群众留言文本挖掘

摘要

群众留言文本数据是政务工作中的一种重要数据形式，通过分析，政府可以直接了解到群众的生活状况以及对管理制度的建议等，提取群众留言所关注的热点问题，并对群众留言按内容（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）进行分类以便为群众精准服务，了解群众情感倾向，增强政府的管理效率。但大数据条件下对大量文本数据进行有效的挖掘和利用已经变得越来越困难。本文主要是先运用词频统计（TF）及逆文本频率指数统计（IDF）技术来寻找群众留言分类的关键词进而画出词云，用卡方检验的方法找出每个分类中关联度最大的两个词语和词语对来反映分类的主题并用同样的方法挖掘热点问题。

关键词：群众留言、文本数据挖掘、词频、卡方检验、贝叶斯检验

业务介绍

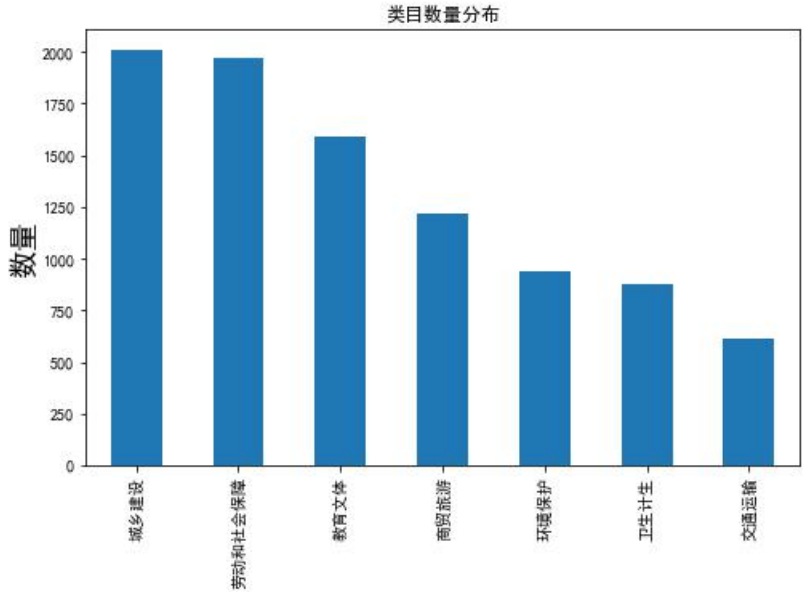
近年来，随着各网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道，各类社情民意相关的文本数据量不断攀升，给以往主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了极大挑战。同时，随着大数据、云计算、人工智能等技术的发展，建立基于自然语言处理技术的智慧政务系统已经是社会治理创新发展的新趋势，对提升政府的管理水平和施政效率具有极大的推动作用。

智慧政务系统是基于文本数据挖掘下的研究，而文本数据的特点就是信息蕴含在自由文本中、没有结构化字段可供查询以及无法直接进行统计分析。要想对这些文本进行分析，就要对它进行预处理，将非结构化的文本数据转成结构化的信息以供分析。文本处理中关键词的抽取目标是识别文本中最重要的一些词即给定一段文本，从中抽取出最能体现文本内容的关键词。而热点问题则是对群众留言的分类进行进一步提取，更有利于政府关注民生。

数据描述

本文所用的留言文本数据使用 jupyter 导入现存数据和网络爬虫获取，所谓网络爬虫，是按照一定的规则，自动地抓取网络信息的程序或者脚本。由于本文主要分析对象是群众留言文本数据，并关注实证层面，所以采用基于留言分类的爬取方式，即选定某几条留言垂直爬取目标网页的留言文本等数据，采用 Python 语言编写网络爬虫或利用其他开源爬虫程序完成数据获取、解析、调用与存储。

现存数据中附件二导入后共有 9210 条文本数据记录，分类过程中用图形化方式来查看不同类别留言的分布情况如下：



数据建模

（一）描述性分析

1. 基本建模方法：

（1）TF、IDF 词频统计技术：

对于一个文本数据，如果仅用某个词语在文本中出现的次数来定义词频，那么词语在长文本中的词频一般会大于这个词在短文本中的词频，因此不能说这个词就是长文本数据的一个好标识词，为了避免这种情况可以使用 TF 技术统计词频，TF=某词语出现次数/总词语数；但像一些常用词：是、的、得等在文本中出现的频率本身就高，如果只用 TF 来定义词频，无法避免此类问题，再引用 IDF 统计技术，IDF=lg（总文本记录数/出现某个词语的文本记录数），这样统计出的词频效果会更好。

（2）特征选择-卡方检验

在文本数据分类中单纯使用 TF-IDF 词频统计来判断文本数据特征是否有区分度是不够的，它没有考虑特征词在不同类间的分布。文本数据中能观测到的量其实只有两个：词频和文本记录频率，所有的方法都是以这两个量为计算基础。而卡方检验的基本思想就是通过观察实际值与理论值（预测值）的偏差程度并构造卡方统计量，从而判定理论的正确性，偏差程度的计算公式为：$\sum_{i=1}^{n}\frac{(x_i-E)^2}{E}$，其中 E 为理论值（预测值），x 为实际值。

（3）贝叶斯分类器

这是一种基于贝叶斯公式：$p(B_i|A)=\frac{p(B_i)p(A|B_i)}{\sum_{j=1}^{n}p(B_j)p(A|B_j)}$ 并假设各特征相互独立的分类方法，其基本方法是：使用特征向量来表征某个实体，并在该实体上绑定一个标签来代表其所属的类别。贝叶斯分类器也就是条件概率即给定一个实体，求解这个实体属于某一类的概率，这个实体用一个 n 维向量来表示，向量中的每一个元素表示相互独立的特征值的量。

（4）混淆矩阵评价指标及 F-score 评价模型

首先，混淆矩阵是指实际类别结果和预测分类结果的对比矩阵，在标签分类问题上常用于对验证数据集的各类分类结果的查看。当文本数据中只有一个二分类指标时，只要用 F-score 评价指标即可实现：$F_1=\frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$，其中 P 称为查准率（精确率），即正确预测为正的占全部预测为正的比例：$P=\frac{TP}{TP+FP}$；R 为查全率（召回率），即正确预测为正的占实际为正的比例：$R=\frac{TP}{TP+FN}$。TP、FP、FN 参数如下表所示：

实际类别	预测类别正例	预测类别负例
正例	TP	FN

负例	FP	TN

若有 n 个二分类混淆矩阵就要引入宏平均和微平均的概念，所谓宏平均就是先对每一个类统计指标值运算，再对所有类求算数平均值；而微平均是对文本数据集中每一个实例不分类别进行统计建立全局混淆矩阵，而后计算相应指标；在代入对应的 F-score 评价指标函数中实现。

（二）问题分析与求解

问题中的群众留言分类问题，需要我们建立关于留言内容的一级标签分类模型，首先我们将所要用的全部文本数据导入到 jupyter 中。

Step 1 数据导入和呈现：对附件二中的群众留言文本数据进行清洗，判断数据是否有缺失值，发现留言编号一列和留言内容一列均无空值，因各种类型的评论内容分布很不均匀，用图形化的方式来查看不同类别留言的大致分布情况；再将留言类名转化为数字形式，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、卫生计生依次编号为0～6。

Step 2 数据清洗：由于文本留言数据都是汉字形式,所以要对文本进行一些预处理工作,包括删除文本中的标点符号、特殊符号以及一些无意义的常用词，因为这些词和符号对系统分类预测文本内容无作用,还会增加系统运算的复杂度，所以在使用这些文本数据之前必须要将它们清洗干净。

Step 3 留言分词及绘制词云：比较关键的一部就是在清洗完文本留言数据后，在此基础上进行留言内容的分词，把每条留言分成有空格隔开的独立词语或词语对，这样就得到了处理后逐条留言的分词字段；再利用这些分词字段生成每个分类的词云，我们罗列每个分类中前 100 个高频词并画出这些高频词的词云：

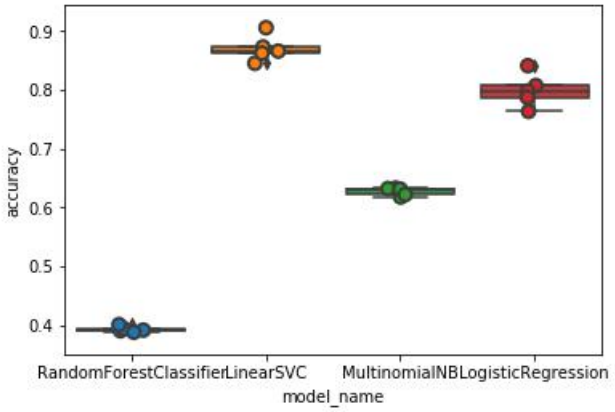
Step 4 **整理分词字段并计算特征数量**：接下来我们要计算这些留言分词字段的 TF-IDF 特征值（计算方法按照上述 TF、IDF 的计算公式），其作用可以降低高频词的权重，增加罕见词的权重。需要注意的是抽取文本留言时，除了要抽取单独的留言词语，还要抽取每个词语相邻的词组成词语对。（目的是扩展特征集的数量，提高分类文本的准确度）抽取结果显示文本分词字段的维度为：共有 9210 条留言数据，776013 个留言特征数（包括词语和词语对的总数）

Step 5 **卡方检验筛选留言关键词语（对）**：我们用卡方检验找出每个留言分类中关联度最大的两个词语和词语对用以反映群众留言的主题，比如在交通运输这个留言类别中，我们找到的关联度最大的两个词语是：快递、出租车，两个

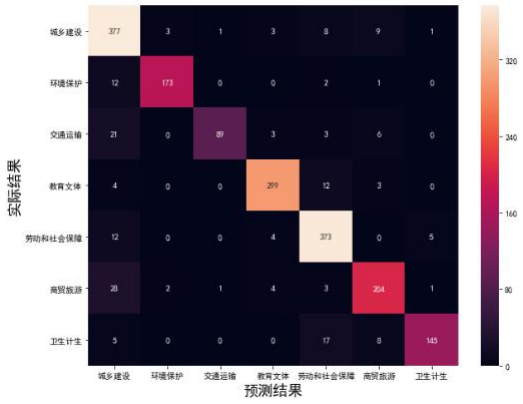
词语对是：的士 司机和出租车 司机；在卫生计生这个类别中词语和词语对分别是：医生、医院，社会 抚养费和乡村 医生等等。具体筛选结果如下图所示：

Step 6 贝叶斯转换及初步识别分类：而后我们利用朴素贝叶斯分类器转换成词频向量，再将词频向量转换成 TF-IDF 向量，最后在训练我们所建立的贝叶斯分类器，当分类器训练完成后，已经可以初步建立预测函数来进行留言内容的分类识别，如下图：

Step 7 其它机器学习模型评估：接下来我们尝试各种机器学习模型来完善分类识别并评估它们的准确率，主要使用以下四种模型：逻辑回归、多项式朴素贝叶斯、线性支持向量机、随机森林；通过箱体图（如下所示）来评估它们的准确率，可以看出线性支持向量机的平均准确率最高，随机森林的平均准确率是最低的。



Step 8 **建立混淆矩阵评估分类预测准确率：**最后我们就利用平均准确率最高的线性支持向量机模型来模拟预测标签和实际标签分类的差异，导入训练模型并生成混淆矩阵，此矩阵的主对角元素表示预测留言分类正确的记录数，而其余各行各列则表示对应标签错误分类的记录数，从下图所示的混淆矩阵可看出分类正确记录数最多即预测准确率最高的分类标签是‘交通运输’。



Step 9 **F-score 预测模型评价及修正：**当训练集数据不平衡（即文本类数据或多或少）时，计算出的准确率也不太能反映出模型的实际预测精度，所以借助 F-score 评价模型检验评估模型的预测精度，将留言文本数据的查准率和查全率代入公式 $F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$ 中，分别计算七个分类标签的 F_1 值，结果如下：

accuracy 0.9011943539630836

	precision	recall	f1-score	support
城乡建设	0.82	0.94	0.88	402
环境保护	0.97	0.92	0.95	188
交通运输	0.98	0.73	0.84	122
教育文体	0.96	0.94	0.95	318
劳动和社会保障	0.89	0.95	0.92	394
商贸旅游	0.88	0.84	0.86	243
卫生计生	0.95	0.83	0.89	175
micro avg	0.90	0.90	0.90	1842
macro avg	0.92	0.88	0.90	1842
weighted avg	0.91	0.90	0.90	1842

通过上表中的结果，可以发现 F_1 评价模型指标值在不同的文本记录数条件下是有较大差异的，比如根据混淆矩阵的结果，交通运输和劳动社会保障这两个分类标签的预测准确度是很高的，但它们的 F_1 值也是有较大差异的；且 F-score 模型本身近似于精确率与召回率的均值，很大程度上受单方面影响，所以评价结果也会存在误差：比如交通运输这一分类标签虽然精确率较高，但其宏平均召回率的值较低导致了评价结果出入较大，可以尝试改用微平均的方法来计算召回率来修正评价结果。

模型评价

本题目应用的模型较多，都是基于 python 语言下的机器学习分类模型，包括卡方检验、贝叶斯检验、混淆矩阵及 F-score 评价等方法，但这些分类模型又都有各自的弊端，碧如 F-score 评价在数据较少的情况下对召回率和准确率的运算是有较大差异的，这就导致最终的分类结果有误差，但因时间和条件的限制，应当继续挖掘更好的分类模型来解决这类问题。