“智慧政务”中的文本挖掘应用

摘要

随着便利的网络问政平台走进千家万户，大量的民意社情数据随之产生。因此，如何利用大数据、人工智能等技术，发展智慧政务平台，让这一平台能够智能、准确地对大量数据进行处理，成为了当下政府工作中的一大热点问题。

针对问题一：首先对附件二中的大量数据进行预处理，利用 jieba 中文分词工具进行分词和去停用词。接着用 TF-IDF 和 N-Gram 进行特征提取。为了避免数据的高维度及同义词对分类的影响，本文用截断式奇异值分解对数据降维，并进行特征缩放。在留言分类时、LinearSVC 分类模型进行留言的分类，最后用 F-score 评价模型。

针对问题二：本文先对附件三中的数据进行预处理，添加词典标注词性，接着利用 jieba 中文分词工具进行分词和去停用词。接着用 tf-idf 算法进行特征提取，为了提高后续聚类准确度，依照词性对特殊词性的词改变权重。为了避免数据高纬度影响聚类准确度，降低聚类效率，本文用截断式奇异值分解对数据降维，并进行特征缩放。在文本聚类时，本文对比了 K-Means 聚类、DBSCAN 聚类、HDBSCAN 聚类三种聚类算法，最终选用了 HDBSCAN 算法.聚类过后，依照留言条数、留言密度、关注数、信息充实度四个指标构造热度评价指标，对聚类结果筛选出热度前五的问题，提取词性为”地名人名”的词作为热点问题的地点人群。用 TextRank 算法提取热点问题排名前 10 的关键词，形成热点问题的问题描述。

针对问题三：从答复意见的相关性、完整性、可解释性、答复效率、文本长度五个方面建立了指标，并通过熵值法确定各指标权重，最后计算出答复意见质量的综合评价得分。

关键词：奇异值分解　TF-IDF HDBSCAN 熵值法 LinearSVM

Abstract

As a convenient online political inquiry platform entered thousands households, a large amount of public opinion and social situation data were generated. Therefore, how to use big data, artificial intelligence and other technologies to develop a smart government platform to enable this platform to process large amounts of data intelligently and accurately has become a hot issue in government work.

In Annex II, we first apply jieba Chinese word segmentation tool to segment and remove stop words for preprocessing a large amount of data. Then use TF-IDF and N-Gram for feature extraction. In order to avoid the impact of high-dimensional data and synonyms on classification, the truncated singular value decomposition is used to reduce the dimension of the data and perform feature scaling. Finally, we use LinearSVC classification model and F1-score evaluation model to classify the message and evaluate the results.

For preprocesses the data in Annex III, we add a dictionary to mark parts of speech, and then use jieba Chinese word segmentation tool to segment and remove stop words. Then use TF-IDF algorithm for feature extraction. In order to improve the accuracy of subsequent clustering, the weight of words with special parts of speech is changed according to parts of speech. In order to avoid the high latitude of the data from affecting the clustering accuracy and reducing the clustering efficiency,

we also apply truncated singular value decomposition to reduce the dimension of the data and perform feature scaling. In text clustering, we compare the three clustering algorithms of K-Means clustering, DBSCAN clustering, and HDBSCAN clustering to show the best performance of HDBSCAN algorithm. After clustering, we construct the heat evaluation index based on the number of messages, message density, the number of concerns and the information of text, select the top five questions of the heat by the clustering results, and extract the words whose part of speech of the hot topic is the names of place names and person names as the location crowd of the questions. We finally use the TextRank algorithm to extract the top 10 keywords of hotspot issues to form a hotspot problem description.

The indicators are established based on relevance, completeness, interpretability, response efficiency, and the length of text, and the entropy method are used to determine the weight of each indicator. Finally, we calculate the comprehensive evaluation value.

Keywords: TSVD TF-IDF HDBSCAN entropy LinearSVM

目录

1、简介

近年网络问政平台逐渐普及，越来越多与社情民意相关的文本数据随之产生，本次建模的目标是对所给数据进行去重、去空、jieba 分词等数据预处理后，再利用 TF-IDF、SDV 等方法处理数据，达到以下三个目标：

其一，利用的方法，建立关于所提供的群众留言数据的一级标签分类模型，并用 F-Score 原理对此分类方法进行评价。

第二，对所给数据根据其所反映的特定地点、特定人群进行分类，给出热度评价指标，评价分类结果。并依此建立热度前五留言的“热点问题表”及“热点问题留言明细表”。

第三，设计一套可实现的留言答复质量评价方案。

2、预处理

所提供的群众留言存在冗余的现象，因此，对于原始数据需要首先进行去重处理，去重后数据剩余 9052 条。为去除数据中存在的英文及符号，对之后的文本分类产生干扰，对去重后的数据再进行过滤特殊符号的处理。

对于去重后的数据我们进行分词处理。分词指的是，将完整的一句话根据其语义分拣成一个词语项集[1]，即将一句话按照字典库中的词语为一个个单元，分割成一个词语集，使得计算机能够统计出每个语句中的词语、词频、词性，从而分析出热点词汇，这里的中文分词采用“jieba”词库作为分割语句的依据。

Python 中的中文分词包“jieba”，其算法是采用前缀词典对词图进行扫描，而后构出有向无环图来表示所有可能构成词语的情况，再用动态规划查找最大概率路径，最后使用 HMM 模型来计算未登陆词的成词能力，且“jieba”库中包含约 20000 条的中文词汇，分词的精确度大大提升。本次的数据处理采用“jieba”分词的精确模式，即以最大精度对语句进行分割，为文本的向量化表示做准备。

去重、分词后的列表中的留言数据，存在无具体含义、但出现频率高的中文字词及字符间的空格，例如：“是”、“的”等字词，对于文本分类结果并无影响。

为了简化分类结果、提高程序运行效率，需建立一个停用词表用于存放此类字词，并调用函数将列表中所含的停用词去除。

在文本处理中，停用词是指那些功能极其普遍，与其他词相比没有什么实际含义的词，它们通常是一些单字，单字母以及高频的单词，比如中文中的“我、的、了、地、吗”等，英文中的“the、this、an、a、of” 等。对于停用词一般在预处理阶段就将其删除，避免对文本，特别是短文本，造成负面影响。本文所用的停用词，取自哈工大常用停用词表。

对处理后的数据使用 TF-IDF 算法提取特征，TF-IDF 算法，比起 One-Hot 算法，多了词频“TF”，即词语在数据中出现的频数，因此，此算法返回的统计结果更为直观。若 n_i 代表词 i 在留言数据中出现的次数， N 代表文本的词汇总个数，那么， TF 为：

$$TF = \frac{n_i}{N}, \quad (1)$$

“IDF”代表逆文本频率，即返回一个词在语料库中所有文本中出频率，反映词语在文本中的重要性[2]。若该词汇在留言数据中出现的次数多，但是在其他与留言数据无关的文本数据中出现的次数不多，那么此词语就可做为文本的特征之一，能够依此提取出文本中具有代表性的词语。若 D 代表文本库中的文本总数，D_j 代表文本中含有词 j 的文本总数，同时考虑到未登录词的逆文本频率的计算[3]，“IDF”的算法为：

$$IDF = \log \frac{N+1}{N_j+1} + 1, \quad (2)$$

TF-IDF 的算法为：

$$TF - IDF = TF \times IDF \text{ 。} \quad (3)$$

TF-IDF 提取的矩阵过于稀疏，直接用于文本分类效果不好，我们使用奇异值分解对其进行降维。奇异值分解（SVD）是一种重要的矩阵分解的方法，也是一种矩阵特征提取方法，矩阵的奇异值反映了矩阵向量间的内在代数本质，具备良好的数值稳定性和几何不变性[4]。群众留言向量可以组成一个矩阵 A，矩阵 A 的同一行的元素表示同一句留言的词语，同一列的元素代表同一个词汇在不同留言中的 $TF\text{-}IDF$ 值，例如 a_{ij} 代表第 i 条留言中第 j 个词语的 $TF\text{-}IDF$ 值。

运用奇异值分解矩阵 A 后，矩阵 A 变为三个矩阵，其变换公式为：

$$A = U \sum V^T, \quad (4)$$

U 矩阵的每一行是同一句留言中的词语，每一列是一个主题，即完成

了留言的主题的分类。Σ矩阵是联系U与V^T的矩阵，每一行为一个主题，每一列为一个语义类，表示了某个词与某个主题之间的相关性。V^T矩阵的每一行为一个语义类，每一列为一个词语，此矩阵对留言数据中的词进行了近义词分类。TSVD 与一般 SVD 不同的是它可以产生一个指定维度的分解矩阵，可以实现降维。本题采用截断式奇异值分解，将留言数据经过降维后，变为 115 维。

为了提高算法的收敛速度以及提高模型的泛化性我们进行了特征缩放，我们采取标准化将数值缩放到 0 附近，且数据的分布变为均值为 0，标准差为 1 的标准正态分布。其数学公式为：

$$x' = \frac{x - \bar{x}}{\sigma} 。 \qquad (5)$$

3、群众留言分类

如图 1 所示对于留言进行加工处理后输出预测结果。

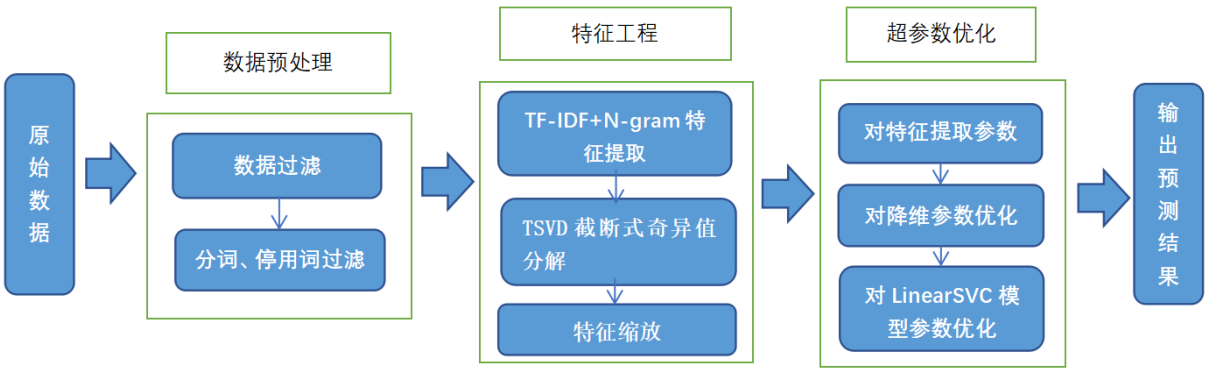

图 1

首先对预处理后的数据使用 N-Gram 模型提高获取的信息量。N-Gram 在 NLP 领域有着非常广泛的应用，在自然语言处理领域中，我们往往是通过语料信息来判断一个句子是否自然语言中的一句话，

具体来说，我们是这样做的：通过将每联系的 N 个单词设置成一个整体，并将这些字段进行切分构成词袋。我们采用 N=2 的 2-gram 模型，来提高获取的信息量。

对处理好的特征本文采用 SVM 分类模型，对预处理后的留言数据特征向量(x_i, y_i)进行分类。SVM 模型是一种二分类模型，其通过求最优分类面$w^T x+b=0$（二维为直线），来判断数据是否属于这一类。该平面不仅能将两类训练样本正确分开，而且要使分类间隔(Margin)最大，即让两类中离分类超平面最近的样本且平行于分类超平面的两个超平面间距离最小[5]

$$\operatorname{margin}(w,b)=\min_{\substack{w,b\\x_i}}\frac{1}{\|w\|}\left|w^T x_i+b\right|, \tag{6}$$

分类器模型为：

$$f(x)=sign\left(w^T x_i+b\right), i=1,\mathrm{L}\ ,N\ , \tag{7}$$

其约束条件为：

$$\begin{cases}\min\limits_{w,b}\dfrac{1}{2}w^T w\\ y_i\left(w^T x_i+b\right)\geq 1, i=1,\mathrm{L}\ ,N\ 。\end{cases} \tag{8}$$

本文利用 TF-IDF+2-Gram 提取的留言数据特征向量，使用 SVM 分类器进行分类。

对于模型的评价，我们使用 F1 值进行评价。

已知：

TP（真正类）：指实际为正类，模型预测结果也为正类的情况。

TN（真负类）：指实际为负类，模型预测结果也为负类的情况。

FP（假正类）：指实际为负类，但模型预测结果为正类的情况。

FN（假负类）：指实际为正类，但模型预测结果为负类的情况。

若用“+”代表正类，“-”代表负类，那么以上关系可表示为：

		预测结果	
		+	−
实际	+	TP	FN
	−	FP	TN

表 1

从上表可以看出，TP 与 TN 代表模型预测正确，可准确分类出不同类别的样本，而 FP 和 FN 则为模型预测失误的情况。那么，准确率、精确率、召回率、F_1值的定义如下：

准确率（Accuracy）：表示模型预测准确的比率。

$$\frac{TP+TN}{TP+TN+FP+FN}, \tag{9}$$

精确率（Precision）：表示模型正确预测为正类的样本数，占预测为正类样本数的比率。

$$\frac{TP}{TP+FP}, \tag{10}$$

召回率（Recall）：表示模型正确预测为正类的样本数，占实际正类样本的比率。

$$\frac{TP}{TP+FN}, \tag{11}$$

F_1 score：表示召回率与精确率的调和平均值。

$$F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision}。 \quad (12)$$

学习曲线是学习算法的一个很好的合理检验（sanity check）。学习曲线是将训练集误差和交叉验证集误差作为训练集实例数量的函数绘制的图表。我们绘制如下图 2 的学习曲线，并利用该曲线查看模型的拟合情况。

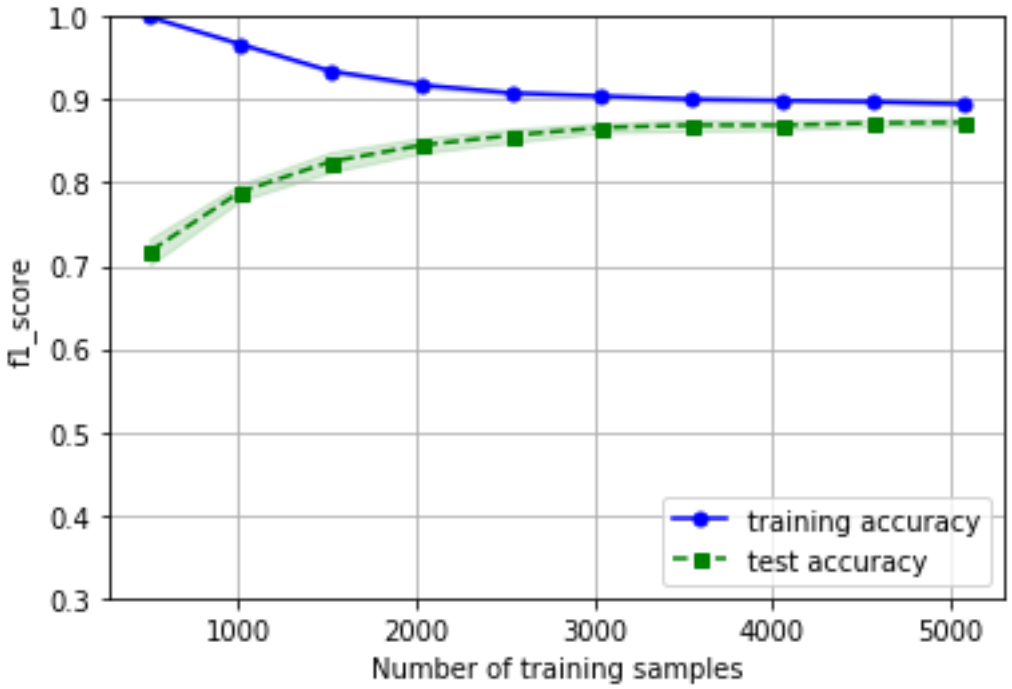

图 2

4、热点问题挖掘

如图 3 所示对于对于数据进行加工处理后进行聚类并提取热点问题。

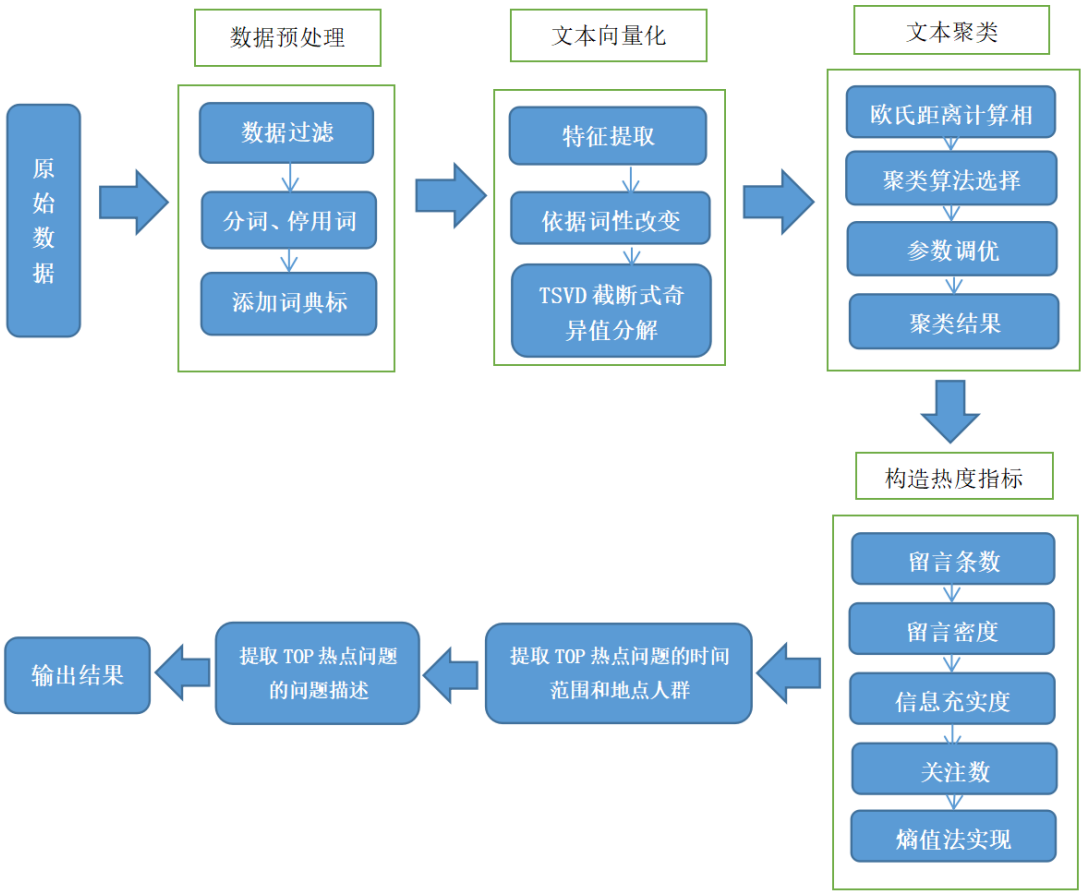

图 3

经过观察，所给的数据有 4327 条，其中大部分为文本格式，而数据中有着大量空白和没用的字符，如果不处理会对分词有影响，数据中还有大量的噪声，如果不做处理会对后续的聚类分析造成干扰。所以，要先对数据进行预处理。清理空白字符和未知字符，提取文本中的英文、数字和中文。

我们认为在后续聚类中，地名对聚类结果有着很重要的影响，为了提高分词质量，将文本中出现频率较高的地名找出，建立一个词典加入 jieba 分词的词库中，保证了在分词时这些地名能够被准确分割。同时，在后续的过程中有依据词语的词性进行处理，故对句子中每个词的词性进行标注。

我们采用 jieba 分词，对语句进行分割，发现结果有很多标点符

号和无意义的字符，会对后续分析产生影响，故引用停用词表对分词结果进行过滤。对过滤后的文本采用 TF-IDF 算法，将上文提取的词集转化为权重向量，得到一个词集的特征矩阵。

词性为‘ns’，‘nr’，‘nz’，‘nt’的词相比于其他词性的词更能够反映出文本的特征，对聚类的帮助更大，因此，将词集中这些词性的词的权重增大。

我们采用的方法是以词集里的每个词为键，每个词的词性为值，建立一个字典，建立一个与词集长度相同的数组，数组里的每个值为1，对于数组与字典中对应的位置的数字，如果其在字典中的值为‘ns’，‘nr’，‘nz’，‘nt’，将其由 1 改为 1.3，1.3，1.2，1.3。将特征矩阵中的每个值乘以对应的数组中的数字，得到改变权重后的特征矩阵。最后采用截断式奇异值分解，将留言数据经过降维后，变为 50 维。

文本聚类就是从很多文档中把一些内容相似的文档聚为一类。文本聚类主要是依据著名的聚类假设：同类的文本相似度较大，而不同类的文本相似度较小。作为一种无监督的机器学习方法，聚类由于不需要训练过程，以及不需要预先对文本手工标注类别，因此具有一定的灵活性和较高的自动化处理能力，已经成为对文本信息进行有效地组织、摘要和导航的重要手段，为越来越多的研究人员所关注[6]。

相似度是用来衡量文本间相似程度的一个标准。在文本聚类中，需要根据个体的差异信息进行归类，也就是需要对文本信息进行相似

度计算，根据相似特性的信息进行归类。本文采用的是基于欧式距离公式计算文本相似度。

欧式距离也称欧几里得距离，是最常见的距离度量，衡量的是多维空间中两个点之间的绝对距离，其计算公式如下[7]：

$$dist(X,Y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}\ , \tag{13}$$

目前的聚类算法主要有基于划分、基于层次、基于密度、基于图形、基于模型和基于网格几种类型，其中，最为常用的算法有k-means 算法和 DBSCAN 算法。

K-means，即均值聚类，对于给定样本集，按照样本之间的距离大小，将样本集划分为 K 个簇。目标是让簇内的点尽量连接在一起，而让簇间的距离尽量大。目标函数：如果用数据表达式表示，假设簇划分为(C1,C2,...Ck)，则我们的目标是最小化平方误差 E：

$$E=\sum_{i=1}^{k}\sum_{x\in C_i}\left\|x-\mu_i\right\|_2^2\ , \tag{14}$$

其中，μ_i是簇C_i的均值向量，也称为质心，表达式为：

$$\mu_i=\frac{1}{C_i}\sum_{x\in C_i}x\ 。 \tag{15}$$

直接求目标函数 E 的值是一个 NP 难的问题。NP(non-deterministic polynomial)是指非确定性多项式，即可用一定数量的运算去解决多

项式时间内可解决的问题，而 NP 难不是一个 NP 问题。（NP->NPC->NP hard）

K-means 算法流程：首先确定 K 值（可以根据经验指定，或者交叉验证选择合适 k 值），其次是选择 k 个初始化的质心（质心的选择对结果的影响很大，因此质心选择不能太近），具体流程如图 4 所示。

Algorithm 1 基于划分的聚类 K-means 算法

Input: D: 输入样本集 $D=\{x_1,x_2,...,x_m\}$, K: 聚类的簇数 k, N: 最大迭代次数.

Output: C: 输出簇划分 $C=\{C_1,C_2,..,C_k\}$

1: (1) 从数据集 D 中随机选择 k 个样本作为初始的 k 个质心向量: $\{\mu_1,\mu_2,...,\mu_k\}$

2: (2) 对于 $n=1,2,..,N$:

3: (a) 将簇划分 C 初始化为 $C_t=\emptyset, t=1,2...k.$

4: (b) 对于 $i=1,2,...,m$, 计算样本 x_i 和各个质心向量 $\mu_j(j=1,2,...,k)$ 的距离: $d_{ij}=||x_i-\mu_j||_2^2$, 将 x_i 标记最小的为 d_{ij} 所对应的类别 λ_i。此时更新 $C_{\lambda_i}=C_{\lambda_i}\bigcup\{x_i\}$

5: (c) 对于 $j=1,2,...,k$ 对 C_j 中所有的样本点重新计算新的质心 $\mu_j=\frac{1}{|C_j|}\sum_{x\in C_j}x$

6: (d) 如果所有的 k 个质心向量都没有发生变化, 则转到步骤(3）.

7: (3) 输出簇划分 $C=C_1,C_2,...,C_k$

图 4

K-means 优点：

（1）原理简单，算法的实现较容易，收敛速度快。

（2）聚类效果较优。

（3）算法的可解释度比较强。

（4）主要需要调参的参数仅仅是簇数。

L-means 缺点：

（1）K 值的选取不好把握。

（2）对于不是凸的数据集比较难收敛。

（3）如果各隐含类别的数据不平衡，比如各隐含类别的数据量严重失衡，或者各隐含类别的方差不同，则聚类效果不佳。

（4）采用迭代方法，得到的结果只是局部最优解。

（5）对噪音和异常点比较的敏感[8]。

由于 K-means 聚类效果不佳，我们尝试使用 DBSCAN 聚类

DBSCAN 定义：是一种基于密度的聚类算法，可以通过样本分布的紧密程度决定，同一类别的样本之间是紧密相连的，不同样本是是分离的。

DBSCAN 原理：基于一组邻域来描述样本集的紧密程度，参数$(\varepsilon, Minpts)$用来描述领域的样本分布紧密程度。其中ε是某一样本的邻域距离阈值，$Minpts$描述了某一样本的距离为ε的邻域中样本个数阈值。

DBSCAN 思想：由密度可达关系导出的最大密度相连的样本集合，即为我们最终聚类的一个类别，或者说一个簇。假定样本集为$D=(x_1,x_2,\Lambda\ ,x_m)$，则有以下重要定义：

（1）**ε-邻域：**对于$x_j\in D$，其ε-邻域包含样本集D中与x_j的距离不大于ε的子样本集，即

$$N\varepsilon(x_j)=\{x_j\in D\mid distance(x_i,x_j)\le\varepsilon\}。\qquad(16)$$

这个子样本集的个数记为$|N\varepsilon(x_j)|$。

（2）**核心对象**：对于任一样本 $x_j \in D$，如果其 ε -邻域对应的 $N\varepsilon(x_j)$ 至少包含 $Minpts$ 个样本，即如果 $\left|N\varepsilon(x_j)\right| \geq Minpts$，则 x_j 是核心对象。

（3）**密度直达**：如果 x_i 位于 x_j 的 ε -邻域中，且 x_j 是核心对象，则称 x_i 由 x_j 密度直达。注意反之不一定成立，即此时不能说 x_j 由 xi 密度直达，除非且 x_i 也是核心对象。

（4）**密度可达**：对于 x_i 和 x_j , 如果存在样本样本序列 p_1, p_2, Λ, p_T , 满足 $p_1 = x_i, p_T = x_j$，且 $p_t + 1$ 由 p_t 密度直达，则称 x_j 由 x_i 密度可达。也就是说，密度可达满足传递性。此时序列中的传递样本 $p_1, p_2, \Lambda, p_{T-1}$ 均为核心对象，因为只有核心对象才能使其他样本密度直达。注意密度可达也不满足对称性，这个可以由密度直达的不对称性得出。

（5）**密度相连**：对于 x_i 和 x_j , 如果存在核心对象样本 x_k，使 x_i 和 x_j 均由 x_k 密度可达，则称 x_i 和 x_j 密度相连。

DBSCAN 聚类算法流程：

(1)输入：样本集 $D = (x_1, x_2, \Lambda, x_m)$，邻域参数 $(\varepsilon, Minpts)$，样本距离度量方式。

(2)初始化核心对象集合 $\Omega = \varnothing$, 初始化聚类簇数 $k = 0$，初始化未访问样本集合 $\Gamma = D$, 簇划分 $C = \varnothing$ 。

(3)对于 $j = 1,2,\Lambda, m$, 按下面的步骤找出所有的核心对象：

1） 通过距离度量方式，找到样本 x_j 的 ε -邻域子样本集 $N\varepsilon(x_j)$；

2) 如果子样本集样本个数满足$\left|N\varepsilon\left(x_j\right)\right|\geq Minpts$，将样本$x_j$加入核心对象样本集合：$\Omega=\Omega \mathrm{U}\left\{x_j\right\}$。

(4)如果核心对象集合$\Omega=\varnothing$，则算法结束，否则转入步骤5。

(5)在核心对象集合Ω中，随机选择一个核心对象o，初始化当前簇核心对象队列$\Omega cur=\{o\}$,初始化类别序号$k=k+1$，初始化当前簇样本集合$C_k=\{o\}$， 更新未访问样本集合$\Gamma=\Gamma-\{o\}$。

(6)如果当前簇核心对象队列$\Omega cur=\varnothing$，则当前聚类簇C_k生成完毕，更新簇划分$C=\{C_1,C_2,\Lambda\ ,C_k\}$,更新核心对象集合$\Omega=\Omega-C_k$， 转入步骤4。

(7)在当前簇核心对象队列Ωcur中取出一个核心对象o',通过邻域距离阈值ε找出所有的ε-邻域子样本集$N\varepsilon(o')$，令$\Delta=N\varepsilon(o')\mathrm{I}\ \Gamma$,更新当前簇样本集合$C_k=C_k\ \mathrm{Y}\Delta$,更新未访问样本集合$\Gamma=\Gamma-\Delta$,更新$\Omega cur=\Omega cur\ \mathrm{Y}(\Delta\mathrm{I}\ \Omega)-o'$，转入步骤6。

(8)输出：簇划分$C=\{C_1,C_2,\Lambda\ ,C_k\}$。

DBSCAN优点：

(1)可以对任意形状的稠密数据集进行聚类，相对的，K-Means之类的聚类算法一般只适用于凸数据集。

(2)可以在聚类的同时发现异常点，对数据集中的异常点不敏感。

(3)聚类结果没有偏倚，相对的，K-Means之类的聚类算法初始值对聚类结果有很大影响。

DBSSCAN缺点：

(1)如果样本集的密度不均匀、聚类间距差相差很大时，聚类质量较差，这时用 DBSCAN 聚类一般不适合。

(2)如果样本集较大时，聚类收敛时间较长，此时可以对搜索最近邻时建立的 KD 树或者球树进行规模限制来改进。

(3)调参相对于传统的 K-Means 之类的聚类算法稍复杂，主要需要对距离阈值 ε，邻域样本数阈值 $Minpts$ 联合调参，不同的参数组合对最后的聚类效果有较大影响[9]。

对特征矩阵进行可视化，结果如图 5：

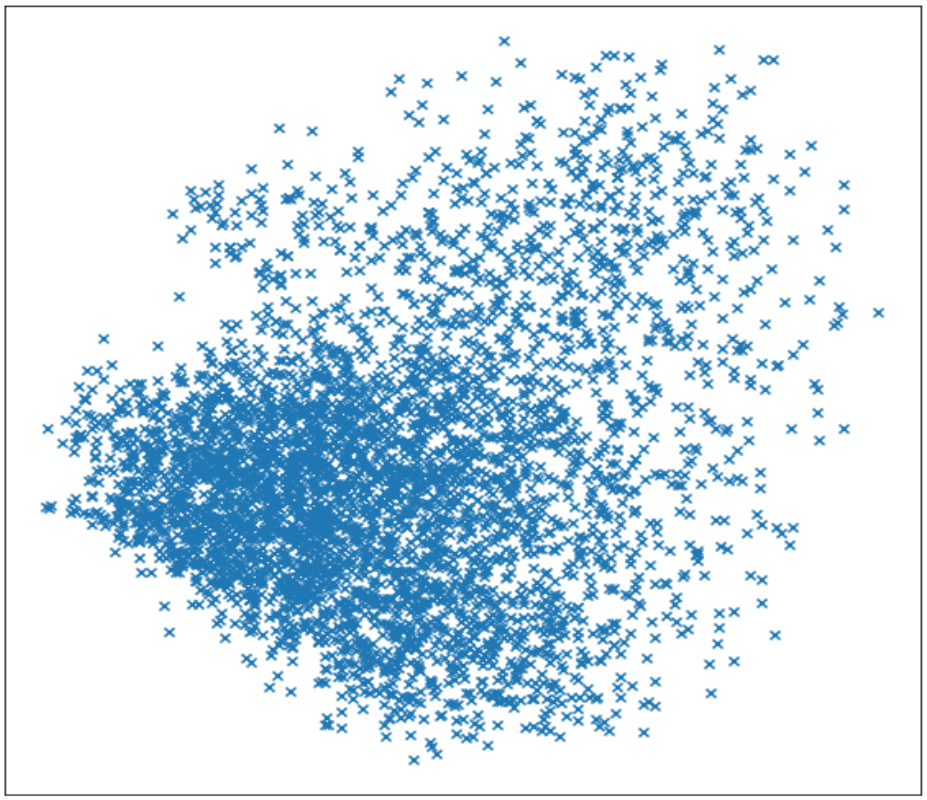

图 5

可以看出，数据密度分布十分不均匀，维度较高，有较多离群点且较多噪音和异常点，本文同时尝试了两种方法。

（一）在使用 K-Means 聚类时，要先确定 k 值，即聚类簇数。本文采用 Canopy 算法来确定 k 值。

Canopy 算法原理

Canopy 属于一种‘粗’聚类算法，即使用一种简单、快捷的距离计算方法将数据集分为若干可重叠的子集 canopy，这种算法不需要指定 k 值、但精度较低，可以结合 K-means 算法一起使用：先由 Canopy 算法进行粗聚类得到 k 个质心，再使用 K-means 算法进行聚类。

Canopy 算法步骤如下：

（1）将原始样本集随机排列成样本列表$L=\left[x_1,x_2,\mathrm{L}\ ,x_m\right]$（排列好后不再更改），根据先验知识或交叉验证调参设定初始距离阈值T_1、T_2，且$T_1>T_2$。

（2）从列表 L 中随机选取一个样本 P 作为第一个 canopy 的质心，并将 P 从列表中删除。

（3）从列表 L 中随机选取一个样本 Q，计算 Q 到所有质心的距离，考察其中最小的距离 D：

如果$\mathrm{D}\le T_1$，则给 Q 一个弱标记，表示 Q 属于该 canopy，并将 Q 加入其中；

如果$\mathrm{D}\le T_2$，则给 Q 一个强标记，表示 Q 属于该 canopy，且和质心非常接近，所以将该 canopy 的质心设为所有强标记样本的中心位置，并将 Q 从列表 L 中删除；

如果 $D>T_1$，则 Q 形成一个新聚簇，并将 Q 从列表 L 中删除。

（4）重复第三步直到列表 L 中元素个数为零[10]。

算法原理如图 6 所示。

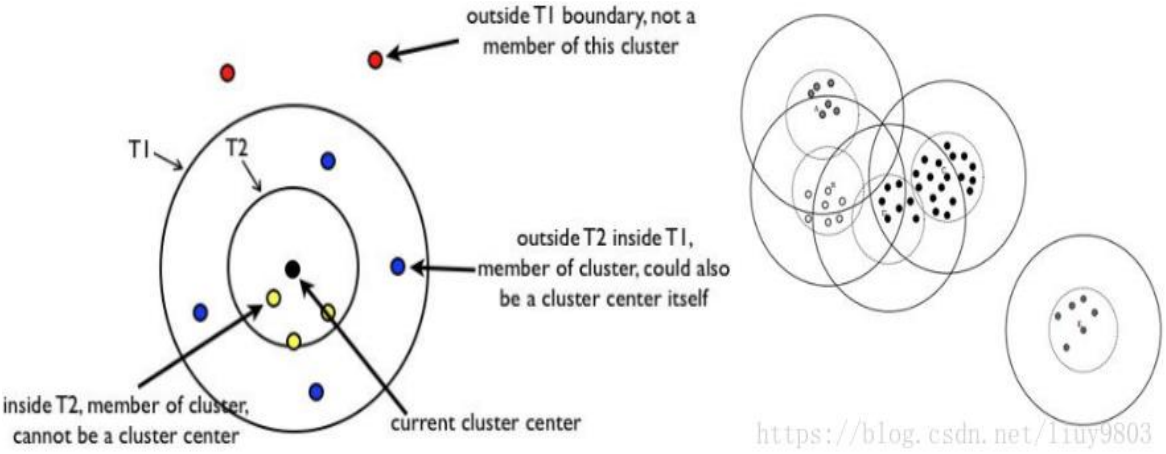

图 6

使用 Canopy 算法粗聚类后得到 3667 个初始中心，将聚类结果可视化得到图：

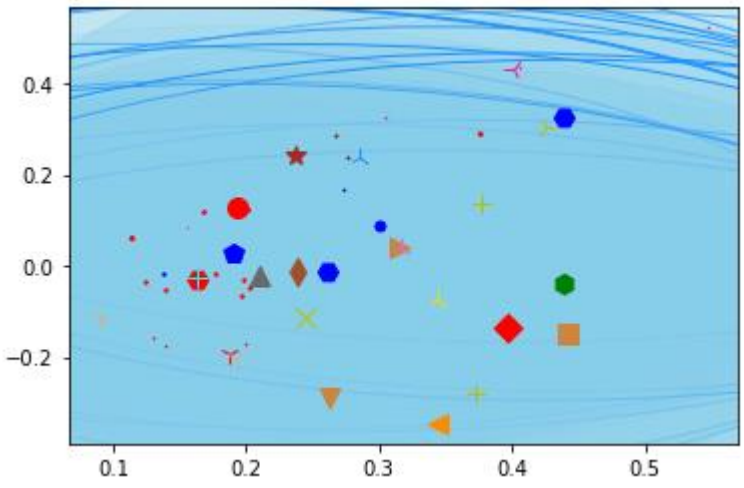

图 7

所以将 k 值定为 492。聚类后观察结果，十条以下的零散小簇较多，抽样查看其中几个标签后，发现聚类误差较大，结果并不满意。

（二）在使用 DBSCAN 聚类时，需要对 eps 和 min_samples 两个参数进行调整，这两个参数对聚类结果的影响巨大。本文使用轮廓系数（silhouette value）来确定这两个参数。

轮廓系数

silhouette 是一个衡量一个结点与它属聚类相较于其它聚类的相似程度。取值范围-1 到 1，值越大表明这个结点更匹配其属聚类而不与相邻的聚类匹配。如果大多数结点都有很高的 silhouette value，那么聚类适当。若许多点都有低或者负的值，说明分类过多或者过少。

轮廓系数结合了凝聚度和分离度，其计算步骤如下：

(1)对于第 i 个对象，计算它到所属簇中所有其他对象的平均距离，记为a_i（体现凝聚度）；

(2)对于第 i 个对象和不包含该对象的任意簇，记为b_i（体现分离度）；

(3)第 i 个对象的轮廓系数为$s_i = (b_i - a_i)/\max(a_i, b_i)$ 。

取 eps 从 0.2 到 0.6，间隔为 0.1，min_samples 从 1 到 5，间隔为 1，进行迭代聚类，结果如图 8 所示：

	eps	min_samples	score	raito	n_clusters
0	0.2	1	0.127185	0.000000	3911
1	0.2	2	-0.259504	0.837032	287
2	0.2	3	-0.282077	0.936662	73
3	0.2	4	-0.185257	0.971567	24
4	0.2	5	-0.067538	0.986593	8
5	0.3	1	0.133604	0.000000	3624
6	0.3	2	-0.200123	0.758900	351
7	0.3	3	-0.269979	0.870781	106
8	0.3	4	-0.203300	0.921174	41
9	0.3	5	-0.164183	0.937587	28
10	0.4	1	0.116371	0.000000	3100
11	0.4	2	-0.105954	0.623902	378
12	0.4	3	-0.170594	0.734628	140
13	0.4	4	-0.172572	0.786639	79
14	0.4	5	-0.164061	0.822700	50
15	0.5	1	0.016168	0.000000	2323
16	0.5	2	-0.087272	0.456542	340
17	0.5	3	-0.104160	0.556172	111
18	0.5	4	-0.083191	0.606796	69
19	0.5	5	-0.072160	0.635229	59
20	0.6	1	-0.204145	0.000000	1337
21	0.6	2	-0.213484	0.267453	195
22	0.6	3	-0.186223	0.318077	63
23	0.6	4	-0.133828	0.363615	40
24	0.6	5	-0.075306	0.391817	34

图 8

其中 score 为轮廓系数，raito 为噪声比，n_clusters 为分类簇数，由表可得，轮廓系数都十分低，说明分类不准确，噪声比大部分都很高,取轮廓系数较高,分类簇数较合理的 eps=0.4,min-samples=2 和 eps=0.5，min_samples=2 两组系数，查看聚类结果，发现偏差较大，结果也不让人满意。

采用上述两种聚类方法没有得到满意结果后，总结原因可能是因为数据密度不均匀导致。在对比了多种算法后，本题采用的是基于DBSCAN 算法的一种改进算法：HDBSCAN 算法。

HDBSCAN 算法的过程可以分为以下几步：1、空间变换；2、构建最小生成树；3、构建聚类层次结构(聚类树)；4、压缩聚类树；5、提取簇。相比于 DBSCAN 算法 HDBSCAN 算法主要做了以下几个优化：

（1）定义了一种衡量两个点互相间的距离的方式 mutual reachability distance，公式如下：

$$d_{mreach-k}(a,b)=\max\{core_k(a),core_k(b),dis\tan ce(1,b)\} \quad (17)$$

（2）使用最小生成树构建点与点之间的层次数模型，引入层次聚类思想，同时对最小生成树剪枝的最小子树做了限制，主要是为了控制生成的类簇不要过小。

（3）定义了一种叫 stability 的分裂度量方式如下：

1)定义每个点的密度度量为$\lambda=\frac{1}{\varepsilon}$，其中$\varepsilon$为该点与剩余聚类中点之间的最短距离。

2)定义一个族的生成密度λ_{birth}是这个簇生成时分裂边的导数

3)综上，一个簇的密度定义为：

$$\sigma=\sum_{p\in cluster}(\lambda_p-\lambda_{birth})。 \quad (18)$$

则 hdbscan 需要找到最大$\sum\sigma$的分裂簇方法，同时需要满足最小族类大小。

总结，HDBSCAN 相比于 DBSCAN 的最大优势在于不用选择人工选

择领域半径 R 和 MinPts，大部分的时候都只用选择最小生成类簇的大小即可，算法可以自动的推荐最优的簇类结果。同时定义了一种新的距离衡量方式，可以更好的与反映点的密度[11]，更加适合本题。

HDBSCAN 参数选择较为简单，本文仅对两个最关键参数 min_cluster_size 和 min_samples 进行调试。两个参数的意义如下：

(1)min_cluster_size：int，可选（默认 5）

集群的最小大小；包含少于此数量的点的单个链接拆分将被视为从群集中掉出的点，而不是被拆分为两个新群集的群集。

(2)min_samples:int,可选（默认 5）

一个点被视为核心点的邻域中的样本数。

经过重复调试，本文选用 min_cluster_size=5，min_samples=2，较能得出预期聚类效果。聚类结果如图 9 所示：

```
聚成几个类别：[ -1   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52
  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70
  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88
  89  90  91  92  93  94  95  96  97  98  99 100]
每个类别所拥有的的item数量：{-1: 2619, 66: 78, 49: 38, 0: 5, 97: 44, 42: 31, 59: 39, 70: 27, 34: 5, 35: 22, 46: 81, 20: 5, 60: 57, 72: 35,
11: 13, 28: 61, 22: 5, 26: 30, 30: 18, 88: 16, 91: 46, 45: 7, 58: 13, 80: 8, 92: 9, 54: 17, 16: 10, 14: 29, 29: 78, 27: 17, 94: 13, 51: 18,
90: 8, 1: 5, 6: 5, 33: 9, 17: 5, 37: 107, 9: 9, 38: 34, 48: 12, 64: 15, 89: 15, 53: 6, 63: 20, 50: 6, 40: 11, 47: 29, 86: 5, 78: 12, 13: 8,
43: 17, 75: 15, 55: 12, 61: 19, 73: 5, 67: 5, 21: 5, 85: 28, 74: 49, 96: 13, 25: 13, 81: 23, 4: 5, 98: 5, 100: 13, 62: 16, 71: 7, 84: 16, 3
1: 20, 5: 6, 8: 5, 12: 9, 10: 8, 82: 6, 15: 6, 23: 5, 65: 8, 39: 8, 41: 5, 83: 7, 57: 8, 93: 13, 77: 5, 24: 7, 7: 5, 68: 7, 79: 7, 76: 6, 9
9: 14, 44: 6, 95: 8, 19: 5, 87: 12, 52: 5, 2: 7, 3: 6, 18: 10, 69: 5, 32: 6, 56: 5, 36: 5}
```

图 9

共 100 个簇，有 2619 个离群点，在聚类出的簇中，簇的大小分布较为均匀，没有过大或过小的簇，随机查看后，发现聚类结果准确度较高，与事实较为相符，故选其为最终聚类结果。

热点问题是指在一定时间内集中反映的问题。留言条数的多少对热度起到决定性作用。利用聚类后的结果提取留言时间，利用最晚留言时间-最早留言时间得到时间差，以留言条数/时间差作为留言密度。利用点赞数＋反对数得到关注数，关注数越多，热度就越高。以留言非停用词数/非停用词数最大值代表信息充实度，留言的字数越多，其表达出的信息越充实，也越容易引发关注。

通过熵值法可以客观地得到以上四个评价指标所占总评分中的权重，可以避免主观确定权重的方法中，因人为因素而造成的误差。熵值法中主要依据“熵”的大小来确定各指标的权重，若该指标的熵值越小，则权重越大，因为熵小，说明该指标所提供的信息量大，在评价中越重要。熵值法反映出在各种评价指标值确定的情况下，各指标在竞争意义上的相对激烈程度，它能将一些边界不清、不易定量的因素定量化，进而实现综合评价[12]。其具体原理如下：

(1)建立原始评价矩阵

原始矩阵中的元素 x_{ij} 代表第 i 条留言，第 j 个指标的值，且有 4 个评价指标，若有 m 条留言样本，那么，建立的原始评价矩阵 X 为：

$$X=\begin{bmatrix} x_{11} & \Lambda & x_{14} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ x_{m1} & \Lambda & x_{m4} \end{bmatrix}, \qquad (19)$$

(2)指标的标准化，算各留言指标权重

由于各个指标的单位不一定相同，因此要先对其进行标准化处理，转为相对值。以此计算第 i 条留言的第 j 个指标的值，占第 j 个指标总值的比重 p_{ij}：

$$p_{ij}=\frac{x_{ij}}{\sum_{i=1}^{m}x_{ij}}, \tag{20}$$

(3)计算第 j 个评价指标的熵值

由于熵值系数为：

$$k=\frac{1}{\ln m}, \tag{21}$$

因此，第 j 个评价指标的熵值为：

$$H_j=-k\sum_{i=1}^{m}p_{ij}\ln p_{ij}, j=1,2,3,4 \quad , \tag{22}$$

(4)计算第 j 个评价指标的差异性系数

$$g_j=1-H_j, \tag{23}$$

(5)计算第 j 个评价指标权重

$$\omega_j=\frac{g_j}{4-\sum_{j=1}^{4}H_j}, j=1,2,3,4 \quad , \tag{24}$$

(6)综合评价值

第 j 个评价指标标准化后的值，与第 j 个评价指标的权重相乘后，将 4 个评价指标的加权评价值相加，即为综合评价值。

$$F=\sum_{j=1}^{4}x_j\cdot\omega_j \text{ 。} \qquad (25)$$

提取 TOP 热点问题的时间范围和地点人群载入我们自己提取的词典并使用 jieba 库对留言进行带词性分词，提取词性为 nr/ns 的词作为地点人群。

提取 TOP 热点问题的问题描述使用 TextRank 算法进行 TOP10 关键词提取后，人工组成一句话作为问题描述。这里需要用到 TextRank 算法，它是一种文本排序算法，由谷歌的网页重要性排序算法 PageRank 算法改进而来，它能够从一个给定的文本中提取出该文本的关键词、关键词组。

TextRank 算法的基本原理

PageRank 算法的基本概念和原理

(1)如果一个网页被很多其他网页链接到，说明这个网页比较重要，即该网页的 PR 值（PageRank 值）会相对较高；

(2)如果一个 PR 值很高的网页链接到一个其他网页，那么被链接到的网页的 PR 值会相应地因此而提高。

以投票机制的观点来看，一个网页的得票数由所有链向它的网页的得票数经过递归算法来得到，有到一个网页的超链接相当于对该网页投了一票。一个网页的 PR 值是由其他网页的 PR 值计算得到的。由于 $PR=A\times PR$（A 为概率转移矩阵）满足马尔科夫链的性质，那么通过迭代可以得到所有网页的 PR 值。经过重复计算，这些网页的 PR 值会趋于正常和稳定。

随着研究的深入，目前 PageRank 算法被广泛应用于众多方面，

例如学术论文的重要性排名、学术论文作者的重要性排序、网络爬虫、关键词与关键句的抽取等。

TextRank 算法是由 PageRank 算法改进而来的，二者的思想有相同之处，区别在于：PageRank 算法根据网页之间的链接关系构造网络，而 TextRank 算法根据词之间的共现关系构造网络；PageRank 算法构造的网络中的边是有向无权边，而 TextRank 算法构造的网络中的边是无向有权边。TextRank 算法的核心公式如下，其中用于表示两个节点之间的边连接具有不同的重要程度：

$$WS(V_i)=(1-d)+d\times\sum_{V_j\in In(V_i)}\frac{\omega_{ji}}{\sum_{V_{k\in Out}(V_j)}\omega_{jk}}WS(V_j)。\quad（26）$$

使用 TextRank 算法提取关键词和关键词组的具体步骤如下：

(1)将给定的文本按照整句进行分割，即；$T=[S_1,S_2,\Lambda,S_m]$；

(2)对于每个句子$S_i\in T$，对其进行分词和词性标注，然后剔除停用词，只保留指定词性的词，如名词、动词、形容词等，即$S_i=[t_{i,1},t_{i,2},\mathrm{L},t_{i,n}]$，其中$t_{i,j}$为句子$i$中保留下的词；

(3)构建词图$G=(V,E)$，其中 V 为节点集合，由以上步骤生成的词组成，然后采用共现关系构造任意两个节点之间的边：两个节点之间存在边仅当它们对应的词在长度为 K 的窗口中共现，K 表示窗口大小，即最多共现 K 个单词，一般 K 取 2；

(4)根据上面的公式，迭代计算各节点的权重，直至收敛；

(5)对节点的权重进行倒序排序，从中得到最重要的 t 个单词，作为 top-t 关键词；

(6)对于得到的 top-t 关键词，在原始文本中进行标记，若它们之间形成了相邻词组，则作为关键词组提取出来[13]。

5、答复意见评价

对答复意见以及留言主题留言详情处理后提取各指标并使用熵值法算出权重最后计算出得分。

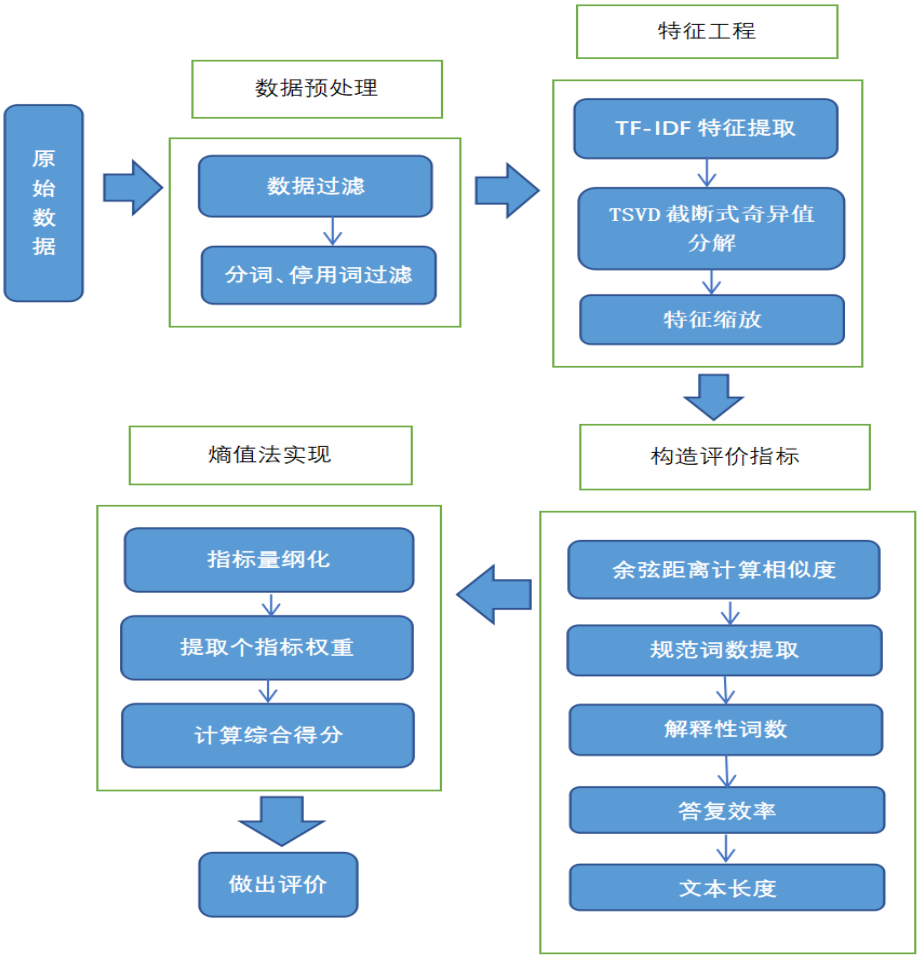

图 10

我们基于相关性、完整性、可解释性、答复效率、构造四项评价指标。分别对留言主题、留言详情、答复意见进行分词、去停用词，基于 TF-IDF 进行余弦相似度计算。

余弦相似度即，对于两条边，可以用夹角的余弦值来表示他

们的贴合程度。同样的，对于多维向量，也可用余弦相似度来计算二者之间的相似程度，计算方法为 ：

$$similarity=\cos(\theta)=\frac{A\cdot B}{\|A\|\|B\|}=\frac{\sum_{i=1}^{n}A_i xB_i}{\sqrt{\sum_{i=1}^{n}(A_i^2)}x\sqrt{\sum_{i=1}^{n}(B_i^2)}}, \quad (27)$$

得出各答复意见与留言内容和留言主题的文本相似度，提取均值作为相似性指标[14]。

对聚类进行词频统计，统计出使用词频较高的词，并人工筛选出答复意见格式用到的词，例如“答复如下”、“针对您反映”、“现将有关情况回复如下”等词，再对答复意见进行分词后从分词中提取这些词，并对词数量进行统计，作为规范词数。

答复意见的可解释性为提取答复意见中引用的法律法规以及政策的数量进行量化，引用法律法规以及政策一般都使用《》，所以我们利用正则表达式进行《(.*?)》提取，并将提取后的词数量进行统计，作为解释行词数。

答复意见的质量一方面也包括问题的答复是否及时，我们将数据中的答复时间与留言时间依照年/月/日的格式提取出来，转换为时间元组再转换为时间戳后通过该公式：

$$Time\_difference=\frac{t_2-t_1}{24\times 60\times 60}, \quad (28)$$

转换为单位为天的时间差。再根据时间差的天数按照 1 个星期内、一个月内、一个月以上分为三个答复效率等级对时间差进行转换。

采用有效字符数作文答复意见的文本长度，文本长度通常与

文本信息含量具有正向关系，一段话越长往往包含的信息也就越多[15]。

人们自身更倾向于认为，长度较短的评论相比于长度较长的评论有用程度更低。Agichtein 对雅虎问答中问题和回答的质量进行了研究，发现文本长度这一特征对于高质量回答的评判具有支配地位。后来学者在研究如何利用机器学习自动评价社区问答回复质量时，均将回复的长度作为一个重要的指标纳入模型之中。我国学者 李晨、巢文涵等以百度知道中 2005 年 6 月 12 日到 2005 年 7 月 12 日的 33637 个问题以及与之对应 145184 个答案为研究样本，同样采用机器学习的方法从文本特征和非文本特征两个维度对其中的答案质量进行评价，也发现最有助于答案判别的 10 大显著特征中，文本长度和平均句子长度的重要性更高。基于此，本文选取了答复意见的长度作为衡量答复意见的质量指标之一。因文中有标点符号以及无意义或影响不大的特殊符号，所以我们利用停用词表将其去除[15]。本文以答复意见中的有效词数作为回复长度的衡量，有效词数的具体获取步骤如下：

(1)将答复意见进行分词

本文选用的 python 常用的 jieba 分词库进行分词处理，其精准模式能够将句子按照最精细的模型进行分割，达到分词的目的。

(2)去停用词和标点符号

本文采用了较为常用的哈工大停用词表。

(3)回复质量指标的计算

具体公式为：

$$Len=\ln(1+words)。\qquad（29）$$

其中， $words$ 代表答复意见的有效词数。

通过如 4.7 的熵值法依据熵的大小，客观地得到以上五个评价指标所占总评分中的权重，其具体原理如下：

(1)建立原始评价矩阵

原始矩阵中的元素 x_{ij} 代表第 i 条留言，第 j 个指标的值，且有 5 个评价指标，若有 m 条留言样本，那么，建立的原始评价矩阵 X 为：

$$X=\begin{bmatrix} x_{11} & \Lambda & x_{15} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ x_{m1} & \Lambda & x_{m5} \end{bmatrix},\qquad（30）$$

(2)指标的标准化

由于各个指标的单位不一定相同，因此要先对其进行标准化处理，转为相对值。（$i=1,2,\Lambda\ ,m;\ j=1,2,3,4,5$）

正向指标：越大评价越好的指标，相关性、完整性、可解释性、文本长度均为正向指标，其标准化公式为：

$$x_{ij}=\frac{x_{ij}-\min\left(x_{1j},x_{2j},\mathrm{L}\ ,x_{mj}\right)}{\max\left(x_{1j},x_{2j},\mathrm{L}\ ,x_{mj}\right)-\min\left(x_{1j},x_{2j},\mathrm{L}\ ,x_{mj}\right)}+1,\qquad（31）$$

负向指标：越小评价越好的指标，答复效率为负向指标，其标准化公式为：

$$x_{ij}=\frac{\max\left(x_{1j},x_{2j},\mathrm{L}\ ,x_{mj}\right)-x_{ij}}{\max\left(x_{1j},x_{2j},\mathrm{L}\ ,x_{mj}\right)-\min\left(x_{1j},x_{2j},\mathrm{L}\ ,x_{mj}\right)}+1,\qquad（32）$$

(3)计算各留言指标权重

第 i 条留言的第 j 个指标的值，占第 j 个指标总值的比重 p_{ij}：

$$p_{ij}=\frac{x_{ij}}{\sum_{i=1}^{m}x_{ij}}, \quad (33)$$

(4)计算第 j 个评价指标的熵值

由于熵值系数为：

$$k=\frac{1}{\ln m}, \quad (34)$$

因此，第 j 个评价指标的熵值为：

$$H_j=-k\sum_{i=1}^{m}p_{ij}\ln p_{ij}, j=1,2,3,4,5 \quad , \quad (35)$$

(5)计算第 j 个评价指标的差异性系数

$$g_j=1-H_j, \quad (36)$$

(6)计算第 j 个评价指标权重

$$\omega_j=\frac{g_j}{5-\sum_{j=1}^{5}H_j}, j=1,2,3,4,5 \quad , \quad (37)$$

(7)综合评价值

第 j 个评价指标标准化后的值，与第 j 个评价指标的权重相乘后，将 5 个评价指标的加权评价值相加，即为综合评价值。

$$F=\sum_{j=1}^{5}x_j\cdot\omega_j \quad 。 \quad (38)$$

6、参考文献

[1] 李康康，龙华.基于词的关联特征的中文分词方法[J]. 通信技术，2018，51（10）：2343-2349.

[2] 隗中杰. 文本分类中 TF-IDF 权重计算方法改进[J]. 软件导报，2018，17（12）:39-42.

[3] 石凤贵.基于 TF-IDF 中文文本分类实现[J].现代计算机,2020(06):51-54+75.

[4] 田东风,欧飞,申维.矩阵奇异值分解理论在中文文本分类中的应用[J].数学的实践与认识,2008,38(24):132-140.

[5] 祁亨年. 支持向量机及其应用研究综述[J]. 计算机工程,2004,(10):6-9.doi:10.3969/j.issn.1000-3428.2004.10.003.

[6]https://scikit-learn.org/

[7]王彬宇,刘文芬,胡学先,魏江宏.基于余弦距离选取初始簇中心的文本聚类研究[J].《计算机工程与应用》,2018,54(10):11-18.

[8] 杨俊闯,赵超.K-Means 聚类算法研究综述[J].《计算机工程与应用》,2019,55(23):7-14,63.

[9]宋董飞,徐华.DBSCAN 算法研究及并行化实现[J].《计算机工程与应用》,2018,54(24):52-56,122.

[10]陈胜发,贾瑞玉.基于密度权重 Canopy 的改进 K-medoids 算法[J].《计算机工程与科学》,2019,41(10):1823-1828.

[11] Mark de Berg,Ade Gunawan,Marcel Roeloffzen.Faster DBSCAN and HDBSCAN in Low-Dimensional Euclidean Spaces[J].

《International Journal of Computational Geometry & Applications》,2019,29(01):21-47.

[12]罗来正,肖勇,王宝瑞,苏艳,朱玉琴,黎小锋,柏遇合.熵值法在涂层老化指标权重确定中的应用[J].装备环境工程,2017,14(07):70-73.

[13]李志强,潘苏含,戴娟,胡佳佳.一种改进的 TextRank 关键词提取算法[J].《计算机技术与发展》,2020,30(3):77-81.

[14]张根宇.基于 TF-IDF 和余弦相似度的文本相似度算法研究和优化策略[J].中国科技成果,2019,20(16):25-26.

[15]霍海疆.中国上市公司投资者互动平台回复质量研究[D].四川:西南财经大学,2019.