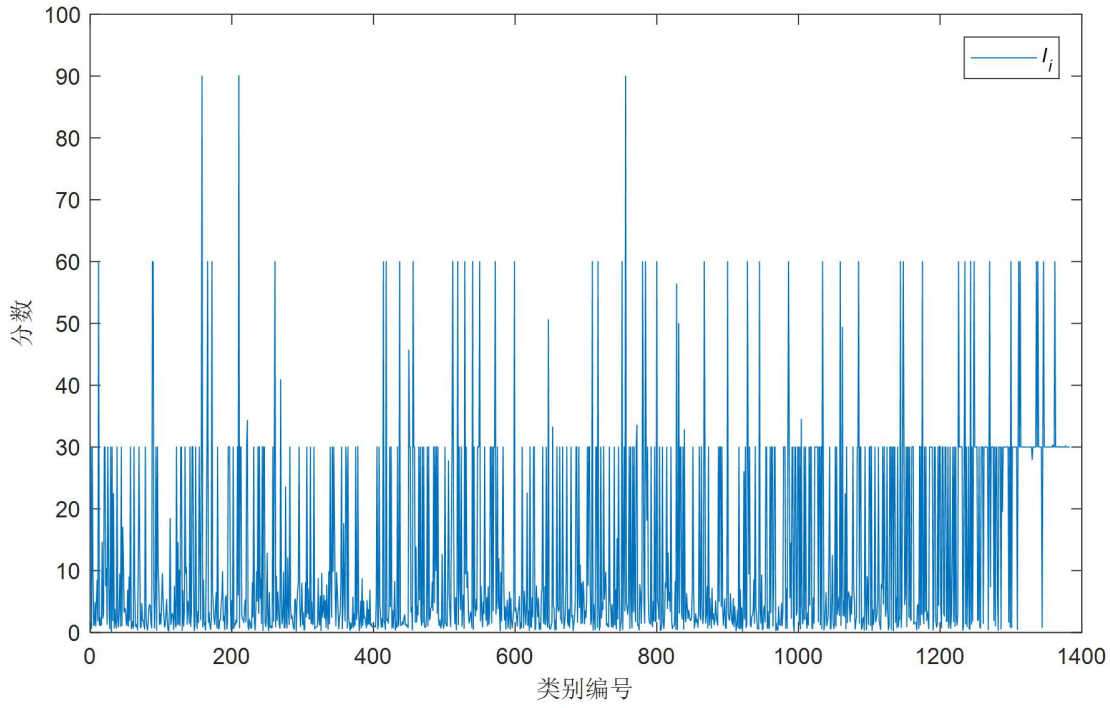


图 6-8 每一类问题的留言频率

6.3 模型结果

通过建模得到排名前5的热点话问题，如表6-2所示。

表 6-3排名前5的热点问题

热点排名	问题ID	热点指数	时间范围	地点/人群	问题描述
1	1	90.04759844	2019/02/14至2019/02/14	A4区悦湖山小区物业管理	A4区悦湖山小区多物业管理混乱，多次发生安全事故
2	2	90.00432713	2019/01/09至2019/11/10	A市服务中心	A市公立医院管理服务中心办事现场秩序混乱
3	3	90.00432713	2019/08/19至2019/08/19	A市	谁让A市民中心的违章建筑演变成非法殡仪馆?
4	4	60.05192557	2019/01/16至2020/01/06	A市	对A市禁摩限令处罚不解
5	5	60.03894418	2019/01/15至2019/12/02	A市高升路	咨询A市高升路往东向延伸等相关问题

通过热点问题挖掘模型得到的“热点问题表.xls”和“热点问题留言明细表.xls”件附件。

7　问题三答复意见的评价

7.1　相关定义

7.1.1　答复的相关性R

相关性代表问题与答复的语义相关度衡量（$0 \le R \le 1$）。

在进行语义相关度计算时，我们依旧采用文本相关度的计算方法，将主题和问题分词后的结果进行拼接然后进行关键字的提取，比较关键字和答复的相似度。

如果 $R < 0.1$，这说明答复与问题的语义相似度很差，此答复得0分，否则进行完整性和可解释性检查。

7.1.2　答复的完整性C

完整性表示回复的格式是否完整，在这里我们定义一个完整的答复应该具备：礼貌问候、内容完整、礼貌谢语、答复时间这四个模块，并且对这四个部分设置有不同的分值。

表 7-1 完整性各模块所占分值

模块名称	符号表示	所占分值（分）
内容完整	c_1	70* R
礼貌问候	c_2	10
礼貌谢语	c_3	10
答复时间	c_4	10

答复的完整性最终得分由四个模块的得分总和来判定，得分情况如公式7-1所示：

$$C = c_1 + c_2 + c_3 + c_4 \tag{7-1}$$

7.1.3　答复的可解释性E

这里我们定义答复可解释程度由留言提问到收到答复的时间长度来确定，答复时间越长，可解释性越差。

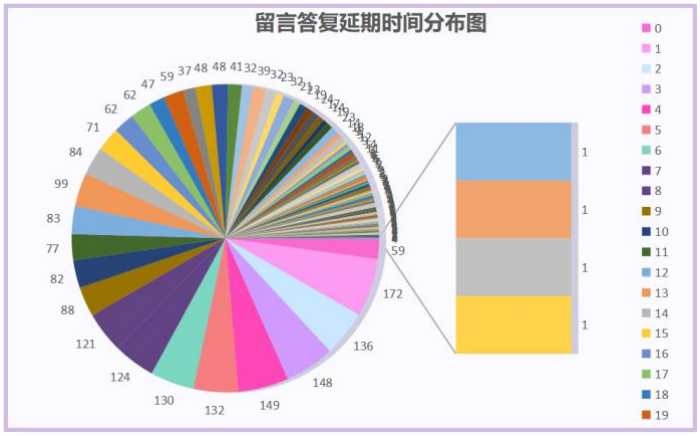


图 7-1 留言答复延长时间分布图

通过分析附件四的全部数据，我们得到留言答复延期时间分布图如图7-1所示，观察可知接近70%的留言在19天内得到答复，评价延期答复平均时间为20天，最短0天，最长1161天。依此得到答复的可解释性E的函数。

$$\Lambda d = T_d - T_l \tag{7-2}$$

其中 T_d 为答复时间，T_l 为留言时间。

$$E = \begin{cases} 100 & \Lambda d \in [0,8) \\ 90 & \Lambda d \in [8,15) \\ 75 & \Lambda d \in [15,22) \\ 55 & \Lambda d \in [22,29) \\ 30 & \Lambda d \in [29,36) \\ 0 & \Lambda d \in [36,+\infty) \end{cases} \tag{7-3}$$

一周内 100分，两周内 90分，三周内 75分，四周内 55分，五周内 30分，五周后0分。

7.2 定义评价函数、评语集

7.2.1 评价函数

对于第i个留言的答复，其得分 P_i 的公式为：

$$P_i = \begin{cases} 0 & R_i \le 0.1 \\ \alpha C_i + (1+\alpha) E_i & R_i > 0.1 \end{cases} \tag{7-4}$$

其中参数α通过熵权法确定。参数α的值是根据熵权法确定E、R权重来确定的，这里我们取60%的留言，对E、R进行计算。

计算E，R之前首先进行数据标准化，即去除数据量纲，我们通过公式7-5将每一个指标下的10所有10个数据转化到[0,1]之间。

$$y_{ij} = \frac{x_{ij} - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{7-5}$$

其次根据信息熵公式计算各指标的信息熵（如果p值为0，那么plnp=0）：

$$E_j = -\frac{\sum_{i=1}^{n} p_{ij} \ln p_{ij}}{\ln n} \tag{7-6}$$

$$p_{ij} = \frac{y_{ij}}{\sum_{i=1}^{n} y_{ij}} \tag{7-7}$$

最后根据指标权重计算公式各指标的权重：

$$W_j = \frac{1 - E_j}{m - \sum_{j=1}^{m} E_j} \tag{7-8}$$

由上述步骤即可得到信息熵和权重。

7.2.2 评语集

对每一条答复对应特定的评语集，公式如下：

Reply={答复及时、格式完整、内容全面、格式残缺、答复过时、答复无效、内容缺失}

评语集可以很直观的看出每条留言的回复情况。

7.3 输出评价

以某条答复为例

其初始化输出评语集reply=''，则评分P=0，然后对其回复评价相关指数进行评估。

1. 计算答复相关指数R

2. 如果答复相关指数R≤0.1，评分P=0，评语reply+=答复无效，输出结果；否则进行下一步

3. 计算答复完整性指数C

(1) 如果内容完整性c1>=50 ，reply评语集中加入‘内容全面’指标。

(2) 如果格式完整性 $c_2 + c_3 + c_4 = 30$，则reply评语集中加入‘格式完整’指标；如果 $c_2 = 0 \| c_3 = 0 \| c_4 = 0$，reply评语集中加入‘格式残缺’指标。

4. 计算答复可解释性指数E

如果E=100, 则reply评语集中加入‘答复及时’指标。

如果E=0, 则 reply评语集中加入‘答复过时’指标。

5. 计算评分

6. 输出评分和评语

整体的输出评价流程如图7-1所示：

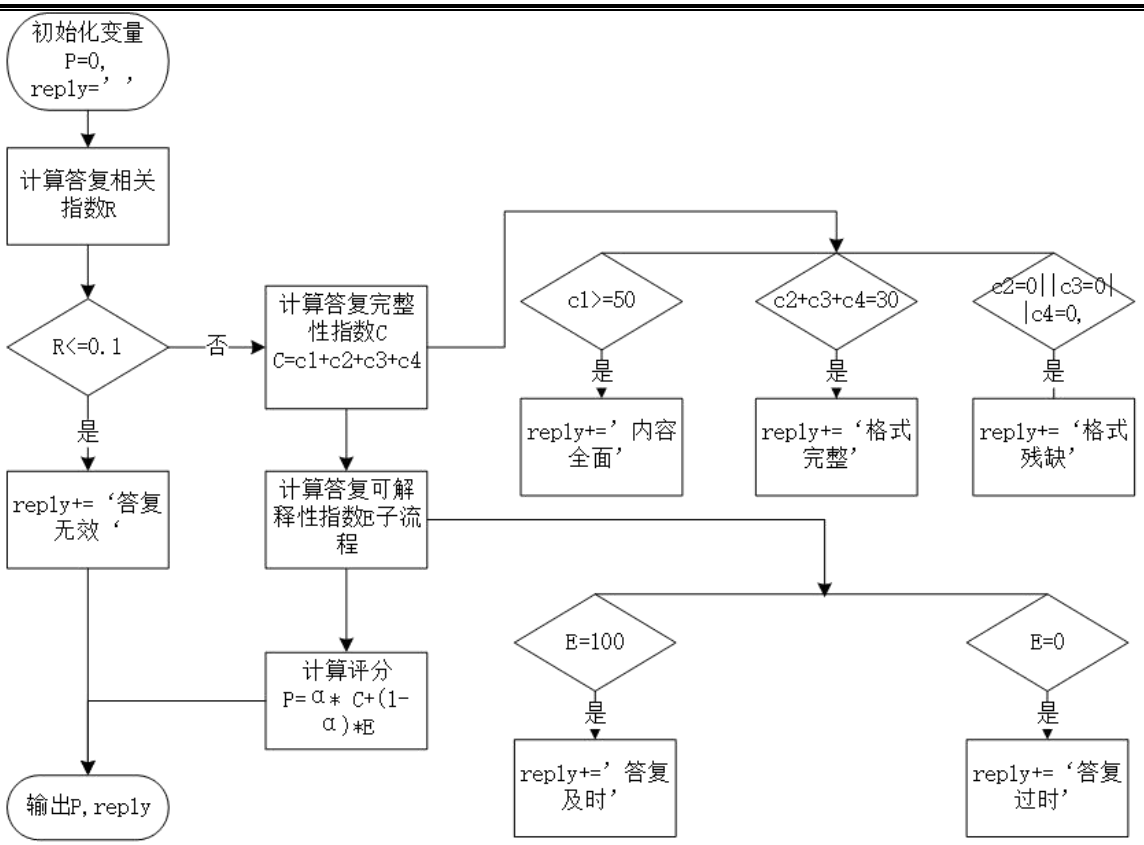


图 7-2 意见评价流程图

7.4 答复评价结果

通过建模我们得到的留言答复的评价结果见附件“答复评价表.csv”。

8 总结

随着网络问政平台的多样化，各类社情民意相关的文本数据量不断攀升，依靠人工来进行留言划分和热点整理难以满足需求，为此我们利用自然语言处理和文本挖掘的方法对该问题进行解决。

本文对文本分类的基础理论知识和相关技术进行了研究，分析了目前文本分类研究中存在的热点问题，从提高分类方法的快速性、准确性和个性化角度出发，首先提出一种特征选择方法，实现特征空间降维；然后利用深入挖掘得到的用户兴趣信息，针对文本分类提出了一种基于高斯朴素贝叶斯分类器模型，并且构建基于Cosine计算相似度的归类模型，根据归类后留言的点赞数、反对数、频率定义热度评价指标。并且创建基于得分和评语集的答复评价体系对留言的答复进行评价。首先量化答复相关性、完整性、可解释性三个指标并对其进行评审。

本文研究内容和创新工作主要包括以下四点： 本文首先对文本预处理的相关技术做了详细、系统的介绍，针对文本分类中的特征提取过程和算法进行了进一步的研究与探讨。然后搭建基于高斯朴素贝叶斯分类器模型，实现了对留言的分类，并且构建基于Cosine计算相似度的归类模型，根据归类后留言的点赞数、反对数、频率定义热度评价指标，得到留言的热度指数并且实现对留言热度的排序。最后创建基于得分和评语集的答复评价体系对留言的答复进行评价，通量化答复相关性、完整性、可解释性三个指标并对其进行评审。极大的提高了留言划分和热点整理的相关部门的工作效率。

9 参考文献

[1]谭静. 基于向量空间模型的文本相似度算法研究[D].西南石油大学,2015.

[2]王子慕. 一种利用TF-IDF方法结合词汇语义信息的文本相似度量方法研究[D].吉林大学,2015.

[3]孙润志. 基于语义理解的文本相似度计算研究与实现[D].中国科学院研究生院（沈阳计算技术研究所）,2015.

[4]黄承慧,印鉴,侯昉. 一种结合词项语义信息和TF-IDF方法的文本相似度量方法[J].计算机学报,2011,34(05):856-864.

[5]陈飞宏. 基于向量空间模型的中文文本相似度算法研究[D].电子科技大学,2011.

[6]孙润志. 基于语义理解的文本相似度计算研究与实现[D].中国科学院研究生院（沈阳计算技术研究所）,2015.

[7]黄承慧,印鉴,侯昉. 一种结合词项语义信息和TF-IDF方法的文本相似度量方法[J].计算机学报,2011,34(05):856-864.

[8]邬启为. 基于向量空间的文本聚类方法与实现[D].北京交通大学,2014.

[9]王纵虎. 聚类分析优化关键技术研究[D].西安电子科技大学,2012.

[10]程杨. 中文留言本聚类算法的研究[D].吉林大学,2016.

[11]陈飞宏. 基于向量空间模型的中文文本相似度算法研究[D].电子科技大学,2011.

[12]彭敏,黄佳佳,朱佳晖,黄济民,刘纪平. 基于频繁项集的海量留言本聚类与主题抽取[J]. 计算机研究与发展,2015,52(09):1941-1953.

附录

附录一：生成文本分类模型部分代码

```
message_category=message_data()
message_category['flag']=message_category['first'].apply(lambda x:0 if x=='城乡建设' else 1)
message_category_sample = pd.concat([message_category[message_category['flag'] == 0].sample(500),
                                     message_category[message_category['flag'] == 1].sample(500)],
                                    axis=0)
# 导入用户词典
#     jieba.load_userdict('./Data/Dict/messageDict.txt')
data_cut = message_category_sample['content'].apply(lambda x: jieba.lcut(x))
# 添加停用词
stop_dict = pd.read_csv('./Data/Dict/stopDict.txt', encoding='utf8', sep='TestTest', header=None, engine='python')
# 添加自定义停用词
custom_stop_dict = pd.read_csv('./Data/Dict/customStopDict.txt', encoding='utf8', sep='TestTest', header=None, engine='python
')
stop_dict = list(custom_stop_dict.iloc[:, 0]) + list(stop_dict.iloc[:, 0])
# 删除停用词
data_cut = data_cut.apply(lambda x: [i for i in x if i not in stop_dict])
#labels标签
labels = message_category_sample.loc[data_cut.index, 'flag']
#data str
data_concat = data_cut.apply(lambda x: ' '.join(x))
data_concat_enter = data_cut.apply(lambda x: '\n'.join(x))
data_tr, data_te, labels_tr, labels_te = train_test_split(data_concat, labels, test_size=0.2)
countVerctorizer = CountVectorizer()
tfidfTransformer=TfidfTransformer()
data_tr = countVerctorizer.fit_transform(data_tr)
X_tr = tfidfTransformer.fit_transform(data_tr.toarray()).toarray()
data_te = CountVectorizer(vocabulary=countVerctorizer.vocabulary_).fit_transform(data_te)
X_te = tfidfTransformer.fit_transform(data_te.toarray()).toarray()
model = GaussianNB()
model.fit(X_tr, labels_tr)
with open('cxjs_vocabulary.pkl','wb') as file:
    pickle.dump(countVerctorizer.vocabulary_,file)
with open('cxjs_tfidf.pkl','wb') as file:
    pickle.dump(tfidfTransformer,file)
```

```
with open('cxjs.model','wb') as file:
    pickle.dump(model, file)
print(model.score(X_te,labels_te))
labels_pr=model.predict(X_te)
print(labels_pr)
labels_te=np.array(labels_te)
print(labels_te)
tp=0
fp=0
tn=0
fn=0
print(labels_pr.__len__())
for i in range(labels_pr.__len__()):
    if labels_te[i]==0 and labels_te[i]==labels_pr[i]:
        tp+=1
    elif labels_te[i]==0 and labels_te[i]!=labels_pr[i]:
        fn+=1
    elif labels_te[i]==1 and labels_te[i]==labels_pr[i]:
        tn+=1
    elif labels_te[i]==1 and labels_te[i]!=labels_pr[i]:
        fp+=1
print(tp,fp,tn,fn)
pr_c=tp/(tp+fp)
re_c=tp/(tp+fn)
print(pr_c*re_c/(pr_c+re_c))
```

附件二：部分留言答复评价明细表

留言编号	留言用户	留言主题	留言时间	问题详细	答复意见	答复时间	答复得分	答复评价
2574	A0009233	关于A市公交站点名称变更的建议	2019/4/23 17:03	建议将“白竹坡路口”更名为“马坡岭小学”，原“马坡岭小学”取消，保留“马坡岭”	网友“A0009233”，您好，您的留言已收悉，现将具体内容答复如下：关于来信人建议“白竹坡路口”更名为“马坡岭小学”，原“马坡岭小学”取消，保留“马坡岭”的问题。公交站点的设置需要方便周边的市民出行，现有公交线路均使用该三处公交站站名，市民均已熟知，因此不宜变更。感谢来信人对我市公共交通的支持与关心。2019年5月5日	2019/5/99:51	83.8269230 8	内容全面、格式完整、答复过时
...	...	....	...	...	...	....	...	...
2759	A00077538	A3区含浦镇马路卫生很差	2019/4/88:37	欢迎领导来A市泥泞不堪的小含浦镇滚泥巴，这个小镇的蓝天保卫战的奇葩战术:没有一台吸尘车，停留在用两台破洒水车把泥巴冲到左边，然后再把泥巴冲到右边，越是上下班，几台破机器越是这样！把整个整个含	网友“A00077538”：您好！针对您反映A3区含浦镇马路卫生很差的问题,A3区学士街道、含浦街道高度重视，现回复如下：您留言中反映的含浦镇在2013年已经析出两个街道，分别是学士街道和含浦街道，鉴于您问题中没有说明卫生较差的具体路段，也没有相应的参照物，同时您也未留下联系方式，请您看到回复后，致电学士街道0731-0000-00000000或者含浦街道0731-0000-00000000反映相关问题。感谢您对我们工作的关心、监督与支持。2019年4月24日	2019/5/910:02	24.525	内容缺失、格式完整、答复过时

				浦搞得泥泞不堪，没有一台干净车，居民民愤极大!!!全国最脏的小镇!职能部门严重造成水资源的浪费!				