“智慧政务”中的文本挖掘应用

摘要

随着网络日新月异，微博、微信、市长信箱等平台的出现，问政方式已经从线下的信笺或者投稿形式，转为了线上平台的留言形式。且留言门槛越来越低，线上问政平台接收的留言量越来越大，也难免会出现大量的垃圾信息。

数据量越来越大，垃圾信息越来越多，显然审核筛选主要问题，已经不能人力来进行，因此，必要用到数据挖掘的手段。文本挖掘是当下热门话题，不论是判断知识产权、著作权的文本相似度识别、还是微博热门话题的筛查、亦或是学术界最常用的论文查重手段，都是文本挖掘的实际应用。

本问探索“智慧政务”的文本挖掘应用。第一问中，分类民众留言模块，按照各部门职能分类民众留言，交由有关部门回复处理。处理过程中，先进行数据清理，删除重复值和缺失值，接着，对文本数据进行分词和删除停用词处理，最后，利用五个算法进行交叉验证，观察其精确度与 f—score，选出最优算法，及 logistic 回归算法。

第二问筛选热门问题，并计算其热度指数，数据清理与文本数据处理方面同第一问，分类通过词频与词频增长率，抽取关键词，并利用关键词分类问题主题。最后以某一热门问题留言的数量与总留言数量的比值，作为最终的热度指数。

第三问为答复意见评价，采用 CountVectorizer 类将分词后的文本进行分词、构建词表以及稀疏矩阵编码的操作。将稀疏矩阵转换为数组应用于协方差矩阵，通过观察协方差矩阵，确定答复与留言问题的相关性，得出相关性低的结果。

关键词： 文本挖掘；交叉验证；logistic 回归；协方差矩阵

目录

目录

1 引言

1.1 问题重述及背景

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、处理问题的一个重要途径，但是随着网络的普及，以及人们生活水平的提高，各类社区意见的相关留言也大幅度提高。这个留言的数据量是非常庞大的，单纯的靠人力来对留言进行划分处理。以及人工整理得到热门问题，也是个不可能完成的任务。

为了提高信息处理得效率，以及减少不必要的劳动力，建立一个基于自然语言处理技术的智慧政务系统，已经成为社会治理创新发展的一个新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。

第一题的群众留言分类问题，是为了得到一个分类体系，分类后得到的不同留言类别，可以直接交给有关的部门处理。仅依靠人工来分类留言的话，庞大的工作量、不高的效率以及出错率高等问题是没有办法避免的。通过自然语言处理，将文本进行分类，建立一个分类系统。这样可以很好的提高解决问题的效率。以及降低人工分类过程中可能出现的错误。

第二题的热点问题挖掘，是为了有助于相关部门及时发现问题，将特定地点的留言进行归类，定义一个合理的热度评价指标，将庞大的数据中的前五个热点问题找出来，并且给出其热度的指数。

并将排名前五的热点问题保存为文件“热点问题表.xls”，以及相应热点问题对应的留言信息，保存为“热点问题留言明细表.xls”。

第三题是对已有的相关部门对留言的答复意见数据中心，对答复的相关性、完整性、可解释性等角度对答复意见的质量给一套评价方案。

1.2 问题分析和创新点

本文基于题目中所给的群众问政留言记录及相关部门对群众留言的答复意见对文本进行挖掘分析，结合机器学习算法构建相应的分类模型。

（一）通过文本处理方法对文本进行预处理。本文首先对中文文本进行分词处理，然后基于分词后的文本进行预处理操作。其次使用不同的预处理方法，对分词后的文本进行处理，将不规则的、杂乱的文本数据转换为算法能够识别的规则的数据结构。最后，将预处理后的数据应用于算法模型进行数据拟合。

（二）机器学习算法分析中文文本数据。首先本文使用五种不同类型的机器学习算法去分析中文文本，根据其泛化性能分析适用于一级标签的文本分类模型。其次本文使用k者交叉验证检验以及基于分类报告对算法模型性能进行分析，针对提升效果最优的机器学习算法，提出适用于一级标签的文本分类模型。

（三）热点问题统计，分为两个步骤，相似问题分类及统计、问题热度指数评价模型实现。相似问题分类，可以根据词频等指标，筛选关键词，做相似文本分类。在分类之后，根据其每一类问题的留言数量与总留言数量作比值计算，以此作为问题的热度指数。

（四）使用协方差矩阵对文本的答复意见与问题进行相关分析。将本小题涉及的中文文本采用相同的预处理方法将其转换为大小和形状相同的矩阵数组，计算协方差矩阵，根据协方差矩阵对答复意见与相应问题进行相关性分析。

2 群众留言分类

本节针对问题一，对群众留言进行分类。首先对群众留言的数据进行预处理，然后通过机器学习算法分析适用于留言分类的算法模型并最终建立群众留言分类模型。

2.1 数据的选取与预处理

2.1.1 数据的选取

随着微信、微博、市长信箱等网络问政平台逐渐成为了政府了解民意的重要渠道，各类社情民意相关文本数据量便不断攀升，本问数据附件 2，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。

2.1.2 检查重复值和缺失值

本文采用Python 进行数据处理，故先将数据导入Python数据分析环境中预备，通过导入结果展示部分数据如图 2-1 所示。

	留言编号	留言用户	留言主题	留言时间	留言详情	一级分类
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\t\t\t\t\t\n\t\t\t\t\tA3区大道西行便道，未管所路口至加油站路段，...	城乡建设
1	37	A000107866	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\t\t\t\t\t\n\t\t\t\t\t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	3742	A00013884	A3区杜鹃文苑小区外的非法汽车检测站要开业了!	2018/12/26 10:13:37	\n\t\t\t\t\t\n\t\t\t\t\tA市政府、市交警支队、市安监局、市环保局、A...	城乡建设

图 2-1 导入数据

从图中可以看出，数据文件给出的数据特征为：留言编号、留言用户、留言主题、留言时间、留言详情以及一级分类。

检查重复值及缺失值，运用到了 Python 的 pandas 模块，其中分别使用 duplicated 函数和 isnull 或 notnull 函数来查看所得数据中的重复值与缺失值的情况，其代码见附录。

通过代码处理结果如下图所示：

```
np.sum(df1.isnull())
```

```
留言编号      0
留言用户      0
留言主题      0
留言时间      0
留言详情      0
一级分类      0
cut       0
label     0
cut_1     0
dtype: int64
```

(a)isnull 检查缺失值情况

```
np.sum(df1.notnull())
```

```
留言编号      495
留言用户      495
留言主题      495
留言时间      495
留言详情      495
一级分类      495
cut       495
label     495
cut_1     495
dtype: int64
```

(b)notnull 检查缺失值情况

```
np.sum(df1.duplicated())
```

```
0
```

(c)duplicated 检查重复值情况

图 2-2 缺失值、重复值情况

从而得知该组数据不存在重复值和缺失值的情况。

2.1.3 删除停用词和文本切分

停用词即指文本中大量出现，但是对分类贡献度不高的词语。删除停用词是信息检索的基本处理方法，也是文本数据挖掘预处理的一环。

目前网上停用词表较多，哈工大停用词表、百度停用词表等均较常见。删除停用词第一步便是准备好停用词表。

中文文本的处理，使用到的是 Python 中的 jieba 模块。利用准备好的停用词表，对留言主题或留言详情进行停用词的过滤和文本切分的操作。分词处理后的结果如下图所示。此处仅展示部分数据。

```
0                A 市 西湖 建筑 集团 占 道 施工 有 安全隐患
1          A 市 在水一方 大厦 人为 烂尾 多年 ， 安全隐患 严重
2     A3 区 杜鹃 文苑 小区 外 的 非法 汽车 检测站 要 开业 了 ！
3    民工 在 A6 区明发 国际 工地 受伤 ， 工 地方 拒绝 支付 医疗费
4                    K8 县 丁字街 的 商户 乱 摆摊
Name: cut, dtype: object
```

图 2-3 文本处理结果

最后，将一级分类标签列的数据进行转换为计算机能够识别的类别，即对一级标签进行映射处理，如图中的 label 列所示。

	留言编号	留言用户	留言主题	留言时间	留言详情	一级分类	cut	label
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\t\t\t\t\t\n\t\t\t\t\tA3区大道西行便道，未管所路口至加油站路段，...	城乡建设	A 市 西湖 建筑 集团 占 道 施工 有 安全隐患	5
1	37	A000107866	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\t\t\t\t\n\t\t\t\t\t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	A 市 在水一方 大厦 人为 烂尾 多年 ， 安全隐患 严重	5
2	3742	A00013884	A3区杜鹃文苑小区外的非法汽车检测站要开业了！	2018/12/26 10:13:37	\n\t\t\t\t\n\t\t\t\t\tA市政府、市交警支队、市安监局、市环保局、A...	城乡建设	A3 区 杜鹃 文苑 小区 外 的 非法 汽车 检测站 要 开业 了 ！	5
3	5315	A0009647	民工在A6区明发国际工地受伤，工地方拒绝支付医疗费	2018/7/17 7:13:58	\n\t\t\t\t\n\t\t\t\t\t胡书记，您好，感谢您百忙之中查看这份留言。我...	城乡建设	民工 在 A6 区明发 国际 工地 受伤 ， 工 地方 拒绝 支付 医疗费	5
4	100026	A00047862	K8县丁字街的商户乱摆摊	2017/10/14 14:25:14	\n\t\t\t\t\n\t\t\t\t\t K8县丁字街的商户乱摆摊，前段时间丁字街...	城乡建设	K8 县 丁字街 的 商户 乱 摆摊	5

图 2-4 一级标签转换

2.2 算法选取

本题解决留言分类问题，解决此类问题算法均为分类算法，其中选取最常用的四类分类算法，分别是 K 近邻分类算法模型（KNN）、决策树、随机森林、朴素贝叶斯算法以及 Logistic 回归。

这里用到了交叉验证的思想，即在实际的训练中，训练的结果对于训练集的拟合程度通常还是挺好的，但是对于训练集之外的数据的拟合程度通常就差强人意。因此并不会把所有的数据集都拿来训练，而是分出一部分来对训练集生成的参数进行测试，相对客观的判断这些参数对训练集之外的数据的符合程度，故为了保证训练效果通常会将一整个训练集分为三个部分：训练集、评估集、测试集。

以下分别对所用到的算法进行一个简单的介绍与概括。

2.2.1 K 近邻分类算法

K 近邻分类方法本身属于一种无监督学习方法，是不需要对训练数据进行建模，而是需要分类测试时才进行。核心思想是在样本特征空间中选择K个相邻的样本，如果 K 个相邻的样本大多数属于某一类，则该测试样本也属于这个类*。

（1）算法思想

在特征空间中，如果一个样本附近的k个最近样本的大多数属于某一个类别，则该样本也属于这个类别。即：给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最近的k个实例，这k个实例的多数属于某个类，就把该输入实例分类到这个类中。给定测试样本，基于某种距离度量找出训练集中与其最靠近的k个训练样本，然后基于这k个"邻居"的信息来进行预测。

（2）算法实现

算法实现分为两个阶段，训练阶段及分类阶段。训练阶段，主要包括训练数据的离散化、读取、存储三个方面。数据离散化，是针对数据的m维属性，对每个属性进行单独的离散化，并将离散化后的数据写入一个表中，以为下一步的数据读取做准备；数据的读取和存储主要是将表中数据读入数组存储，以便程序执行处理。为了确定分类的效果将离散后的数据分成了十份，用以进行十重交叉测试试验。

分类阶段，主要是将训练阶段处理完毕的数据进行交叉验证，具体步骤：a.准备数据作为待分类实例， 计算此实例与其它实例的欧氏距离；b.比较各个距离的大小，选取最近邻的K个实例；c.查看上步中K个实例的分类标记，采用多数表决的方式确定待分类实例的类别；d.应用十重交叉测试试验方法，重复以上各步骤。

2.2.2 决策树

决策树是一种树形结构的算法模型，它是一种常用于分类和回归任务的监督学习算法。决策树算法的本质是在一系列的if/else规则中进行不断地学习，并最终预测得出尽量能够满足于人们意愿的结论。

（1）算法思想

一般来说，普通的一棵决策树的树形结构是由根结点、内部结点和叶结点以及连接结点之间的边构建而成。其中，根结点是样本数据集合的全集，叶结点则是从根结点到该叶结点所对应的决策结果，而内部结点则是对应某个分支路径的可能属性的判断。利用if/else规则构建一个模型，进而可以使用监督学习从数据中学习模型构建决策树，最终利用构建好的决策树对未知的新数据进行处理，而不需要人工干预构建模型。决策树的构建过程采用把一个复杂问题分成两个或更多个相似的子问题进行逐步处理的“分而治之”的思想。

构造决策树采用的是递归的方法。算法递归地在所有属性集合中，搜索所有可能的if/else规则，针对目标变量寻找在所有未被划分的属性集合中信息量能够是最大的那个属性测试，根据该属性对数据集进行相应的划分。在一般的情况下，随着按照属性递归划分数据集的过程不断进行，我们尽可能的想要让在决策树的分支结

点中所包含的样本数据尽可能是同一个类别的，也就是说：分支结点的“纯度”随着划分的进行而越来越高。

（2）算法实现

在机器学习中，用于度量数据样本集合的纯度最常用的指标之一是 entropy（信息熵）。而有名的ID3决策树分类算法是以信息增益这一指标为准则来进行特征选择划分属性的，而信息增益的计算方式是建立在信息熵的计算基础上的。

假定在含有$|y|$个不同类别标签的数据样本集合D中，记第k类样本数据在数据样本集合D中所占的比例为$p_k\left(k=1,2,3,...,|y|\right)$，则可以将数据样本集合的信息熵定义为：

$$Ent(D)=-\sum_{k=1}^{|y|}p_k log_2 p_k$$

其中，$|y|$表示样本的类别。信息熵的值要是越小，则表明的纯度越高。

假定在数据样本集合D中，有不连续的属性集合a，将属性集合a的V个可能的取值记为$\{a_1,a_2,...,a_v\}$，如果根据属性集合a的不同取值对数据样本集合D进行相应的属性特征划分，则可以从数据样本集合D划分出V个分支结点，其中在第v个分支结点中包含在数据样本集合D中所有属性集合a上取值为a_v样本数据，将其记为集合D_v。根据信息熵计算公式（2-1）式可以计算出集合D_v的信息熵，然后由于在数据样本集合D划分出的不同的分支结点上所包含的样本数是不同的，也就是说属性集合a中的不同取值的子集所包含的样本数是不同的，因此，给每个分支结点赋予各自的权重$|D_v|/|D|$，即在不同的分支结点中，如果某个分支结点所包含的样本数越多，则该分支结点对属性划分的影响越大，于是计算出用属性a对数据样本集D进行属性划分所获得的“信息增益”为：

$$Gain(D,a)=Ent(D)-\sum_{v=1}^{V}\frac{|D_v|}{|D|}Ent(D_v)$$

在一般情况下，信息增益的值要是越大，则说明根据属性集合来对数据样本集合进行划分所获得的“纯度”的提升将会越大。

决策树构建完成的标志是，在构造决策树的过程中随着分支结点和叶结点的不断产生直到决策树的所有叶结点都是纯的叶结点时才会停止。这一行为导致构造的决策树模型将会非常复杂，对于训练数据的拟合将会无限逼近于 100%，这会导致对训练数据高度过拟合的现象。在机器学习理论中，有防止过拟合的的两种常见策略：第一种是提前停止树的划分和增长，通常作法是直接限制树的深度，即预剪枝方法；另一种是先对树进行构造，然后折叠或删除信息量很少的结点，即后剪枝方法。

2.2.3 随机森林

随机森林（Random Forest，RF）的出现是为了解决决策树容易对训练数据过拟合的问题，它是以决策树为基础的集成学习算法[23]。因此，随机森林算法的本质是许许多多的决策树的集成，也就是说本质源于决策树算法，但是构成随机森林的每棵决策树之间都略有差异，并且 Beriman 通过概率论的大数定律知识证明了随机森林算法是不容易过拟合的。

（1）算法思想

构成随机森林的每一棵决策树对测试数据集合的预测性能都能够达到相对较好的程度，但不同的树之间可能存在对训练数据集合中的部分数据出现过拟合的现象。如果随机森林算法构造了很多不同的决策树，并且每棵决策树对未知数据的泛化性能都在可接受的范围内，但每棵树之间都存在差异且都以各自不同的方式对训练数据集合中的部分数据存在过拟合现象，那么针对不同的过拟合现象可以通过对这些树的预测结果取它们的平均值的方法来降低树的过拟合。该降低过拟合的方法既能减少树对训练数据集合的过拟合问题又能保持树的泛化能力。

对于随机森林算法的思想，为了确保每棵树之间存在差异性，随机森林中树的随机化方式有两种方法：一种是通过从数据集合中选择用于构造树的数据点进行随机化，另一种是通过从数据属性集合中选择每次划分数据集属性的特征进行随机化。这两种随机森林算法随机化方法是随机森林算法与决策树算法之间的主要差异。

（2）算法实现

根据随机森林随机化的方法构建随机森林算法模型。首先，算法需要确定用来构造随机森林的树的个数。这些构成随机森林的树在构造的时候是相互独立的，算法会对每棵决策树做出不同的随机选择，为的是确保每棵树之间都存在差异性。而在构造树的时候，为了实现树之间的差异性，对数据进行自助采样的方法，从全部的数据点中有放回地重复抽取一个样本，并且样本数与数据点全集的大小是一样的。也就是说，假设有n个数据点，那么从这n个数据点中，有放回地抽取n次，将抽取的数据构成一个新的集合，将这个集合作为构造树的数据样本点。自助采样方法的特点是，该方法会创建一个新的且与原数据集大小相同的样本数据集合，但是有些样本数据点会缺失，而且缺失的样本数据大约是原数据集合的 1/3，而有些样本数据点则会存在重复。

其次，利用通过自助采样方式获得的新的数据集来构建决策树。在构建决策树的过程中，不能和构建传统决策树的方法完全一样，需要稍作修改。在结点处，随机选择特征属性集合的一个子集，在该子集中寻找信息量最大的属性来对结点进行划分。而选择的特征属性集合中属性的个数由参数来控制，该参数如果设置的较大，那么构成随机森林的树将会十分相似，利用比较独特的特征就可以很好的拟合数据；该参数如果设置的较小，那么构成随机森林的树将会存在很大的差异，而为了拟合数据的需要，可能会导致树的深度将会很大。因此，对随机选择特征属性个数的这个参数的控制将会对随机森林的构造造成一定的影响。假设该参数为k，特征属性个数为d，若令$k=d$，则构建随机森林的树将与传统决策树的构建方式相同，没有差异性；若$k=1$，则是随机选择一个属性用于划分，将无法选择哪个特征属性信息量更大；因此，一般情况下，推荐的值为$k=\log_2 d$。最后，当每一棵树构造完成后，将其集成在一起就是一个随机森林模型。

随机森林构造完成后对未知的新数据的预测，首先对构成森林的每棵决策树进行预测，然后，根据具体问题具体分析。在回归问题中，可以将每棵树对未知的新数据的预测结果取其平均值作为随机森林模型的最终预测结果；在分类问题中，则是将树对未知的新数据的预测给出的每个可能的分类标签的预测概率取其平均值，将其中预测概率最大的类别作为最终预测结果。

2.2.4 朴素贝叶斯算法

基于贝叶斯定理与特征条件独立假设的分类方法。

贝叶斯方法是以贝叶斯原理为基础，使用概率论统计的知识对样本数据集进行分类。由于其有着坚实的数学基础，贝叶斯分类算法的误判率是很低的。贝叶斯方法的特点是结合先验概率和后验概率，即避免了只使用先验概率的主观偏见，也避免了单独使用样本信息的过拟合现象。贝叶斯分类算法在数据集较大的情况下表现出较高的准确率，同时算法本身也比较简单。

（1）算法思想

朴素贝叶斯算法是在贝叶斯算法的基础上进行相应的简化，即假定给定目标值是属性之间相互条件独立。也就是没有哪个属性变量对于决策结果来说占有较大的比重，也没有哪个属性变量对于决策结果占着较小的比重。虽然这个简化方式在一定程度上降低了贝叶斯分类算法的分类效果，但是在实际的应用场景中，极大地简化了贝叶斯方法的复杂性。

朴素贝叶斯分类是以贝叶斯定理为基础并且假设特征条件之间相互独立的方法，先通过已给定的训练集，以特征词之间独立作为前提假设，学习从输入到输出的联合概率分布，再基于学习到的模型，输入 X 求出使得后验概率最大的输出 Y。

（2）算法实现

设有样本数据集 $D=\{d_1,d_2,...,d_n\}$，对应样本数据的特征属性集为 $X=\{x_1,x_2,...,x_d\}$，类变量为 $Y=\{y_1,y_2,...,y_m\}$，即 D 可以分为 y_m 类。其中 $x_1,x_2,...,x_d$ 相互独立且随机，则 Y 的先验概率为 $P_{prior}=P(Y)$，Y 的后验概率为 $P_{post}=P(Y|X)$。由朴素贝叶斯算法可得，后验概率可以由先验概率 $P_{prior}=P(Y)$、证据 $P(X)$、类条件概率 $P(X|Y)$ 计算出：

$$P(Y|X)=\frac{P(Y)P(X|Y)}{P(X)}$$

朴素贝叶斯基于各特征之间相互独立，在给定类别为 y 的情况下，上式可以进一步表示为下式：$P(X|Y=y)=\prod_{i=1}^{d}p(x_i|Y=y)$；由以上两式可以计算出后验概率为：

$$P_{post}=P(Y|X)=\frac{P(Y)\prod_{i=1}^{d}p(x_i|Y=y)}{P(X)}$$

由于 $P(X)$ 的大小是固定不变的，因此在比较后验概率时，只比较上式的分子部分即可。因此可以得到一个样本数据属于类别 yi 的朴素贝叶斯计算为：

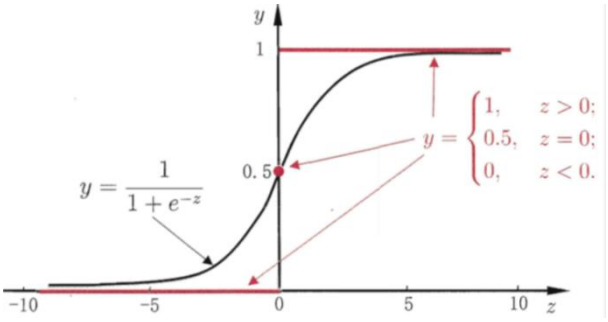
$$p(y_i|x_1,x_2,...,x_d)=\frac{p(y_i)\prod_{j=1}^{d}p(x_j|y_i)}{\prod_{j=1}^{d}p(x_j)}$$

2.2.5 Logistic 回归

逻辑回归假设数据服从伯努利分布，通过极大似然函数的方法，运用梯度下降来求解参数，来达到将数据二分类的目的。

考虑二分类任务，其输出标记为 0, 1；线性回归模型产生的预测值 $z=w^Tx+b$ 是实值，于是将实值转换为 0, 1，理想的方法是使用“单位越阶函数”。

单位越阶函数：$y=\begin{cases}0, z<0\\0.5, z=0\\1, z>0\end{cases}$，即若预测值 z 大于 0 则预测为正例，小于 0 则预测为反例，预测值为临界值则可以任意预测。如图所示，从图中可以看出单位越阶函数不连续。



对数几率函数（越阶函数）： $y=\frac{1}{1+e^{-z}}$，是一种 sigmoid 函数（形似S的函数），如上图黑色曲线。它将 z 值转化为一个接近 0 或 1 的 y 值，并且其输出在 z=0 的附近变化的很陡。

将对数几率函数应用于线性回归，得到对数几率回归模型 logistic regression。

$$y=\frac{1}{1+e^{-(w^Tx+b)}};\ln\frac{y}{1-y}=w^Tx+b$$

该式子是在使用线性回归模型的预测结果去逼近真实标签的对数几率；将对数几率函数应用于线性回归，得到对数几率回归模型 logistic regression。

2.3 分类模型分析

2.3.1 基于留言内容的一级标签分类模型

以留言内容作为分类依据，建立基于留言内容的一级标签分类模型。将中文文本表示为结构化的能够让自己算计识别的数据，是文本特征表示阶段所要处理的问题。该处理过程，使用 Python 机器学习的 scikit-learn 模块中，CountVectorizer 类，此类处理文本的过程为分词、构建词表以及稀疏矩阵编码。该过程将非结构化的中文文本转化为结构化的稀疏矩阵用于分类算法模型。CountVectorizer 类只考虑每种词汇在该训练集合中出现的频率，即将文本中的词语转换为词频矩阵。选取留言主题使用 jieba 文本预处理后的文本，对其使用 CountVectorizer 类将其转换为词频矩阵。

将转换得到的词频矩阵应用于算法模型中。分别使用 K 近邻分类算法模型、决策树、随机森林、朴素贝叶斯算法以及 Logistic 回归，通过实验对比分类报告中的数据建立基于留言主题的一级标签分类模型，分类报告的数据主要包含的信息为：准确率(Precision)、召回率(Recall)以及 f1-分数(f1-score)。最后再分析算法模型。

首先，从算法精度的角度分析。如表所示，从表中可以看出，K 近邻分类算法的精度是最低的，而逻辑回归分类算法模型精度是最高的，因此，优先考虑选择 Logistic 回归分类算法模型作为分类模型。

表 2-1 分类算法稳定型

tree average	p:0.7414221955817581
knn average	p:0.32749135203731433
mulNB average	p:0.8088920233350902
forest average	p:0.7893458985969394
logistic average	p:0.8281458830877183

接着，从准确率、召回率以及 f1 评分等角度分析，计算出如下表所示各算法分类报告。

表 2-2 各算法分类报告

		Precision	Recall	F1-score	Support
KNN	1	0.24	0.42	0.31	150
	2	0.58	0.36	0.45	497
	3	0.13	0.79	0.23	199
	4	0.84	0.26	0.40	317
	5	0.62	0.21	0.32	506
	6	0.48	0.31	0.38	390
	7	0.82	0.11	0.20	244
	Accuracy			0.32	2303
	Macro avg	0.53	0.35	0.32	2303
	Weighted avg	0.57	0.32	0.35	2303
Tree	1	0.85	0.80	0.82	150
	2	0.76	0.81	0.78	497
	3	0.78	0.69	0.73	199
	4	0.76	0.64	0.70	317
	5	0.63	0.76	0.69	506
	6	0.83	0.78	0.80	390
	7	0.79	0.69	0.74	244
	Accuracy			0.75	2303
	Macro avg	0.77	0.74	0.75	2303
	Weighted avg	0.75	0.75	0.75	2303
Forest	1	0.96	0.83	0.89	150
	2	0.82	0.86	0.84	497
	3	0.79	0.77	0.78	199
	4	0.90	0.69	0.78	317
	5	0.66	0.86	0.75	506
	6	0.89	0.81	0.85	390
	7	0.88	0.73	0.80	244
	Accuracy			0.81	2303
	Macro avg	0.84	0.79	0.81	2303

	Weighted avg	0.82	0.81	0.81	2303
mulNB	1	0.91	0.66	0.76	150
	2	0.77	0.92	0.84	497
	3	0.86	0.69	0.77	199
	4	0.86	0.71	0.78	317
	5	0.75	0.88	0.81	506
	6	0.87	0.85	0.86	390
	7	0.91	0.78	0.84	244
	Accuracy			0.82	2303
	Macro avg	0.85	0.78	0.81	2303
	Weighted avg	0.83	0.82	0.82	2303
Logistic	1	0.98	0.82	0.89	150
	2	0.86	0.89	0.87	497
	3	0.84	0.81	0.83	199
	4	0.86	0.73	0.79	317
	5	0.74	0.88	0.80	506
	6	0.86	0.86	0.86	390
	7	0.91	0.78	0.84	244
	Accuracy			0.84	2303
	Macro avg	0.86	0.82	0.84	2303
	Weighted avg	0.84	0.84	0.84	2303

观察上述分类报告，分类模型通常使用 F-score 进行评价，而对比五大模型的 F 评分，依旧是 Logistic 回归模型的精确度最高，及五种分类模型中最优的分类算法模型。

2.3.2 基于留言详情的一级标签分类模型

使用于 2.3.1 相同的分析方法与思路对留言详情进行分析，从准确率、召回率以及 f1 评分等角度分析，计算出如下表所示各算法分类报告。

表 2-3 一级标签各算法分类报告

		Precision	Recall	F1-score	Support
KNN	1	0.44	0.34	0.38	150
	2	0.68	0.66	0.67	497
	3	0.28	0.60	0.38	199
	4	0.28	0.63	0.39	317
	5	0.66	0.36	0.46	506
	6	0.87	0.43	0.58	390
	7	0.84	0.31	0.46	244
	Accuracy			0.49	2303
	Macro avg	0.58	0.48	0.47	2303
	Weighted avg	0.62	0.49	0.50	2303
Tree	1	0.62	0.65	0.63	150
	2	0.82	0.82	0.82	497
	3	0.72	0.70	0.71	199
	4	0.67	0.66	0.66	317
	5	0.67	0.72	0.69	506
	6	0.86	0.87	0.86	390
	7	0.79	0.66	0.72	244
	Accuracy			0.75	2303
	Macro avg	0.73	0.72	0.73	2303
	Weighted avg	0.75	0.75	0.75	2303
Forest	1	0.77	0.55	0.64	150
	2	0.73	0.92	0.82	497
	3	0.82	0.65	0.73	199
	4	0.80	0.60	0.68	317
	5	0.67	0.80	0.72	506
	6	0.90	0.84	0.87	390
	7	0.87	0.74	0.80	244
	Accuracy			0.77	2303
	Macro avg	0.79	0.73	0.75	2303
	Weighted avg	0.78	0.77	0.77	2303

mulNB	1	0.97	0.52	0.68	150
	2	0.84	0.92	0.88	497
	3	0.89	0.76	0.82	199
	4	0.88	0.74	0.80	317
	5	0.79	0.90	0.84	506
	6	0.87	0.92	0.89	390
	7	0.91	0.95	0.93	244
	Accuracy			0.85	2303
	Macro avg	0.88	0.82	0.83	2303
	Weighted avg	0.86	0.85	0.85	2303
Logistic	1	0.84	0.82	0.83	150
	2	0.89	0.94	0.91	497
	3	0.91	0.81	0.86	199
	4	0.86	0.80	0.83	317
	5	0.81	0.88	0.84	506
	6	0.93	0.91	0.92	390
	7	0.94	0.88	0.91	244
	Accuracy			**0.88**	2303
	Macro avg	**0.88**	**0.86**	**0.87**	2303
	Weighted avg	**0.88**	**0.88**	**0.88**	2303

观察上述分类报告，分类模型通常使用 F-score 进行评价，而对比五大模型的 F 评分，Logistic 回归模型的精确度最高，而朴素贝叶斯分类算法的效果次之。

因此，不管是以留言主题还是以留言详情为研究对象，对于关于留言内容的一级标签分类模型，从上述实验对照中可以看出，logistic 回归模型的精确度最高、F 分数最高，即：logistic 回归模型适用于关于留言内容的一级标签分类模型。

2.3.3 不同文本特征表示的 logistic 回归分类模型分析

在一级标签分类模型中，采用 Logistic 回归模型根据不同的特征表示方法：对分类模型进行分析，如下表所示。

表 2-4 CountVectorizer 类与 TfidfVectorizer 类的分析结果

Logistic		Precision	Recall	F1-score	Support
CountVectorizer类--主题	1	0.96	0.82	0.88	150
	2	0.85	0.88	0.87	497
	3	0.85	0.82	0.84	199
	4	0.88	0.72	0.79	317
	5	0.73	0.88	0.80	506
	6	0.86	0.86	0.86	390
	7	0.92	0.80	0.85	244
	Micro avg	0.84	0.84	0.84	2303
	Macro avg	0.86	0.83	0.84	2303
	Weighted avg	0.85	0.84	0.84	2303
CountVectorizer类--详情	1	0.84	0.82	0.83	150
	2	0.89	0.94	0.91	497
	3	0.91	0.81	0.86	199
	4	0.86	0.80	0.83	317
	5	0.81	0.88	0.84	506
	6	0.93	0.91	0.92	390
	7	0.94	0.88	0.91	244
	Micro avg	0.88	0.88	0.88	2303
	Macro avg	0.88	0.86	0.87	2303
	Weighted avg	0.88	0.88	0.88	2303
TfidfVectorizer类--主题	1	0.99	0.69	0.82	150
	2	0.81	0.90	0.85	497
	3	0.89	0.72	0.80	199
	4	0.89	0.67	0.77	317
	5	0.65	0.90	0.75	506
	6	0.89	0.84	0.86	390
	7	0.92	0.70	0.80	244
	Micro avg	0.81	0.81	0.81	2303
	Macro avg	0.86	0.77	0.81	2303
	Weighted avg	0.83	0.81	0.81	2303

TfidfVectorizer 类--详情	1	0.98	0.64	0.77	150
	2	0.87	0.94	0.90	497
	3	0.91	0.78	0.84	199
	4	0.90	0.78	0.84	317
	5	0.74	0.91	0.82	506
	6	0.91	0.90	0.91	390
	7	0.93	0.81	0.87	244
	Micro avg	0.86	0.86	0.86	2303
	Macro avg	0.89	0.82	0.85	2303
	Weighted avg	0.87	0.86	0.86	2303

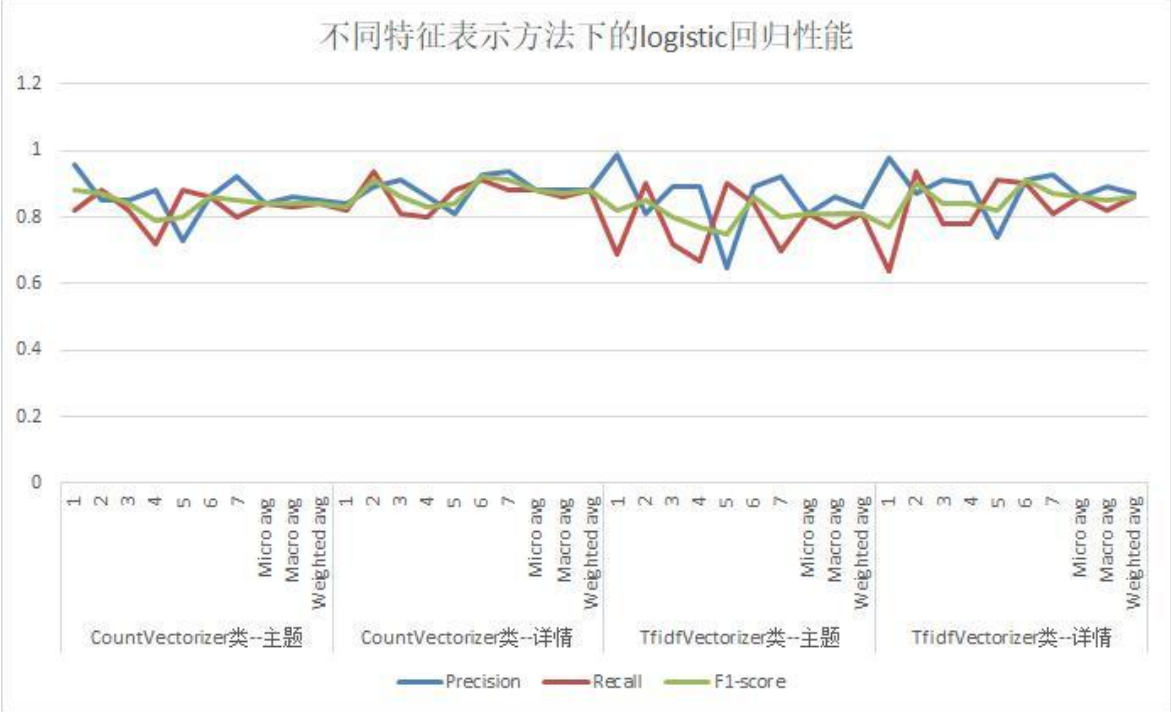


图 2-1 不同特征表示方法下的 logistic 回归性能

从上表可知，在本问题探究中，使用 CountVectorizer 类或者 TfidfVectorizer 类对文本进行特征表示并结合分类算法模型进行实验对照，可以发现两种算法导致的分类算法模型的泛化性能相差不大，其 F1-分数在 0.85 左右徘徊。

2.3.4 Logistic 回归模型建立与优缺点分析

logistic 回归模型的优点为：实现简单、分类时计算量小、速度快、计算代价不高、易于理解与实现。Logistic 回归模型的缺点为：当特征空间很大时，逻辑回归模型的分类性能不佳、容易出现欠拟合的情况。

Logistic 回归模型一般是用来处理二分类问题的分类模型，在处理多分类问题的时候，可以使用 OVR 或 OVO 方法。OVR 方法是若某个分类为 N 分类，将某一类与剩余的类比较作为二分类题，N 个类别进行 N 次分类，得到 N 个分类模型，当给定一个新的样本点时，求出每种二分类的概率，概率最高的一类作为新样本点的预测类别。OVO 方法是某个分类为 N 分类，将某一类与另一类作为二分类问题，总共可以分为C_n^2种不同的二分类模型，当给定一个新的样本点时，求出每种二分类的概率，概率最高的一类作为新样本点的预测类别。在 python 的 scikit-learn 中默认是支持多分类问题的，且多分类方法默认为 OVR。

对于关于留言内容的一级标签分类模型从上述分类报告可以看出，logistic 回归模型是适用于该问题的分类模型，其 f1 分数为 0.88 左右，说明分类效果很好，因此建立基于 logistic 回归的关于留言内容的一级标签分类模型。

3 热点问题挖掘

本节针对问题二，对群众留言当中的热点问题进行筛选汇总，根据时间区间内的，被民众多次反馈及赞同的情况，建立热度指数评价模型，以此作为筛选热点问题的依据。

网络文本碎片化特性，使海量的群众反馈信息分散，且存在着大量的垃圾信息，这些垃圾信息成为了影响主要问题的噪声，因此热点问题关键词筛选之前，需要进行数据的清洗，以及同第一问的分词等操作。

通过词语的词频、词频增长率等特征计算方法抽取关键词，利用关键词作为主题特征来表示每个文本，以此来分类以关键词为特征的文本主题。

（1）词频

词频，即词语出现的频率，如果一个有效词汇在某篇文档中的频次够高，那么足以说明该词汇在文档有着重要地位.彭泽映等*通过研究发现，在一段时间窗口中，词语出现的次数呈现长尾现象，即高频词的词汇量少，而低频词的词汇量多。计算公式如下:

$$F_{ij} = f_{ij}$$

F_{ij}代表第 j 个时间窗内的第 i 个词项的词频，f_{ij}即为词频数值。

（2）词频增长率

词频增长率可以衡量一个词项在单位时间窗内的变化情况，可以认为,在相邻的时间窗内，某-一个词项其增长率高，那么在这个时间窗内，该词很可能代表着一个主题的数目在增多。但是，这隐含着一些问题，例如在时间段早上 6 点到 8 点，晚

上 10 点到 12 点的时间窗内，“早安”、“晚安”等状态词被网民频繁发布，这些词并不是能够组成主题的关键词，会扰乱视听，因此，本文考察在相对时间窗内，词项的词频增长率，即:若以天为单位，那么相对时间窗就是前一天的时间窗:若以小时为单位,那么相对时间窗就为前-天的该时间段的时间窗，计算公式如下:

$$FK_{ij} = \frac{f_{ij}}{1+ f_{ij'}}$$

其中 FK_{ij} 表示词项 w_i 在时间窗 j 内的词频增长率， f_{ij} 为其词频， $f_{ij'}$ 为相对时间窗内的词频。

在此之后，利用这两项指标，分类出不同问题的关键词，再利用关键词进行问题的分类。并统计相关问题的数量，及某问题的留言数，并记留言数与总留言数量的比值，为热度指数。

4 答复意见评价

本节针对问题三，从相关部门对留言的答复，分析其相关性、完整性、可解释性等，通过建立答复意见质量的评价方案，对相关部门及问题的答复质量，进行评估。

首先，将附件 4 中的留言详情以及答复意见分别将其进行去除停用词以及分词的操作。其次，采用 CountVectorizer 类将分词后的文本进行分词、构建词表以及稀疏矩阵编码的操作。最后，将稀疏矩阵转换为数组应用于协方差矩阵的计算，计算结果如下图所示。从该协方差矩阵中可以看出答复意见与相应问题的相关性并不大。

```
(array([[ 1.        , -0.00148768,  0.02611574, ...,  0.02234033,
          0.02442559,  0.01937301],
        [-0.00148768,  1.        ,  0.01215768, ...,  0.01262868,
          0.01386274,  0.02574814],
        [ 0.02611574,  0.01215768,  1.        , ...,  0.08266033,
          0.0140979 ,  0.00901935],
        ...,
        [ 0.02234033,  0.01262868,  0.08266033, ...,  1.        ,
          0.02246919,  0.06964617],
        [ 0.02442559,  0.01386274,  0.0140979 , ...,  0.02246919,
          1.        ,  0.16791888],
        [ 0.01937301,  0.02574814,  0.00901935, ...,  0.06964617,
          0.16791888,  1.        ]]),
 array([[1.        , 0.16728905, 0.10534854, ..., 0.05171803, 0.08401223,
         0.11411414],
        [0.16728905, 1.        , 0.03797838, ..., 0.03027652, 0.00607379,
         0.16143183],
        [0.10534854, 0.03797838, 1.        , ..., 0.01706682, 0.01770531,
         0.05627563],
        ...,
        [0.05171803, 0.03027652, 0.01706682, ..., 1.        , 0.01847686,
         0.05837539],
        [0.08401223, 0.00607379, 0.01770531, ..., 0.01847686, 1.        ,
         0.16975778],
        [0.11411414, 0.16143183, 0.05627563, ..., 0.05837539, 0.16975778,
         1.        ]]))
```

图 4-1 相关性分析的协方差矩阵

参考文献

[1]赵京胜, 宋梦雪, 高祥. 自然语言处理发展及应用综述[J]. 信息技术与信息化, 2019, 000(007):142-145.

[2]许晋军, 苏新宁. 信息搜索引擎综述[J]. 计算机系统应用, 1999, 000(004):22-24.

[3]孙晓蕾. 数字化图书馆建设现状与发展趋势[J]. 黑龙江科学, 2019(17).

[4]胡雅萌. 基于词典与Doc2vec融合的文本情感分析研究[D]. 武汉邮电科学研究院.2018.

[5]朱梦. 基于机器学习的中文文本分类算法的研究与实现[D]. 2019.

[6]汪岿, 刘柏嵩. 文本分类研究综述[J]. 数据通信, 2019(3).

[7]王婧雅. 微博数据挖掘可视化系统的设计与实现[D]. 吉林：吉林大学,2017.

[8]冯莉. 面向英文电影评论的文本情感倾向性分类研究[D]. 大连海事大学.2013.

[9]张磊. 文本分类及分类算法研究综述[J]. 电脑知识与技术, 2016(34):231-232+238.

[10]秦春秀, 祝婷, 赵捧未, et al. 自然语言语义分析研究进展[J]. 图书情报工作, 2014, 58(22):130-137.

[11]张庆庆. 基于机器学习的文本情感分类研究[D].2016.

[12]严军超, 赵志豪, 赵瑞. 基于机器学习的社交媒体文本情感分析研究[J]. 信息与电脑(理论版), 2019(20)..