

# 基于“智慧政务”中的文本挖掘应用

## 摘要

近年来，随着互联网技术的提升和各大网络平台的迅速发展，各大网络问政平台逐步成为政府了解民意、改进工作的重要渠道，普通民众借助互联网以留言的方式发表自己的意见和想法，相关部门针对相应的问题做出回答。通过文本分类帮助政府快速、准确的获取有用信息，具有重要的研究意义与社会价值。本文将基于留言详情及答复意见的文本数据，运用统计、机器学习、自然语言处理与文本挖掘的相关知识，借助 Python 软件对文本进行处理。

对于问题一：首先将留言详情以及对应的一级分类按照 8: 2 的比例划分为训练集与测试集，对训练集与测试集下面的每个分类里面的每条留言进行处理分词处理，借助 python 里面的 jieba 库进行分词，去除停用词。其次将已经去除停用词之后的文本数据结构化表示后构建词向量空间，以 sklearn 库里面的 bunch 函数形式存储，再用 pickle.dump 保存为二进制文件，构建 TF-IDF 词向量。最后应用朴素贝叶斯分类器对留言进行分类，并利用 F-score 对分类方法进行评价。

对于问题二：首先利用 Single-pass 聚类算法进行文本处理，从而建立新的主题，将新的主题文档与已有主题进行相似度比较，用条件判断语句对相似度的值与阈值  $\delta$  进行比较，并通过实验取最优阈值  $\delta$  为 0.045，以此判别出聚类种子，建立新的主题类别。

对于问题三：为方便后面的处理首先将留言详情与答复意见分别保存为 txt 文档，所有留言详情用 python 读取并以列表形式存储，并对其进行分词并去除停用词。其次用词袋模型（Bag-of-Words）来提取文本特征，将构建好的词袋语料库，用 LSI 模型构建一个 LSI 向量空间，接着将答复意见全部读取出来存在列表中，分词并去除停用词，构建词袋模型并转为 LSI 向量空间并对其中的每个文档建立索引。最后依次进行全匹配查询，利用余弦相似性来度量两个向量之间的相似性，并取出对应的测试结果存到 Excel 中。

**关键词：**朴素贝叶斯，词袋模型，Single-pass 聚类，LSI 模型，余弦相似性

## **Abstract**

In recent years, with the improvement of Internet technology and the rapid development of major network platforms, major online political platforms have gradually become an important channel for the government to understand public opinion and improve its work. Ordinary people express their opinions and ideas through comments on the Internet, and relevant departments respond to relevant questions. It is of great significance and social value to help the government obtain useful information quickly and accurately through text classification. Based on the text data of comments and replies, this paper USES the knowledge of statistics, machine learning, natural language processing and text mining to process the text with the help of Python software.

For question 1: firstly, the message details and corresponding first-level classification are divided into training set and test set according to the ratio of 8:2. Each message in each category under the training set and test set is processed into word segmentation, and word segmentation is carried out with the help of jieba library in python to remove stop words. Secondly, after the structural representation of the text data after the stop words have been removed, the word vector space is constructed and stored in the form of bunch function in the sklearn library. Then pickle.dump is saved as binary file to construct the tf-idf word vector. Finally, the naive bayes classifier is used to classify the message and f-score is used to evaluate the classification method.

For question 2: first, Single - pass clustering algorithm is used for text processing, so as to establish a new theme, will the new subject document similarity comparison with existing theme, with the conditional statements to the similarity value compared with the threshold the delta, and through the experiment in the delta is 0.045, the optimal threshold to distinguish the clustering seeds, establish new subject categories.

For question 3: to facilitate the following processing, the message details and the reply comments are saved as TXT documents respectively. All message details are

read in python and stored in a list, and the word segmentation is carried out and the stop word is removed. Secondly, the bag-of-words model is used to extract text features. The constructed word Bag corpus is constructed, an LSI vector space is constructed with LSI model, and then all replies are read out and stored in the list. Word segmentation and stop Words are removed. Finally, the full-match query was conducted in turn, the cosine similarity was used to measure the similarity between the two vectors, and the corresponding test results were saved into Excel.

**Keywords:** Naive Bayes, Bag-of-Words, Single-pass clustering, LSI model, cosine similarity

# 目录

<b>1. 问题描述 .....</b>	<b>8</b>
1.1 挖掘背景 .....	8
1.2 挖掘目标 .....	8
<b>2. 问题分析 .....</b>	<b>9</b>
2.1 问题 1 的分析 .....	9
2.2 问题 2 的分析 .....	9
2.3 问题 3 的分析 .....	9
<b>3. 分析方法与过程 .....</b>	<b>10</b>
3.1 问题 1 分析方法与过程 .....	10
3.1.1 划分数据集 .....	11
3.1.2 文本预处理 .....	11
3.1.3 结构化表示 .....	12
3.1.4 特征权重 .....	13
3.1.5 文本分类 .....	14
3.2 问题二的分析方法与过程 .....	15
3.2.1 文本预处理 .....	17
3.2.2 基于词袋模型向量化 .....	18
3.2.3 相似度计算 .....	19
3.2.4 找出最大相似度的已有主题 .....	19
3.2.5 输出结果 .....	19
3.2.6 热度评价指标 .....	20
3.3 问题三的分析方法与过程 .....	21
3.3.1 文本预处理 .....	21
3.3.2 LSI 向量空间的构建 .....	21
3.3.3 相似度计算及评价定义 .....	22
3.3.4 评价结果 .....	22

4.	总结.....	24
5.	模型评价 .....	25
5.1	模型的优点 .....	25
5.2	模型的缺点 .....	26
5.3	模型的改进 .....	26
6.	参考文献 .....	27

# 表录

表 1 Bunch 数据结构存储表.....	12
表 2 权重矩阵样例表.....	14
表 3 分类模型评价结果.....	15
表 4 热点问题留言明细表样表 .....	19
表 5 热点问题表 .....	21
表 6 相关性大于 85%的群众留言编号 .....	22
表 7 相关性小于 10%的部分群众留言编号.....	23

# 图录

图 1 问题 1 算法流程图.....	10
图 2 训练集与测试集的存储形式 .....	11
图 3 每一类中留言存储形式 .....	11
图 4 结巴分词结果 .....	12
图 5 去除停用词后的结果.....	12
图 6 问题 2 算法流程图.....	17
图 7 分词并去停用词后的内容 .....	17
图 8 获取分词后词汇与词汇 id 映射关系.....	18
图 9 得到语句的向量表示.....	18
图 10 获取语句的 TF-IDF 向量表示 .....	18
图 11 LSI 向量空间.....	22
图 12 相关部门对群众留言的相关性图.....	24
图 13 留言编号为 50409 的留言内容与答复意见 .....	24
图 14 留言标号为 30019 的留言内容与答复意见 .....	24

# 1. 问题描述

## 1.1 挖掘背景

随着社会的发展，互联网技术的提升和各大网络平台的迅速发展，信息的发布渠道、来源也多样化，人们通过网络问政平台针对社会上出现的一系列问题发表自己的意见和想法，相关部门根据留言内容进行回复，这种方式对提升政府的执政能力和水平具有很大的作用。

大数据时代的到来导致信息量迅速增长，一般传统的数据处理方法已经无法很好的解决信息泛滥的问题<sup>[1]</sup>，将大量的文本信息进行整理、分类给相关部门以便于及时回复带来了极大的挑战，如何从群众留言信息中挖掘出关键信息并进行留言分类，对热点问题进行准确、有效的整理以及判断答复意见的实时性、准确性、高效性至关重要，为解决海量文本数据处理的问题，数据挖掘技术应运而生<sup>[2]</sup>。文本挖掘最早由费尔德曼和达冈提出，它是一种文本分析方法，通常与统计和机器学习结合在一起<sup>[3,4]</sup>。文本挖掘的关键步骤主要包括文本预处理、文本分类和聚类、以及结果的可视化等<sup>[5,6]</sup>。根据提供的文本数据运用文本挖掘的方法，建立一套评价答复意见的质量的方案很有必要，对文本处理具有极大的经济和社会价值。

## 1.2 挖掘目标

本次建模是将留言详情以及对应的一级分类中的文本数据，利用 jieba 对留言进行分词，再通过 F-score 方法进行评价、Single-pass 聚类算法、用 LSI 模型构造向量空间，达到以下目标：

（1）利用所给数据（附件 2），即已经根据一级标签分好类的留言内容，划分训练集与测试集，建立关于留言内容的一级分类标签分类模型。

（2）利用所给数据（附件 3），对某一时间段内反映特定地点或特定人群问题的留言进行分类，定义合理热度评价指标，给出评价结果，并挖掘排名前 5 的热点问题。

（3）利用所给数据（附件 4），由相关部门对留言的答复意见，从答复意见



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。

## 2. 问题分析

### 2.1 问题 1 的分析

本题要求从所给数据（附件 2）中，建立一个关于留言内容的一级标签分类模型。这是一个典型的文本分类问题，文本分类过程一般包括文本预处理、特征选择、特征权重计算、训练和分类。

原始数据为 Excel 表格，为了方便计算，将留言详情根据所对应的一级标签存储为 txt 文件。然后按照文本分类的过程利用 python 来实现中文文本的分类。

### 2.2 问题 2 的分析

问题二要求利用所给数据（附件 3），从众多留言内容中识别出相似留言，并把特定地点或人群的数据归为一类，即把相似的留言归为同一问题。然后定义热度评价指标的计算方法，给出评价结果。

该题涉及到文本相似度计算与文本聚类问题，难点在于地点、人群的表达多样化，以及特征多、两两之间计算相似计算量大。因此需要一个效果较为良好的聚类方法。

### 2.3 问题 3 的分析

问题三的数据给出了群众留言与相关部门的答复意见，要求从答复的相关性、完整性、可解释性等角度对答复意见的评价。

该题难点在于如何将相关性、完整性和可解释性等描述量化，还有构建什么指标来计算和评价。

### 3. 分析方法与过程

#### 3.1 问题 1 分析方法与过程

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，且差错率高等问题。为了能将群众的留言内容进行分类，建立一个关于留言内容的一级标签分类模型。其算法流程图如下：

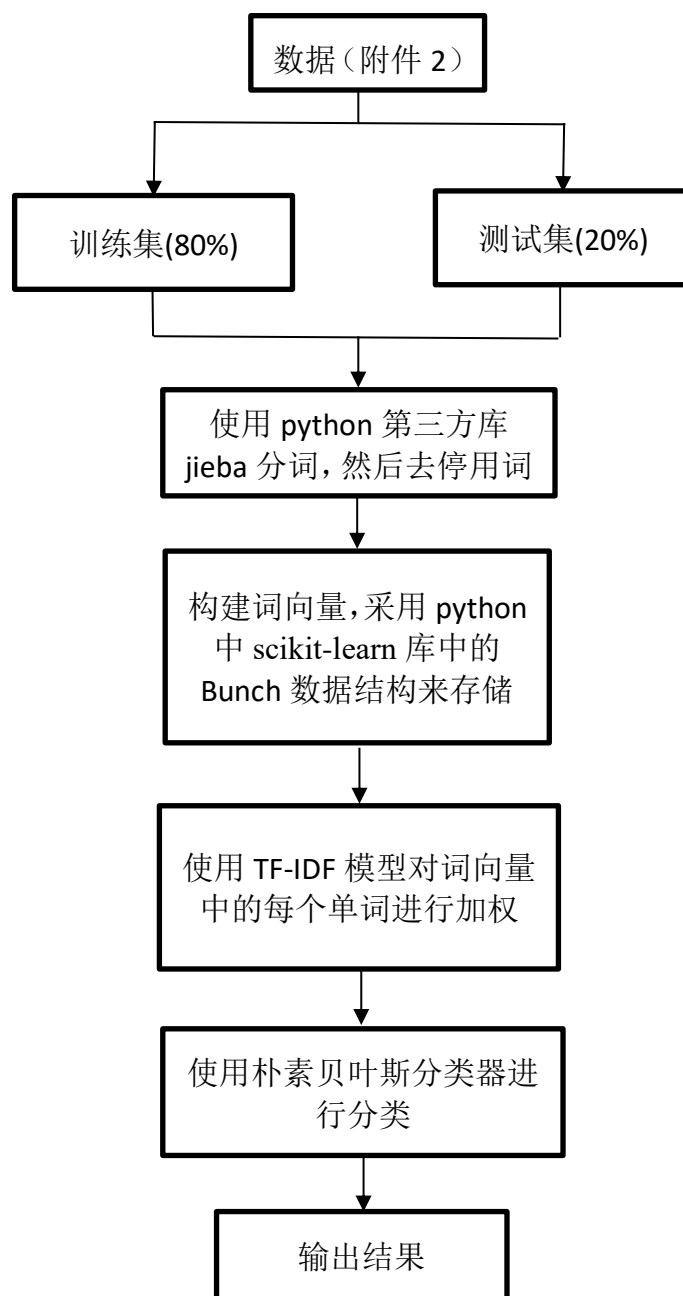


图 1 问题 1 算法流程图

### 3.1.1 划分数据集

原始文件（附件 2）是以 Excel 表格形式呈现的，并且已经将一级标签注释好。现将数据以 8:2 的比例拆分为训练集与测试集，为了便于计算，将训练集与测试集的留言内容分别按照对应的一级标签储存为 txt 文件。然后再按照中文文本分类的步骤，进行分类实现。

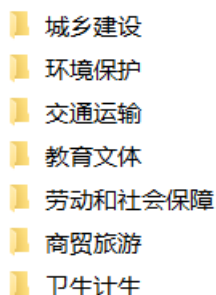


图 2 训练集与测试集的存储形式

如图 2 所示，总的有 7 类，每类中有群众留言内容见图 3。

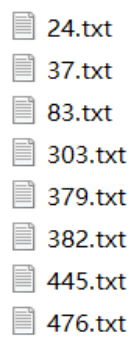


图 3 每一类中留言存储形式

留言内容的命名为每个留言的留言编号，如图 3 是训练集中一级标签为城乡建设的小部分，表示留言编号为 24、37、83、303、379、382、445、476 的留言内容。

### 3.1.2 文本预处理

在进行文本分类之前，先要对文本进行预处理，即利用 python 对文本进行分词、去除停用词。文本分词在这里用到 python 的第三方库 jieba 库。jieba 分词依靠中文词库，利用一个中文词库，确定汉字之间的关联概率，汉字之间概率大的组成词组，形成分词结果。除了分词，使用者还可以添加自定义的词组。jiaba.cut()方法可以实现文本的分词，这个方法有两个参数，第一个参数是需要分

词的文本数据，第二个参数是分词模式的选择。分词模式有三种：

- 1) 全模式(`jieba.cut(text,cut_all=True)`);
- 2) 精准模式(`jieba.cut(text,cut_all=False)`)，默认设置也是精准模式;
- 3) 搜索引擎模式(`jieba.cut_for_search(text)`)

本题使用精准模式，适合进行文本分析。这样才能基于单词的基础上，对文档进行结构化表示。图 4 显示了 `jieba.cut()`方法采用精准模式将训练集中一级标签为城乡建设，留言编号为 24 的留言内容进行分词后的结果。

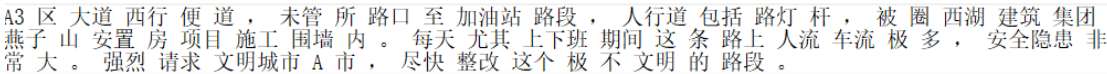


图 4 结巴分词结果

分词结束后，进行去除停用词操作。去除停用词即在分词后的文件集中去掉不需要的词汇、字符，这样能达到一个降维的效果。图 5 图 5 显示了训练集中一级标签为城乡建设，留言编号为 24 的留言内容进行分词后，再去掉停用词之后的结果。



图 5 去除停用词后的结果

### 3.1.3 结构化表示

将分词并去停用词之后的训练集和测试集表示为向量，就是将分好的词统一到一个词向量中，形成一个  $n$  维的词向量空间。这里采用 `python` 里面的 `scikit-learn` 库中的 `Bunch` 数据结构来表示这两个数据集。

`Bunch` 的结构和 `python` 中的字典的结构类似，和字典的区别在于其键值可以被实例对象当作属性使用。`Bunch` 对象里面有四个参数，这些参数就相当于键，存放的数据就相当于值，存储结构都为列表结构：

- 1) `target_name`: 存放的是整个数据集的类别集合。
- 2) `label`: 存放的是所有文本的标签。
- 3) `filenames`: 存放的是所有文本文件的名称。
- 4) `contents`: 分词后的文本文件。

表 1 `Bunch` 数据结构存储表

<code>target_name</code>	<code>label</code>	<code>filenames</code>	<code>contents</code>
--------------------------	--------------------	------------------------	-----------------------

	城乡建设	24.txt	……人行道 包括 路……
城乡建设	城乡建设	37.txt	……在水一方 大厦 ……
	城乡建设	83.txt	……火炬 小区 物业……
	环境保护	4445.txt	……大车 噪音 扰民……
环境保护	环境保护	5017.txt	……住工 每到 凌晨……
	环境保护	5875.txt	……郑家 冲组 村民……
	交通运输	3757.txt	……时代 倾城 小区……
交通运输	交通运输	3769.txt	……交通 知识 宣传……
	交通运输	3772.txt	……来往 车辆 车速……

如表 1 所看到的，是将训练集里面的城乡建设、环境保护和交通运输这三个标签里面的三个留言内容以 Bunch 数据结构存储的结果，在这里以表格的形式展现出来。其中 `target_name` 存储的就是一级分类，`label` 存储的是每个留言对应的标签，`filenames` 存储的是文档的名称，`contents` 则是存放的每条留言进行分词并去停用词之后的内容。

### 3.1.4 特征权重

在已经处理成词向量空间的两个数据集中，需要赋予权值将文档表示为加权的特征向量。在文本处理领域中，使用最广泛的权重计算方法是 TF-IDF (Term Frequency\*Inverse Document Frequency) 权重方法。TF 是词频 (Term Frequency)，IDF 是逆文本频率指数 (Inverse Document Frequency)。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。

词频可表示为：

$$tf_{i,j} = \frac{n_{k,j}}{\sum_k n_{k,j}} \quad (1)$$

(1) 式中分子是该词在文件中的出现次数，而分母则是在文件中所有字词的出现次数之和。

逆文本频率可表示为：

$$idf_i = \log \frac{|D|}{|\{d: d \ni t_i\}|} \quad (2)$$

逆文本频率是一个词语普遍重要性的度量。某个特定词语的 IDF，可由总文件数目除以包含该词语文件的数目，再将得到的商取对数得到。其中分子表示语料库中的文件总数，分母表示包含 $t_i$ 的文件数目，但如果该词不在语料库中，就会导致分母为零，因此一般情况下使用 $1 + |\{d: d \ni t_i\}|$ 作为分母。然后 TF-IDF 的值就等于 $tf_{i,j} \times idf_i$ 。

python 中的 sklearn 库中提供了 TfidfVectorizer 方法，使用 TfidfVectorizer 初始化向量空间模型，可以很方便的实现特征权重的计算。得到结果权重矩阵 tdm，权重矩阵是一个二维矩阵， $tdm[i][j]$ 表示，第 j 个词（即该词在词典中的序号）在第 i 个类别中的 TF-IDF 值。通过 TfidfVectorizer 方法，将训练集 Bunch 数据结构里面 contents 存储的内容转换为 tdm 形式，下面以表 2 的形式展现。

表 2 权重矩阵样例表

tdm[i][j]	tf-idf
(0, 15534)	0.26393380316959947
(0, 50042)	0.17737190173771117
⋮	⋮
(1, 22924)	0.08023922305375071
(1, 22658)	0.13193285401162472
⋮	⋮
(7221, 67790)	0.1656950975566633

在表 2 中，左边括号的第一个数表示第几类，计算得出总的分为 7221 个类别；第二个数表示在这类中的第几个词，右边表示该词的 TF-IDF 权重值。

### 3.1.5 文本分类

中文文本分类<sup>[7]</sup>通常采用基于统计的文本分类方法，主要有朴素贝叶斯、最近邻方法、支持向量机和 LDA 模型。这些方法实现机制比较简单，文本分类效果良好。本题采用朴素贝叶斯分类器<sup>[8]</sup>。

已知待分类数据对象 X，预测 X 所属类别，计算方式如下：

$$y_k = \arg \max_{y_k \in Y} (P(y_k|X)) \quad (3)$$

所得 $y_k$ 即为 X 的所属类别。上式表示，已知待分类数据对象 X 的情况下，

分别计算  $X$  属于  $y_1$ 、 $y_2$ 、...、 $y_k$  的概率，选取其中概率的最大值。

根据贝叶斯定理， $P(y_k|X)$  计算方式如下：

$$P(y_k|X) = \frac{P(X|y_k)P(y_k)}{P(X)} \quad (4)$$

也可以更直观的表达这个式子：

$$P(\text{类别}|\text{特征}) = \frac{P(\text{特征}|\text{类别})P(\text{类别})}{P(\text{特征})} \quad (5)$$

在 python 里的 sklearn 库中，朴素贝叶斯分类器有封装好了的函数：MultinomialNB。它获取训练集的权重矩阵和标签，进行训练，然后获取测试集的权重矩阵，进行预测。

最后用 F-Score 对分类方法进行评价：

$$F_1 = \frac{1}{n} \sum_{i=1}^n \frac{2P_iR_i}{P_i + R_i} \quad (6)$$

其中  $P_i$  为第  $i$  类的查准率即精确率， $R_i$  为第  $i$  类的查全率即召回率。

最后通过 sklearn 库中的 metrics.precision\_score、metrics.recall\_score 和 metrics.f1\_score 分别计算出了测试集的精确率、召回率以及 F-Score 值，结果见表 3。

表 3 分类模型评价结果

精确率	0.967
召回率	0.967
F-Score	0.967

如果把每一类的预测看作一个二分类问题，即用正类和负类来表示，那么精确率就是正确预测为正类的占全部预测为正类的比例，本题计算得出精确率为 0.967；召回率就是正确预测为正类的占全部实际为正类的比例，计算出召回率为 0.967；F-Score 是两者的综合评价标准，得到结果为 0.967。由 F-Score 值可以看出，此模型的测试结果还是很不错的。

## 3.2 问题二的分析方法与过程

本题的主要目的是把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，也就是文本聚类。常用的文本聚类方法是 K-means 聚类方法，但是该方

法需要事先设定聚类数，就本题来说，数据量较大，不知道到底能聚出来多少主题。Single-pass 聚类算法不需要指定聚类数量，可以通过设定相似度阈值来限定聚类数量。通过实验，将阈值  $\theta$  设为 0.045。考虑此因素，本题使用 Single-pass 聚类。

Single-pass 聚类<sup>[9]</sup>，中文名一般译作“单遍聚类”，是一种简洁高效的文本聚类算法。Single-pass 算法顺序处理文本，以第一篇文档为种子，建立一个新主题。之后再进行新进入文档与已有主题的相似度，将该文档加入到与它相似度最大的且大于一定阈值的主题中。如果与所有已有话题相似度都小于阈值，则以该文档为聚类种子，建立新的主题类别。

其算法流程图如下：



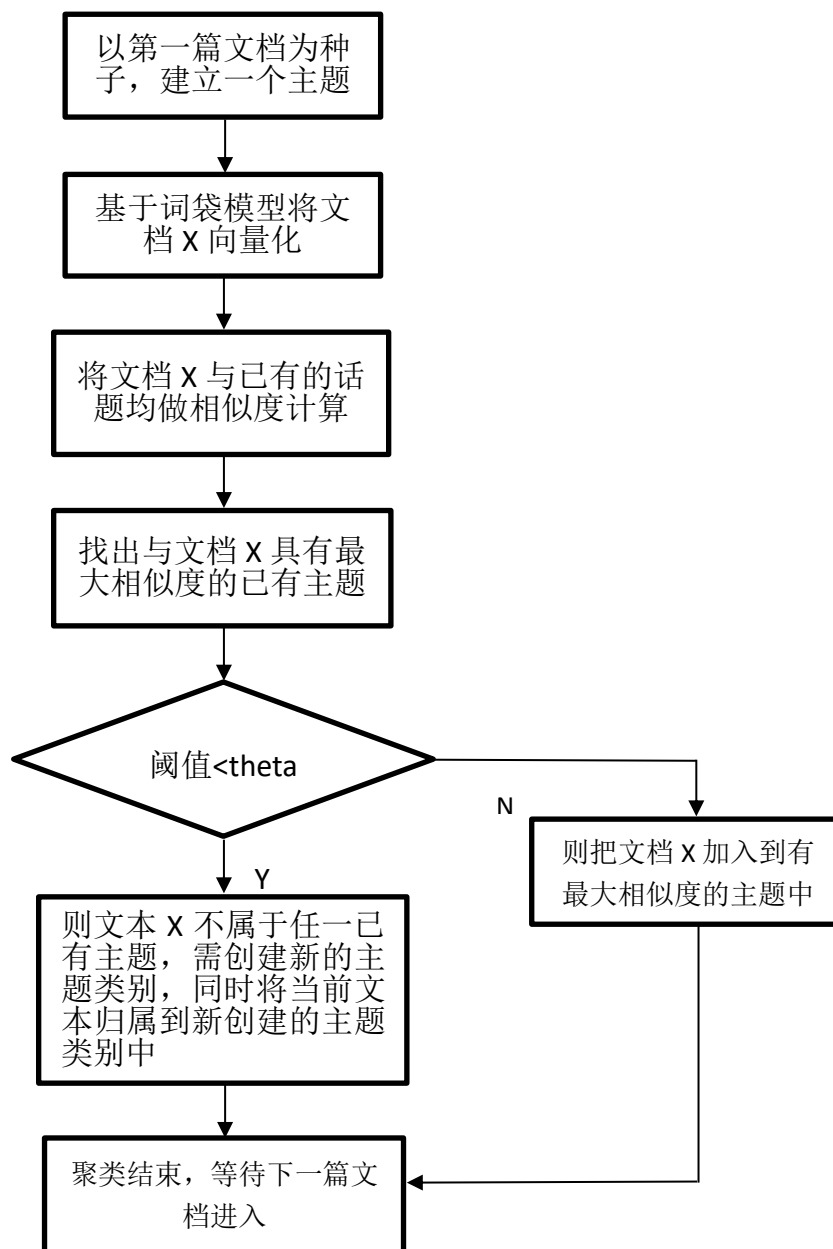


图 6 问题 2 算法流程图

### 3.2.1 文本预处理

将留言文本读取到 python 之后, 进行分词与去除停用词。同样使用 python 的 jieba 库中的精准模式进行分词。在分词后, 同样进行去除停用词操作, 有标点符号、语气助词等。这些词汇可以看作无效词, 会以噪音的形式影响后续运算, 需要去除。

```
['座落在', '市', 'A3', '区联', '丰路', '米兰', '春天', 'G2', '栋', '320', '一家', '名叫', '一米阳光', '婚纱', '艺术摄影', '影楼', '年单', '工作室', '营业额', '上百万', '地处', '居民楼', '内部', '蛮长', '时间', '请', '税务局', '工商局', '查', '一米阳光', '有没有', '纳税', '操作']
```

图 7 分词并去停用词后的内容

如图 7, 是附件 3 中留言编号为 188006 的留言详情, 在分词并去除停用词

之后的结果图。

### 3.2.2 基于词袋模型向量化

词袋模型 Bag of Words (BoW) 最早出现在自然语言处理 (NaturalLanguage Processing) 和信息检索 (Information Retrieval) 领域。在信息检索中, 词袋模型假定对于一个文本, 忽略其词序和语法, 句法, 将其仅仅看作是一个词集合, 或者说是词的一个组合, 文本中每个词的出现都是独立的, 不依赖于其他词是否出现, 或者说当这篇文章的作者在任意一个位置选择一个词汇都不受前面句子的影响而独立选择的。

文本特征选择及词典构建。对于长文本文档, 在构建词典前有必要通过特征选择方法来选择一批特征词, 然后使用这些特征词构建词典, 否则构建的词典将会非常庞大, 即不利于存储, 也不利于后续词频统计运算等。采用 VSM (vector space model) 得到文档的空间向量表示, 利用 python 中 gensim 模块里面的 corpora 可以获取分词后词汇和词汇 id 的映射关系, 形成字典, 见图 8。然后将其转为向量<sup>[10]</sup>表示, 见图 9。最后同样使用 TF-IDF 权重方法计算特征向量的权重矩阵, 见图 10。

```
{'320': 0, 'A3': 1, 'G2': 2, '一家': 3, '一米阳光': 4, '上百万': 5, '丰路': 6, '内部': 7, '区联': 8, '名叫': 9, '地处': 10, '婚纱': 11, '居民楼': 12, '工作室': 13, '工商局': 14, '市': 15, '年单': 16, '座落在': 17, '影楼': 18, '操作': 19, '时间': 20, '春天': 21, '有没有': 22, '查': 23, '栋': 24, '税务局': 25, '米兰': 26, '纳税': 27, '艺术摄影': 28, '营业额': 29, '蛮长': 30, '请': 31, '10': 32, 'A6': 33, '作用': 34, '充分发挥': 35}
```

图 8 获取分词后词汇与词汇 id 映射关系

如图 8, 是附件 3 中留言编号为 188006 的留言详情, 进行分词并去停用词之后, 将每个词建立一个对应的索引。

```
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 2), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (10, 1), (11, 1), (12, 1), (13, 1), (14, 1), (15, 1), (16, 1), (17, 1), (18, 1), (19, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1), (29, 1), (30, 1), (31, 1)], [(15, 1), (32, 1),
```

图 9 得到语句的向量表示

图 9 是将图 8 中得到的结果进行向量化, 每个元组中的第一个元素对应的是字典中词汇的 ID, 第二个对应于该词汇出现的次数。

```
[(0, 0.7678181833503347), (1, 0.23806024347515312), (2, 0.6579722563318935), (3, 0.3671406803367077), (4, 1.6742464940298165), (5, 0.6579722563318935), (6, 0.6985131196857612), (7, 0.41377157064129294), (8, 0.7678181833503347), (9, 0.6068968089096042), (10, 0.4123328973073482), (11, 0.8371232470149083),
```

图 10 获取语句的 TF-IDF 向量表示

图 10 是将得到的向量使用 TF-IDF 模型转换成的向量空间, 其中每个元组

的第一个元素是词汇 ID，第二个得到的是 TF-IDF 加权值。

### 3.2.3 相似度计算

将文档 X 与已有的所有话题均做相似度计算，可采用欧氏距离、余弦距离等，这里我们采用余弦相似度。余弦相似度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。相比距离度量，余弦相似度更加注重两个向量在方向上的差异，更多的用于对内容评分来区分兴趣的相似度和差异，同时修正了群众间可能存在的度量标准不统一的问题。

其公式如下：

$$\text{sim}(X,Y) = \cos \theta = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|} \quad (7)$$

其中分母表示两个向量的长度，分子表示两个向量的内积。当两个文档的向量表示夹角余弦等于 1 时，这两个文档完全重复；当夹角的余弦值接近于 1 时，两个文档相似；夹角的余弦值越小，两个文档越不相关。

### 3.2.4 找出最大相似度的已有主题

1) 通过对文档 X 与已有的所有话题做相似度计算后，找出与文档 X 具有最大相似度的已有主题；余弦值接近 1，夹角趋于 0，表明两个向量越相似，余弦值接近于 0，夹角趋于 90 度，表明两个向量越不相似。通过找一个验证集合，遍历可能的阈值，计算评价指标，确定最佳阈值。

2) 将计算出的相似度与阈值  $\theta$  比较，若相似度值大于阈值  $\theta$ ，则把文档 X 加入到有最大相似度的主题中，结束聚类，等待下一篇文章进入。

3) 若得到的相似度值小于阈值  $\theta$ ，则文档 X 不属于任一已有主题，则需创建新的主题类别，同时将当前文本归属到新创建的主题类别中，结束聚类，等待下一篇文章进入。

### 3.2.5 输出结果

最后，得到结果有 444 个类别，然后按照每个主题的留言次数，保存为“热点问题留言明细表.xls”。结构如表 4，详情请见附件“热点问题留言明细表.xls”。

表 4 热点问题留言明细表样表

问题 ID	留言编号	留言用户	留言主题	留言时间	留言详情	点赞数	反对数
-------	------	------	------	------	------	-----	-----

1	188801	A909180	投诉滨河苑 针对广铁职 工购房的霸 王规定	2019/8/1 0:00:00	尊敬的张市长， 您好！我叫李建 义，来自湖北仙 桃，虽然已经在 ...	0	0
1	190337	A00090519	关于伊景园 滨河苑捆绑 销售车位的 维权投诉	2019/8/23 12:22:00	投诉伊景园.滨 河苑开发商捆 绑销售车位！...	0	0
:	:	:	:	:	:	:	:
2	190822	A00031618	A 市国王陵考 古遗址公园 一期整治项 目周边环境 常年恶劣	2019/7/17 14:21:05	地处 A 市国王 陵考古遗址公 园一期整治...	1	0
2	191872	A00031618	请 A 市加快 轨道交通建 设力度	2019/3/1 15:19:28	地处中部中心 城市的 A 市在 高铁和...	9	2
:	:	:	:	:	:	:	:
3	191508	A00077323	A2 区中建幼 儿园 2019 年 小班不合理 操作	2019/8/30 23:18:27	尊敬的领导：A2 区中建幼儿园 ...	0	0
3	191860	A909138	A5 区时代阳 光大道中建 嘉和城附近 防护绿地被 侵占	2019/11/19 18:17:24	您好，我是 A 市 A5 区时代阳光 大道中...	0	0
:	:	:	:	:	:	:	:
444	360114	A0182491	A 市经济学院 体育学院变 相强制实习	2017/6/8 17:31:20	书记您好，我是 来自西地省经 济学院体育学 院的	0	9

### 3.2.6 热度评价指标

得到结果后，按照下面的热度评价方法进行评价：

$$H = \frac{C + L}{D + 1} \times \left( \frac{D + 1}{365} \right) \quad (8)$$

其中，C 为同一主题内，不同用户的留言总数，L 为同一主题的点赞数减去反对数，D 为时间段，加 1 的目的是防止分母为零的情况出现。在乘号的右边，

相当于给予一个权重，解释为对此主题留言的时间段占一年的比重。根据计算出来的热度，将排名前五的保存为“热点问题表.xls”，其格式如表 5。详情请见附件“热点问题表.xls”。

表 5 热点问题表

热度排名	问题 ID	热度指数	时间范围	地点/人群	问题描述
1	42	6.616	2019/01/08 至 2019/12/17	A 市市民	A 市车贷诈骗案
2	116	5.775	2019/01/23 至 2019/08/19	A 市小区业主	违规租房装修
3	47	4.953	2019/01/04 至 2019/12/31	A 市小学生家长	孩子上学问题
4	61	2.022	2019/02/07 至 2019/11/19	A 市小区居民	交通规划问题
5	20	0.795	2019/01/04 至 2020/01/03	A 市小区居民	居民用水问题

在表 5 中，可以看到热度排名第一的是 A 市的车贷诈骗案，通过（8）式计算得出热度指数为 6.616。在 2019/01/08 至 2019/12/17 这期间内，有 24 个群众留言说明此事件，并获得了 2397 个点赞数。详情请见附件“热点问题留言明细表.xls”。

### 3.3 问题三的分析方法与过程

#### 3.3.1 文本预处理

（1）将原 Excel 文件里面的留言详情与答复意见分别保存为 txt 文档，以便于后续调用。

（2）利用所有留言详情用 python 读取并以列表形式存储，之后对其进行分词并去除停用词。

#### 3.3.2 LSI 向量空间的构建

（1）例如有两个句子：“Jane wants to go to Shenzhen.”“Bob wants to go to Shanghai.”首先，把所组成语句的词都装进一个袋子，每个词语之间相互独立，从而构成一

个词袋，将袋子里的 Jane、wants、to、go、Shenzhen、Bob、Shanghai 建立一个数组用于映射匹配[Jane、wants、to、go、Shenzhen、Bob、Shanghai]。

将上面两个例句分别用以下两个向量表示，对应的下标与映射数组的下标相匹配，其值为该词语出现的次数[1,1,2,1,1,0,0]、[0,1,2,1,0,1,1]。从而利用词袋模型（Bag-of-Words）来提取文本特征。词袋模型是个在自然语言处理和信息检索(IR)下被简化的表达模型。词袋模型被广泛应用在文件分类，词出现的频率可以用来当作训练分类器的特征。

（2）将留言详情和答复意见已经构建好的词袋语料库，用 LSI（Latent Semantic Indexing）模型构建一个 LSI 向量空间<sup>[11]</sup>。LSI 是概率主题模型的一种，LSI 通过奇异值分解的方法计算出文本中各个主题的概率分布。每个主题上的概率就是文章对于这个主题的隶属度，同一个文本可能夹杂着多个主题，只是对应的各个主题的概率不相同。

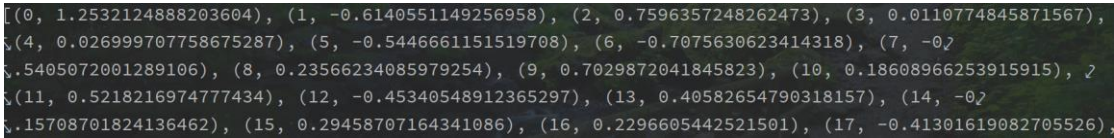


图 11 LSI 向量空间

如图 11 所看到的，这是附件 4 中留言编号为 2549 的答复意见一小部分 LSI 向量空间展示，其中每个元组的第一个元素为文本的主题索引，第二个为该文本对这个主题的概率也就是隶属度。

3.3.3 相似度计算及评价定义

本题依旧采用余弦相似性来测量留言详情与答复意见之间的相似性，相似性越高，就视为相关部门对群众留言的答复的相关性越高，然后相关性越高，表明对留言的答复越完整。

3.3.4 评价结果

表 6 相关性大于 85%的群众留言编号

留言编号	相似度	留言编号	相似度
50409	0.969	9199	0.881
18413	0.968	98413	0.880

9128	0.938	94222	0.878
58468	0.927	104077	0.876
139949	0.919	104025	0.870
17173	0.915	74615	0.864
4865	0.913	168031	0.862
107296	0.909	88150	0.859
4423	0.908	145252	0.857
126648	0.902	17525	0.856
99633	0.900	30206	0.855
107209	0.899	96760	0.855
105696	0.897	119178	0.855
158613	0.896	10210	0.855
7107	0.894	10484	0.853
117369	0.888	4090	0.851
8764	0.885	9113	0.850

如表 6 所示，在相关性计算中，有 34 个留言答复与留言内容的相关性达到 85%以上。

表 7 相关性小于 10%的部分群众留言编号

留言编号	相似度		留言编号	相似度
124039	0.0998	...	12303	-0.027
17135	0.0996	...	10029	-0.041
117104	0.0977	...	108077	-0.044
88236	0.0976	...	76462	-0.047
27590	0.0974	...	30019	-0.068

表 7 为留言答复与留言内容的相关性在 10%以下的部分留言编号，一共由 307 个留言答复与留言内容的相关性在 10%以下。

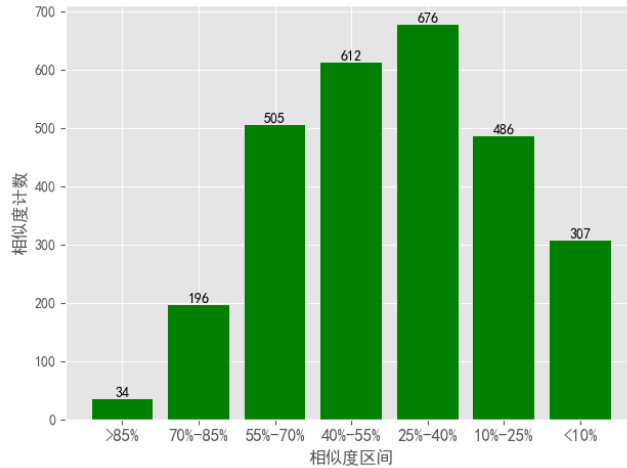


图 12 相关部门对群众留言的相关性图

图 12 是利用 python 画出的，可以看出，大部分对群众留言的答复还是较为相关且完整的，下面分别是相关程度最高的相关程度最低的对留言的答复，见图 13 和图 14。

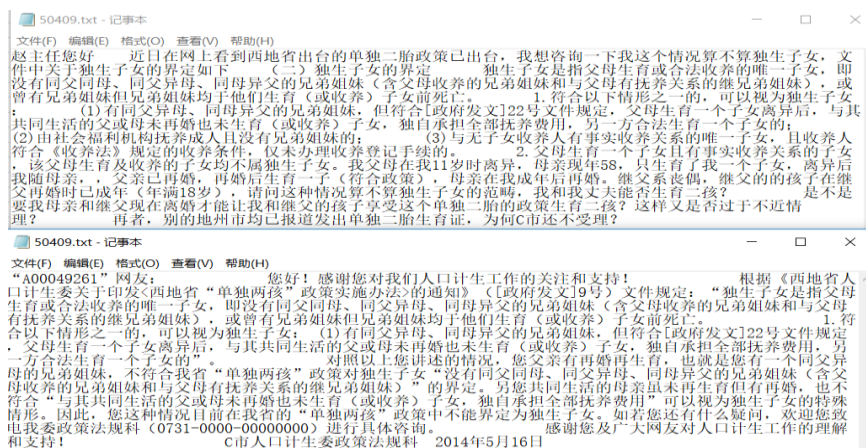


图 13 留言编号为 50409 的留言内容与答复意见

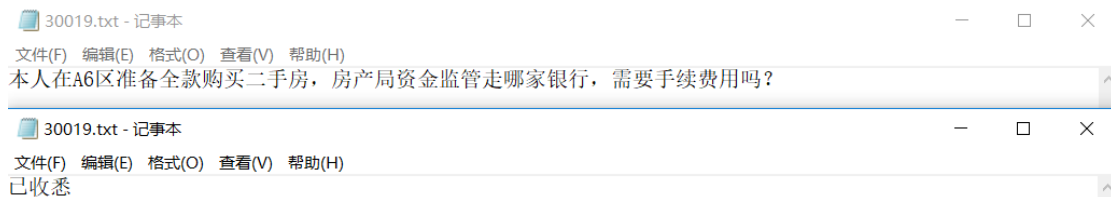


图 14 留言标号为 30019 的留言内容与答复意见

从以上两张留言内容与答复意见图的情况看，以相似性来衡量留言内容与答复意见的相关性和完整性是可取的。

## 4. 总结

本文综合运用统计、机器学习、自然语言处理与文本挖掘的相关知识，借助



Python 软件对“智慧政务”进行文本挖掘，得到的结果具有极大的经济和社会价值。结论如下：

1) 对于问题一：首先通过将留言详情以及对应的一级分类按照 8: 2 的比例划分为训练集与测试集后，借助 python 里面的 jieba 库进行分词，去除停用词，以 sklearn 库里面的 bunch 函数形式存储，再用 pickle.dump 保存为二进制文件，构建 TF-IDF 词向量，最后应用朴素贝叶斯分类器对留言进行分类，并利用 F-score 对分类方法进行评价。得到了 F-Score 分值为 0.967 的测试结果，解决了建立关于留言内容的一级标签分类模型的问题。

2) 对于问题二：首先利用 Single-pass 聚类算法进行文本处理。通过实验，取最优阈值  $\delta$  为 0.045。将新的主题文档与已有主题进行相似度比较，用条件判断语句对相似度的值与阈值  $\delta$  进行比较。以此来找出主题数或者建立一个新的主题，解决了将某一时段内反映特定地点或特定人群问题的留言归类问题。

3) 对于问题三：首先将留言详情与答复意见分别保存为 txt 文档，所有留言详情用 python 读取并以列表形式存储，并对其进行分词并去除停用词；其次用词袋模型 (Bag-of-Words) 来提取文本特征，将构建好的词袋语料库，用 LSI 模型构建一个 LSI 向量空间，构建词袋模型并转为 LSI 向量空间并对其中的每个文档建立索引；最后依次进行全匹配查询，利用余弦相似性来度量两个向量之间的相似性，并取出对应的测试结果存到 Excel 中。得到了群众留言与相关部门答复意见的留言编号以及它们之间的相似度，解决了从答复的相关性、完整性等角度对答复意见的质量评价问题。

## 5. 模型评价

### 5.1 模型的优点

朴素贝叶斯算法假设了数据集属性之间是相互独立的，因此算法的逻辑性也很简单，并且算法比较稳定，当数据呈现不同的特点时，朴素贝叶斯的分类性能不会有太大的差异。当数据集属性之间的关系相对比较独立时，朴素贝叶斯分类算法会有较好的效果。

single-pass 聚类是一种简洁高效的聚类算法，single-pass 聚类是针对流式文

本进行聚类的，即来一条聚一条。它同时还是一种增量聚类算法（Incremental Clustering Algorithm），可以很好的应用于话题监测与追踪、在线事件监测等社交媒体大数据领域。

## 5.2 模型的缺点

属性独立性的条件同时也是朴素贝叶斯分类器的一个不足之处。数据集属性的独立性在很多情况下是很难满足的，因为数据集的属性之间往往都存在着相互关联，在分类过程中出现这种问题，就会导致分类的效果大大降低。

single-pass 聚类的不足之处主要表现在该方法具有输入次序依赖特性，即对于同一聚类对象按不同的次序输入，会出现不同的聚类效果，并且其聚类的相似阈值也不好设定。

## 5.3 模型的改进

在基于 LSI 进行文档相似度检索，还有可以优化的方面，其一是对语义的挖掘力度还不够，这个属于硬匹配，意义相近但说法不一样的词汇的相似性还不能很好的度量。其二是速度有待改进，当检索文档数据量相当大时，这个检索系统的效率就会非常低。

在利用 TF-IDF 向量表示时，还可以采用 Doc2vec 或者 Skip-thoughts 等算法直接获取文档的向量表示；计算文本相似度的余弦距离，可以尝试用欧式距离、jaccard 或者 hellinger 等算法。

## 6. 参考文献

- [1] Jones Q,Ravid G,Rafaeli S. Information Overload and the Message Dynamics of Online Interaction Spaces:A Theoretical Model and Empirical Exploration[J]. Information Systems Research, 2004, 15(2):194-210.
- [2] Hearst M A.Untangling Text Data Mining[J].Proceedings of ACL-99, 2002:3-10.
- [3] Kolovou A.Machine Learning Methods for Opinion Mining In text:The Past and the Future[J]. 2019,13:429-457.
- [4] Uriarte,María,Chazdon R L.Incorporating natural regeneration in forest landscape restoration in tropical regions:synthesis and key research gaps[J]. Biotropica, 2016, 48(6):915-924.
- [5] Hotho, A.,Nürnberger, A., & Paa, G. A brief survey of text mining[J]. LDV Forum-GLDV Journal for Computational Linguistics and Language Technology, 2015, 20, 19-62.
- [6] Kumar B S,Ravi V. A survey of the applications of text mining in financial domain[J]. Knowledge-Based Systems, 2016, 114:128-147.
- [7] 廖一星,严素蓉.基于 Python 的中文文本分类的实现[J].2016, 032(012):6-6,14.
- [8] 王国才.朴素贝叶斯分类器的研究与应用[D].重庆交通大学.
- [9] 黄建一,李建江,王铮, et al. 基于上下文相似度矩阵的 Single-Pass 短文本聚类[J].计算机科学, 2019, 46(04):56-62.
- [10] 于政. 基于深度学习的文本向量化研究与应用[D].2016.
- [11] 龙军, 彭毅.基于 LSI/SVD 的文本分类方法研究[J]. 微计算机信息, 2009, 25(30):10-12.