

# 基于自然语言处理的智慧政务数据挖掘分析

摘要：随着大数据、云计算、人工智能等技术的蓬勃发展，网络问政平台也逐步成为政府倾听人民心声，了解民众所惑的重要渠道。各类社情民意相关的文本数据量庞大，如何对这些大量的数据进行细致深入的分析，从中发现并提取出这些潜在的有用的信息，高效的处理这些文本亟待解决。自然语言处理的高速发展，可以处理政务数据并进行挖掘分析，搭建智慧政务平台，提高办事效率。

对于本次赛题给出的数据，我们对群众留言问题用四种模型 Logistic Regression(逻辑回归)，(Multinomial) Naive Bayes(多项式朴素贝叶斯)，Linear Support Vector Machine(线性支持向量机)，Random Forest(随机森林)来进行处理最后用 F-Score 来打分评判模型，借助箱体图和混淆矩阵来筛选。同时我们对民众留言运用 TextRank 提取关键词实行热点追踪并用 Reddit 排名算法进行热度排名。最后我们用空间向量模型对政府答复意见进行相关性评价判断。

关键词：标签分类    F-Score 评分    Reddit 排名算法    空间向量模型

## **Abstract**

With the vigorous development of technologies such as big data, cloud computing, artificial intelligence, etc., the network questioning platform has gradually become an important channel for the government to listen to the hearts of the people and understand the confusion of the people. The amount of text data related to various social conditions and public opinion is huge. How to analyze these large amounts of data in detail and in-depth analysis, find and extract these potentially useful information, and efficiently process these texts need to be resolved.

For the data given in this question, we use four models Logistic Regression , Multinomial Naive Bayes, Linear Support Vector Machine, Random Forest to process and finally use F-Score to score the evaluation model, with the help of box plot and confusion matrix to filter At the same time, we use TextRank to extract keywords from people's messages, track hot spots, and use Reddit ranking algorithm to rank hotly. Finally, we use the space vector model to evaluate the relevance of government responses.

Keywords: label classification F-Score score Reddit ranking algorithm space vector model

# 目录

一. 挖掘目标.....	1
二. 数据准备.....	1
三. 任务一：构建一级标签分类模型.....	2
3.1 数据预处理及初步分析.....	2
3.2 分类器的选择.....	3
3.3 模型的选择.....	4
3.4 模型的评估.....	5
四. 任务二：热点追踪.....	7
4.1 总体的分析框架.....	7
4.2 文本数据的预处理.....	8
4.3 热度计算.....	10
4.4 热点问题提取和热度排名.....	12
五. 任务三：意见答复评价.....	12
5.1 答复评价.....	12
六. 总结.....	15
七. 参考文献.....	16

## 一. 挖掘目标

数据挖掘(Data Mining, 也称作知识发现)由于涉及的面比较广, 至今没有形成一个统一的定义。在本文中, 我们采用元昌安等人提出的定义<sup>[1]</sup>。数据挖掘就是从大量数据中获取有效的、新颖的、潜在有用的、最终可理解的模式的非平凡过程。简单地说, 数据挖掘就是从大量数据中挖掘想获取的知识。若是从挖掘任务角度来看, 可以将 数据挖掘分为关联规则、分类、聚类、回归、预测和异常检测等。在这当中数据分类是最活跃的课题, 得到了广泛和充分的研究。本次的第一个任务就是构建一个一级标签分类模型。

本次的挖掘目标: 首先, 通过对民众信访留言基于自然语言处理技术构建一个一级标签分类模型可以把群众的问题分派给相关的职能部门提高效率。其次, 群众留言进行一个热点追踪并进行热度排名及时发现热点问题, 有助于相关部门进行有针对性地处理, 提升服务效率。再者, 就是对政府给出的答复进行相关性、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。以便搭建一个智慧政务平台, 快速高效为人民办实事。

## 二. 数据准备

我们把题目给的相关 excel 文件转化成为 CSV 文件, 便于后期处理。CSV 文件逗号分隔值(Comma-Separated Values, 有时也称为字符分隔值, 因为分隔字符也可以不是逗号), 其文件以纯文本形式存储表格数据(数字和文本)。纯文本意味着该文件是一个字符序列, 不含必须像二进制数字那样被解读的数据。CSV 文件由任意数目的记录组成, 记录间以某种换行符分隔; 每条记录由字段组成, 字段间的分隔符是其它字符或字符串, 最常见的是逗号或制表符。通常, 所有记录都有完全相同的字段序列, 通常都是纯文本文件。

CSV 文件与 Excel 文件相比, CSV 文件的一个主要优点是有很多程序可以存储、转换和处理纯文本文件; 相比之下, 能处理 Excel 文件的应用程序却不多。使用 CSV 文件的另一个问题它只能保存数据, 不能保存公式, 但是通过数据存储(CSV 文件)和数据处理(Python 脚本)分离, 我们可以很容易地在不同数据集

上进行加工处理。当数据存储和数据处理过程分开进行时，错误（不管是数据处理中的错误，还是数据存储中的错误）不但更容易被发现，而且更难扩散。

在任务二中要进行热点追踪，热点追踪一般是在某个时间段的某个地方的相关问题，这就需要对地名进行提取，普通的关键词提取对一些地名提取结果不理想。这就需要用户自定义一个词典，以便准确提取地名和相关信息。

### 三. 任务一：构建一级标签分类模型

#### 3.1 数据预处理及初步分类

观察数据我们可以发现群众留言可以大致建立七个一级标签：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。我们将一级标签转换成了 Id(0 到 7)，我们需要对一级标签和留言主题进行一些预处理工作，这包括统计各个类别的数据量及类目分布。七个类目的分布如图图 3-1 所示：

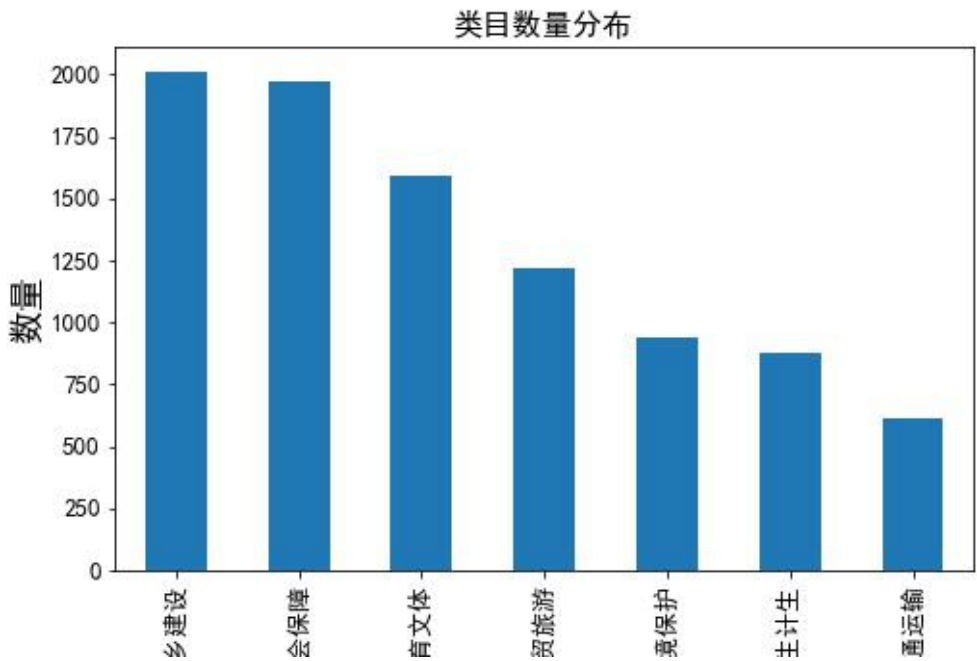


图 3-1 标签类目数量分布

由于我们的留言主题都是中文,对留言主题进行包括删除文本中的标点符号,特殊符号,还要删除一些无意义的常用词(stopword)比如:中文停用词包含了很多日常使用频率很高的常用词,如 吧,吗,呢,啥等一些感叹词等。这些高频常用词无法反应出文本的主要意思,所以要被过滤掉。因为这些词和符号对系统分析预测文本的内容没有任何帮助,反而会增加计算的复杂度和增加系统开销,所有在使用这些文本数据之前必须要将它们清理干净。文本分词的实现,文本分词是指使用计算机自动对文本进行语句的切分<sup>[2]</sup>。我们用 jieba 进行分词来进行词频统计绘制每个分类中罗列前 150 个高频词。然后我们要画出这些高频词的词云图。我们要计算留言主题中的 TF-IDF 的特征值,TF-IDF 是一种统计方法,用以评估一字词对于一个文件集或一个语料库中的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。

这里我们会使用 `sklearn.feature_extraction.text.TfidfVectorizer` 方法来抽取文本的 TF-IDF 的特征值。这里我们使用了参数 `gram_range=(1, 2)`,这表示我们除了抽取评论中的每个词语外,还要抽取每个词相邻的词并组成一个“词语对”,如:词 1,词 2,词 3,词 4,(词 1,词 2),(词 2,词 3),(词 3,词 4)。这样就扩展了我们特征集的数量,有了丰富的特征集才有可能提高我们分类文本的准确度。参数 `norm='l2'`,是一种数据标准化处理的方式,可以将数据限制在一定的范围内比如说  $(-1, 1)$  下面我们要是卡方检验的方法来找出每个分类中关联度最大的两个词语和两个词语对。卡方检验是一种统计学的工具,用来检验数据的拟合度和关联度。在这里我们使用 `sklearn` 中的 `chi2` 方法。我们可以看到经过卡方(`chi2`)检验后,找出了每个分类中关联度最强的两个词和两个词语对。这些词和词语对能很好的反映出分类的主题。

简而言之,我们要做的实质就是数据分类。数据分类(Classification)就是从现有的带标签的数据样本中根据事物间的相似性构造一个分类器(也称为分类函数或分类模式),进而用该分类函数对新来的 未标记样本的标签作出预测的过程。

## 3.2 分类器的选择

我们采用朴素贝叶斯分类器来进行分类,朴素贝叶斯分类器最适合用于基于

词频的高维数据分类器，最典型的应用如垃圾邮件分类器等，准确率可以高达95%以上。这里我们使用的是 sklearn 的朴素贝叶斯分类器 MultinomialNB，我们首先将留言主题转换成词频向量，然后将词频向量再转换成 TF-IDF 向量，还有一种简化的方式是直接使用 TfidfVectorizer 来生成 TF-IDF 向量(正如前面生成 features 的过程)，这里我们还是按照一般的方式将生成 TF-IDF 向量分成两个步骤：1. 生成词频向量。2. 生成 TF-IDF 向量。最后我们开始训练我们的 MultinomialNB 分类器如下图 3-2 所示：

```
myPredict('第二中学的老师数量不够')  
myPredict('医院的床位不够')
```

教育文体  
卫生计生

图 3-2 模型预测演示

可见模型预测效果还是很不错

### 3.3 模型的选择

我们尝试了四种模型 Logistic Regression(逻辑回归)，(Multinomial) Naive Bayes(多项式朴素贝叶斯)，Linear Support Vector Machine(线性支持向量机)，Random Forest(随机森林)并用箱体图进行准确率判断。如下图 3-3 所示：

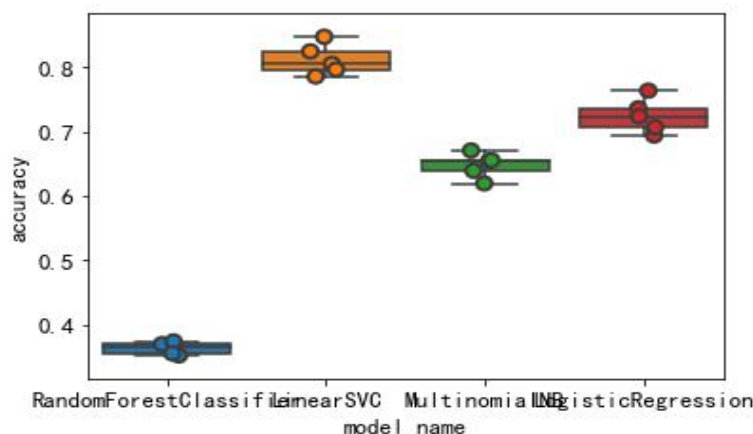


图 3-3 箱体图

可以看出随机森林分类器的准确率是最低的，因为随机森林属于集成分类器

(有若干个子分类器组合而成)，一般来说集成分类器不适合处理高维数据(如文本数据)，因为文本数据有太多的特征值，使得集成分类器难以应付，另外三个分类器的平均准确率都在 80%以上。其中线性支持向量机的准确率最高。通过计算我们发现线性支持向量机的平均准确率达到了 86.7%，其次是逻辑回归和朴素贝叶斯。

### 3.4 模型的评估

下面我们就针对平均准确率最高的 LinearSVC 模型，我们将查看混淆矩阵，并显示预测标签和实际标签之间的差异。如图 3-4 所示：

混淆矩阵的主对角线表示预测正确的数量，除主对角线外其余都是预测错误的数量。可以看出预测结果还不错，正确率比较高。

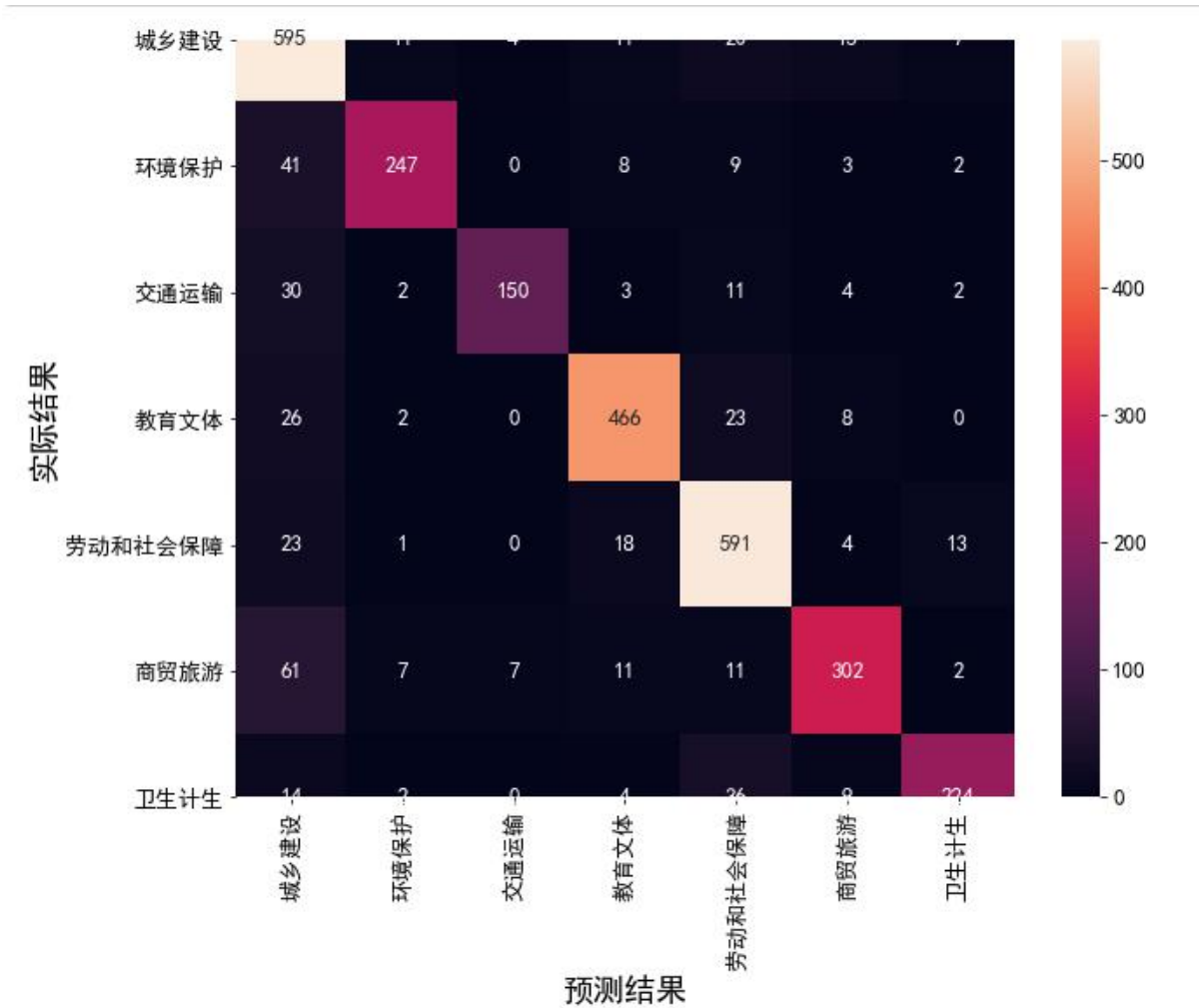


图 3-4 混淆矩阵



同时我们也借助于 F-Score 分数等指标来对模型进行评估,如 3-1 公式所示,其中  $P_i$  为第  $i$  类的查准率,  $R_i$  为第  $i$  类的查全率。

$$F_1 = \frac{1}{n} \sum_{i=1}^n \frac{2 P_i R_i}{P_i + R_i} \quad (3-1)$$

观察图 3-5 可以知道整个一级标签平均得分 85%,"教育文本"类的 F1 分数最高达到 89%,"商贸旅游"类 F1 分数最差只有 81%,分数低的原因可能是因为各个分类的训练数据比较少,使得模型学习的不够充分,导致预测失误较多。

accuracy 0.8470394736842105				
	precision	recall	f1-score	support
城乡建设	0.75	0.90	0.82	663
环境保护	0.91	0.80	0.85	310
交通运输	0.93	0.74	0.83	202
教育文体	0.89	0.89	0.89	525
劳动和社会保障	0.84	0.91	0.87	650
商贸旅游	0.88	0.75	0.81	401
卫生计生	0.90	0.78	0.83	289
accuracy			0.85	3040
macro avg	0.87	0.82	0.84	3040
weighted avg	0.85	0.85	0.85	3040

图 3-5 F-Score 评估

下面我们来查看一些预测失误的相关数据:

- 环境保护 预测为 城乡建设 : 41 例.
- 环境保护 预测为 教育文体 : 8 例.
- 环境保护 预测为 劳动和社会保障 : 9 例
- 交通运输 预测为 劳动和社会保障 : 11 例.
- 交通运输 预测为 城乡建设 : 30 例.
- 教育文体 预测为 劳动和社会保障 : 23 例.
- 教育文体 预测为 商贸旅游 : 8 例.
- 教育文体 预测为 城乡建设 : 26 例.
- 劳动和社会保障 预测为 教育文体 : 18 例.
- 劳动和社会保障 预测为 卫生计生 : 13 例.
- 劳动和社会保障 预测为 城乡建设 : 23 例.

商贸旅游 预测为 环境保护 : 7 例.  
商贸旅游 预测为 交通运输 : 7 例.  
商贸旅游 预测为 劳动和社会保障 : 11 例.  
商贸旅游 预测为 教育文体 : 11 例  
商贸旅游 预测为 城乡建设 : 61 例  
卫生计生 预测为 劳动和社会保障 : 36 例.  
卫生计生 预测为 商贸旅游 : 9 例.  
卫生计生 预测为 城乡建设 : 14 例.  
城乡建设 预测为 环境保护 : 11 例.  
城乡建设 预测为 教育文体 : 11 例.  
城乡建设 预测为 劳动和社会保障 : 20 例.  
城乡建设 预测为 商贸旅游 : 15 例.  
城乡建设 预测为 卫生计生 : 7 例.

通过对这些预测失误的例子,我们发现有些民众的留言主题里面涉及多标签问题,模型不能准确的把它归为其中一类,可以加大各个分类的训练数据来改善我们的分类器,从而提高预测精度。

## 四. 任务二: 热点追踪

在互联网高速发展的时代,人们获取信息的方式也相对以前有了变化。互联网也成了获取信息的又一途径,越来越多的人在网上表达自己所遇到的问题。所以及时发现群众所留言的热点问题,会更有助于有关部门进行有针对性地处理问题,从而提升服务效率。

现如今,网友们通过互联网来表达自己的对热点问题的看法和意见,那么及时处理和分析这些看法和意见就可以更好地反馈给相关部门,并给予回应。但是,现如今的网络过于发达,传统的人工式分析已经无法应对爆炸式的信息增长,需要采用数据挖掘、大数据等先进技术手段。此次任务,通过分析从网络问政平台得到的数据,定义了合理的热度评价指标,并给出了评价结果。

### 4.1 总体的分析框架

基于文本挖掘应用中的“智慧政务”在处理时主要包括信息获取、信息预处理、热度计算以及热点问题匹配。

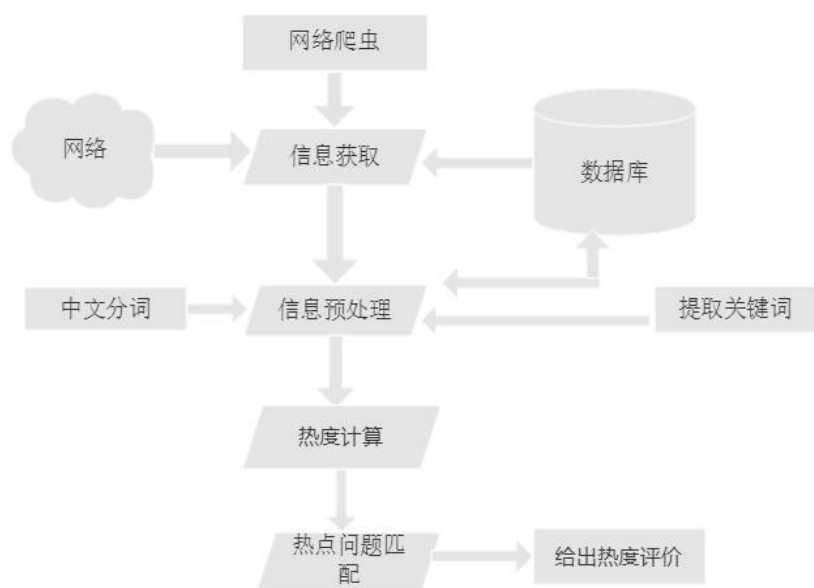


图 4-1 基于文本挖掘应用的热点问题挖掘的总体框架

## 4.2 文本数据的预处理

将文本中的数据进行清洗，除去空值、进行分词、词频统计、提取关键词等处理。在文本信息预处理之前，要先先读取中文文本，然后除去中文文本中的非文本部分例如 HTML 的标签，少量的非文本内容可以用 Python 中的正则表达式（re）删除，去除这些非文本内容过后才能进行文本的预处理。用 Python 处理中文文本时，需要处理中文文本的编码问题，存储数据都使用 utf8；读出的数据进行处理时，使用 GBK 之类的中文编码。处理好编码问题过后需要进行除去停用词等操作。

文本分词的实现，我们采用 jieba 分词，jieba 分词有三种分词模式：精确模式、全模式、搜索引擎模式，在这任务二中采用精确模式。在使用 jieba 分词的时候需要进行读取用户自定义词典然后用 jieba.lcut() 生成一个列表，然后导入停用词，并除去所生成列表文件中的停用词。jieba 分词过后进行词频统计绘制词云图如图 4-2 所示。



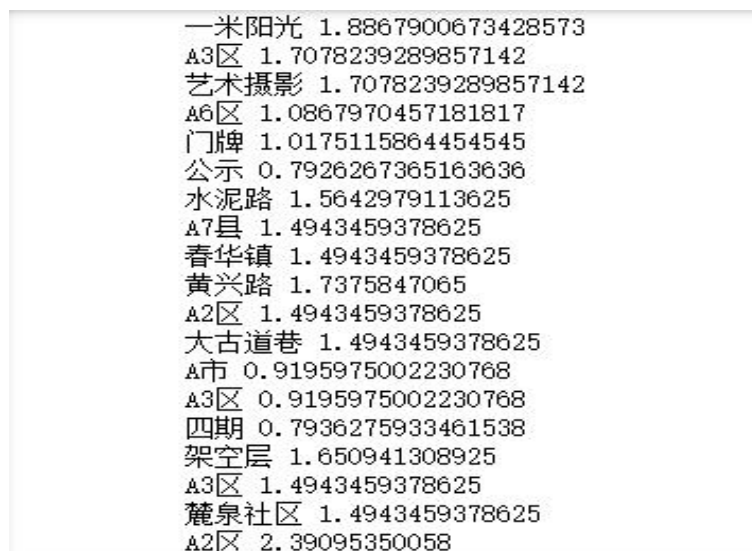
(3) 构建候选关键词图  $G = (V, E)$ ，其中  $V$  为节点集，由 (2) 生成的候选关键词组成，然后采用共现关系 (Co-Occurrence) 构造任两点之间的边，两个节点之间存在边仅当它们对应的词汇在长度为  $K$  的窗口中共现， $K$  表示窗口大小，即最多共现  $K$  个单词。

(4) 根据 TextRank 的公式，迭代传播各节点的权重，直至收敛。

(5) 对节点权重进行倒序排序，从而得到最重要的  $T$  个单词，作为候选关键词。

(6) 由 (5) 得到最重要的  $T$  个单词，在原始文本中进行标记，若形成相邻词组，则组合成多词关键词。

下图 4-3 为 Python 调用 jieba.analyse.textrank 提取出的关键词示例：



```
一米阳光 1.8867900673428573
A3区 1.7078239289857142
艺术摄影 1.7078239289857142
A6区 1.0867970457181817
门牌 1.0175115864454545
公示 0.7926267365163636
水泥路 1.5642979113625
A7县 1.4943459378625
春华镇 1.4943459378625
黄兴路 1.7375847065
A2区 1.4943459378625
大古道巷 1.4943459378625
A市 0.9195975002230768
A3区 0.9195975002230768
四期 0.7936275933461538
架空层 1.650941308925
A3区 1.4943459378625
麓泉社区 1.4943459378625
A2区 2.39095350058
```

图 4-3 关键词提取示例

## 4.3 热度计算

此处的热度计算方法采用的 Reddit 排名算法的模型，Reddit 算法可以对既有点赞数，又有反对数的话题进行热度处理计算。算法的数学描述如下图 4-4：

Given the time the entry was posted  $A$  and the time of 7:46:43 a.m. December 8, 2005  $B$ , we have  $t_s$  as their difference in seconds

$$t_s = A - B$$

and  $x$  as the difference between the number of up votes  $U$  and the number of down votes  $D$

$$x = U - D$$

where  $y \in \{-1, 0, 1\}$

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

and  $z$  as the maximal value, of the absolute value of  $x$  and 1

$$z = \begin{cases} |x| & \text{if } |x| \geq 1 \\ 1 & \text{if } |x| < 1 \end{cases}$$

we have the rating as a function  $f(t_s, y, z)$

$$f(t_s, y, z) = \log_{10} z + \frac{y t_s}{45000}$$

图 4-4 Reddit 算法的数学表达式

问题的发表时间  $t$ =发表时间-2020 年 5 月 6 日 18:19:00，可以看出该算法的时间单位是秒。发表时间对热点问题排名的影响很大，新问题的排名会相对于旧的话题排名较高，但是话题的得分不会因为时间的流逝减少。点赞数与反对数的差  $x$ =点赞数-反对数， $x$  可以让一些具有争议的话题的排名靠后。

投票方向  $y$  是一个符号变量，表示对该话题的总体看法，如果点赞数居多， $y$  就是 +1；如果反对数居多， $y$  就是-1；如果点赞数和反对数相等， $y$  就是 0。 $y$  是文章评价的一种定性表达，0 表示没有倾向，大于 0 表示正面评价，小于 0 表示负面评价。话题的受肯定程度  $z$ ，表示点赞数超过反对数的数量，当点赞数少于反对数时  $z=1$ 。

综合以上 Reddit 的最终得分计算公式表示为： $\text{Score} = \log_{10} z + \frac{y t}{45000}$  可以

看出得分受两个部分的影响。可以看出  $\log z$  对于得分的影响不是太大，所以说

此处为了加强点赞数的作用就需要将得分的计算公式改为： $Score = z + \frac{y^t}{45000}$  将该算法写入 Python 中进行热度计算。

### 4.4 热点问题提取和热度排名

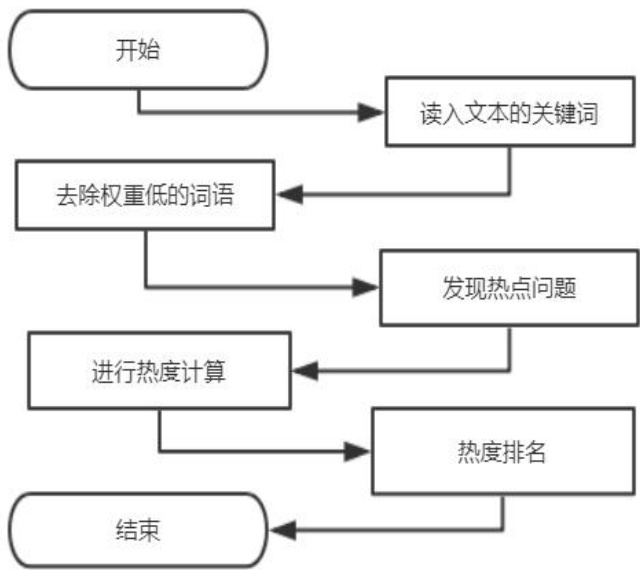


图 4-5 热点问题提取总体框架

由图 4-5 可知，对热点问题热度排名需要对文本中的关键词进行处理，要去除语句中权重较低的词语，进行热点问题统计，将相同问题的热度进行相加然后就可以对热度进行排名了。热度排名后需要将相同地点或相同人群的问题进行归类，并用 Python 将所有的文本数据读入表格当中。

## 五：任务三：意见答复评价

### 5.1 答复评价

首先，我们来看一下答复相关性。我们可以把群众留言详情看做一个文本，答复意见看做另一个文本。文本间关系的计算是对文本之间的相关性进行度量，考查两个文本之间的关联程度的计算过程。它涉及到计算机、人工智能以及认知科学等多个研究领域。作为文本处理中的一项重要内容，文本间关系的计算长期以来一直在自然语言处理的各类应用以及与之相关的交叉领域中被广泛的使用，

它是其它许多更加深入的文本处理的前提，对于帮助人们更加高效、准确地掌握蕴藏于文本中的各类信息具有非常重要的现实意义。现有文本的相关性计算模式如图 5-1 所示。两个文本之间的(内容)相关程度(Degree of Relevance)常常用他们之间的相似度 Sim 来度量。

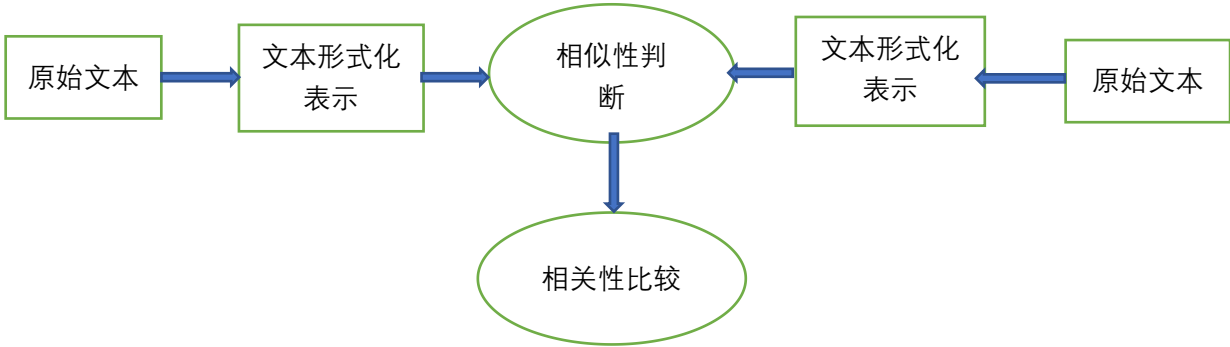


图 5-1 文本相关性计算模式

对留言群众的意见答复相关性，我们采用向量空间模型来进行评判。向量空间模型概念简单，把对文本内容的处理简化为向量空间中的向量运算，并且它以空间上的相似度表达语义的相似度，直观易懂。当文档被表示为文档空间的向量，就可以通过计算向量之间的相似性来度量文档间的相似性。文本处理中最常用的相似性度量方式是余弦距离。向量空间模型（Vector Space Model, VSM）自提出以来<sup>[4]</sup>，由于其所拥有的简洁、高效等优点，被广泛地应用于信息过滤、采集以及索引当中<sup>[5-9]</sup>，是一种优秀的评估相似性的代数模型。在向量空间模型中，进行相似性比较的双方被表示为多次元向量空间，模型假设，两者间的相似程度，可以经由比较每个向量间的夹角偏差程度而得知。在由 Salton, Wong and Yang 提出的古典的向量空间模型中，一个词在文档向量中的权重，为区域和全域参数的乘积。即所谓的 TF-IDF（词频-逆向文件频率）。对于权重向量为

$\mathbf{V}_d = \{w_{1,d}, w_{2,d}, \dots, w_{n,d}\}$  的文档  $d$ ，其项权重可以通过公式（5-1）所示方法得到：

$$w_{t,d} = tf_{t,d} \cdot \log \frac{|D|}{|\{t \in d\}|} \quad (5-1)$$



其中  $tf_t$  是词  $t$  在文档  $d$  中出现的次数； $\log \frac{|d|}{|\{t \in d\}|}$  是逆文献频率， $|D|$  是文档总数， $|\{t \in d\}|$  是含有词  $t$  的文档数。

在向量空间模型中，对于两个文本向量  $D_i = \{d_{1,i}, d_{2,i}, \dots, d_{n,i}\}$  和  $D_j = \{d_{1,j}, d_{2,j}, \dots, d_{n,j}\}$  ( $n \leq m$ )，比较每个向量间的夹角偏差程度的匹配函数有多种形式，主要包括：余弦相似度计算如式 5-2：

计算两个向量间夹角余弦，来源于点积运算的规范化。

$$\text{sim}(D_i, D_j) = \frac{\sum_{k=1}^n d_{k,i} \cdot d_{k,j}}{\sqrt{\sum_{k=1}^n d_{k,i}^2} \cdot \sqrt{\sum_{k=1}^m d_{k,j}^2}} \quad (5-2)$$

距离相似度计算如式 5-3：

$$\text{sim}(D_i, D_j) = L_p(D_i, D_j) = \left[ \sum |d_{k,i} - d_{k,j}|^p \right]^{\frac{1}{p}} \quad (5-3)$$

参数  $p$  决定了选择哪一种距离进行计算。如，当  $p=2$  时，即为欧式距离；当  $p \rightarrow \infty$  时，则为最大方向距离。项匹配个数相似度计算：以向量中项匹配的个数作为相似度计算的依据，具体计算方法有 Dice 系数法，如公式 (5-4) 所示，

$$\text{sim}(D_i, D_j) = \frac{2 \sum d_{k,i} \cdot d_{k,j}}{\sum d_{k,i}^2 + \sum d_{k,j}^2} \quad (5-4)$$

以及 Jaccard 系数法，如公式 (5-5) 所示，

$$\text{sim}(D_i, D_j) = \frac{\sum d_{k,i} \cdot d_{k,j}}{\sum d_{k,i}^2 + \sum d_{k,j}^2 - \sum d_{k,i} \cdot d_{k,j}} \quad (5-5)$$

向量空间模型具有直观、简洁、高效等优点，但是同时它也存在许多缺点，

人们对该模型的批评集中在如下几个方面：缺乏理论基础；向量空间中各词彼此独立。而且，在自然语言处理的应用中，传统的相量空间模型的缺点还包括：词的顺序被忽略；一词多义和一义多词问题被忽略。我们也需要考虑答复意见的完整性和可解释性。因为如果只是相关但对于群众而言却没有多大的实际用处就没有什么意义，所以这也是我们需要考虑的一个点。

## 六. 总结

本文通过对四种分类器进行比较得出 Linear Support Vector Machine(线性支持向量机)对群众留言建立的一级标签分类模型可以比较精准的进行预测，针对一些预测失误的例子我们也把它拿来分析，希望找到其中原因进而提高模型精度。它存在一个失误反馈，可以及时发现问题错在哪儿。针对群众的留言问题我们运用 Reddit 排名算法进行热度排名追踪热点。及时发现群众所留言的热点问题，会更有助于有关部门进行有针对性地处理问题，从而提升服务效率。群众留言后，我们也需要对政府给出的答复意见进行相关性，完整性以及可解释性进行评价。针对相关性，我们选用常规的空间向量模型来判断，模型在自然语言处理的应用中，传统的相量空间模型存在一些缺点，这也是我们需要努力改进的地方。

## 七. 参考文献

- [1] 元昌安等 数据挖掘原理与应用宝典. 北京: 电子工业出版社, 2009.
- [2] 吴刚勇, 张千斌, 吴恒超, 顾冰. 基于自然语言处理技术的电力客户投诉工单文本挖掘分析[J]. 电力大数据, 2018, 21(10): 68-73.
- [3] 吴柳, 程恺, 胡琪. 基于文本挖掘的论坛热点问题时变分析[J]. 软件, 2017, 38(04): 47-51
- [4]. G. Salton, C. S. Yang. A Vector Space Model for Automatic Indexing. Communications of the ACM. 1975, 18: 613-620
- [5]. J. Mostafa, M. Palakal, W. Lam. A Multilevel Approach to Intelligent Information Filtering: Model, System, and Evaluation. ACM Transactions on Information Systems (TOIS). 1997, 15(4): 368-399
- [6]. J. Wu, S. Wang, D. Pan, K. Yamamoto, Z. Wang. An Improved VSM Based Information Retrieval System and Fuzzy Query Expansion. Lecture Notes in Computer Science. 2005, 3613: 537-546
- [7]. Z. Chao, G. Juzhong. A New Approach to Email Classification Using Concept Vector Space Model. Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking Symposia. 2008, 3: 162-166
- [8]. N. Liu, J. Yan, Q. Yang, S. Yan, Z. Chen, F. Bai, W. Ma. Learning Similarity Measures in Non-orthogonal Space. Proceedings of the thirteenth ACM international conference on Information and knowledge management. 2004: 334 - 341
- [9]. W. Mao. The Phrase-based Vector Space Model for Automatic Retrieval of Free-text Medical Documents. Data & Knowledge Engineering. 2007, 61(1): 76-92