“智慧政务”中的文挖掘与综合分析

摘要

智慧政务即通过“互联网+政府服务”构建智慧型政府，利用人工智能以及数据挖掘等技术，简化群众办事环节、提升政府行政效能、畅通政府服务渠道，解决群众“办事难”等问题。但同时也带来了一些问题，例如：大数据背景下社情民意的反映和政府部门对此的反馈，大量的文本数据量使得原本的人工处理方法变得不再快捷有效。

对于问题 1，利用 Jieba 中文分词工具对附件 2 留言详情数据进行中文分词及删除停用词，利用 TF-IDF 算法对分词结果进行向量化，分别使用不同模型对训练数据集训练，利用运行时间，F1-Score 值和 ROC 曲线图，得到分类最优模型。

对于问题 2，针对附件三留言内容进行 Jieba 分词与 TF-IDF 向量化，分别利用 K-means 算法与 repeated bisection 算法对其聚类，比较二者聚类效果，将留言用户数量，时间跨度，反对数和点赞数作为指标量化热度评价指标，利用 Topsis 对其评价，给出热点问题排序结果，Pyhanlp 分词给出地点或人名，利用 TextRank 自动摘要给出问题描述。

对于问题 3，利用文本相似度量化相关性，关键词涵盖量化完整性，时间跨度量化时效性，同类问题答复意见时间间隔量化可解释性，利用 Topsis 与 AHP 评价方法对期分析，利用熵权法得到结果。

关键词：Jieba 分词　TF-IDF　K-Means 聚类　Pyhanlp　Topsis　AHP

Article Mining and Comprehensive Analysis in "Smart Government Affairs"

Abstract

Smart government is to build a smart government through "Internet + government services", using artificial intelligence and data mining technologies to simplify the masses 'work links, improve the government's administrative efficiency, smooth the government's service channels, and solve the masses' "difficult to handle" issues. But at the same time, it also brought some problems, such as: the reflection of social conditions and public opinion in the context of big data and the feedback of government departments. The large amount of text data makes the original manual processing method no longer fast and effective.

According to the first question, using Jieba Chinese word segmentation tool to perform Chinese word segmentation and delete stop words on the Appendix 2 message details data, using TF-IDF algorithm to vectorize the word segmentation results, using different models to train the training data set, and using the running time, F1-Score value and ROC curve graph to get the optimal model for classification.

To solve the second question, jieba word segmentation and TF-IDF vectorization are carried out on the content of the message in Annex 3, and they are clustered using the K-means algorithm and repeated bisection algorithm respectively, and the clustering effect of the two is compared. The number of message users, time span, and opposition Numbers and likes are used as indicators to quantify the heat evaluation index, use Topsis to evaluate it, give the ranking results of hot issues, pyhanlp segmentation to give the location or person name, and use TextRank to automatically summarize the problem description.

For the third question , using text similarity to measure relevance, keywords cover quantified completeness, time span quantification and timeliness, time interval quantifiable interpretability of responses to similar questions, using Topsis and AHP evaluation methods for period analysis, and using entropy weight method got the answer.

Keywords : Jieba participle ;TF-IDF; k-means clustering ;Pyhanlp ;Topsis ;AHP

目　录

1.挖掘目标 4

2.分析方法与过程 4

2.1 问 1 群众留言一级标签分类 5
2.1.1 流程图 5
2.1.2 数据预处理 5
2.1.3　一级标签分类模型 7
2.1.4 分类结果评价 10
2.2 问 2 热点问题挖掘 12
2.2.1 数据预处理 13
2.2.2 文本聚类 16
2.2.3 热点问题 17
2.3 问 3 答复意见评价 20
2.3.1 答复意见特征量化 21
2.3.2 评价计算方法 21

3. 结论 25

参考文献 26

1.挖掘目标

大数据时代背景下，加快推动智慧政务系统建设已经是社会发展的新趋势，对提升政府办事效率以及紧密联系政府与民众之间的关系都有极大助力作用。本文针对给出的群众留言记录以及相关部门对部分群众留言的答复意见。采用 Jieba 中文分词工具、线性支持向量机以及 K-Means 聚类等方法，来完成以下三个目标：

(1) 利用中文文本分词和词语向量化方法来对留言详情文本进行文本预处理，再运用不同模型建立关于留言内容的一级标签分类结果，最后对不同模型进行评价，确定最优的模型分类器。

(2) 对留言内容聚类，确定最优留言类别，用不同聚类方法对其聚类，利用不同指标给出热点问题热度指标，排序得到排名前五的热点问题。

(3) 针对答复意见，量化其相关性质，采用不同评价方法训练指标后进行熵权法评价，给出不同答复意见的评价结果。

2.分析方法与过程

总体流程图

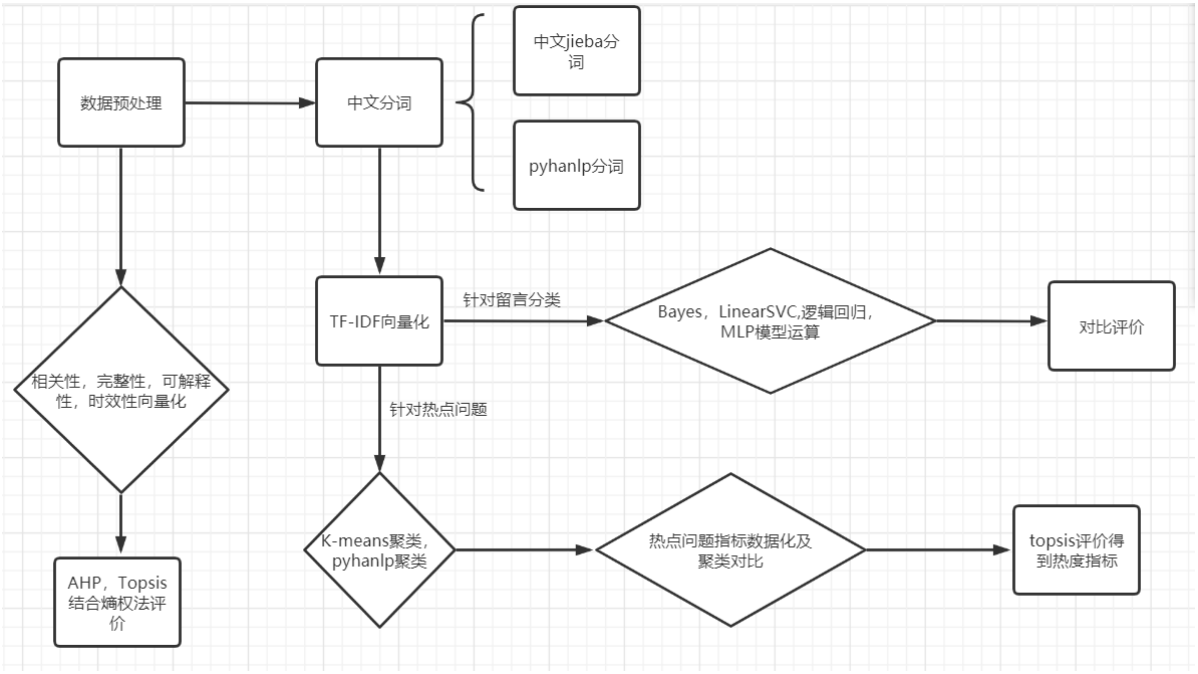

图 2-1 总体流程图

2.1 问 1 群众留言一级标签分类

2.1.1 流程图

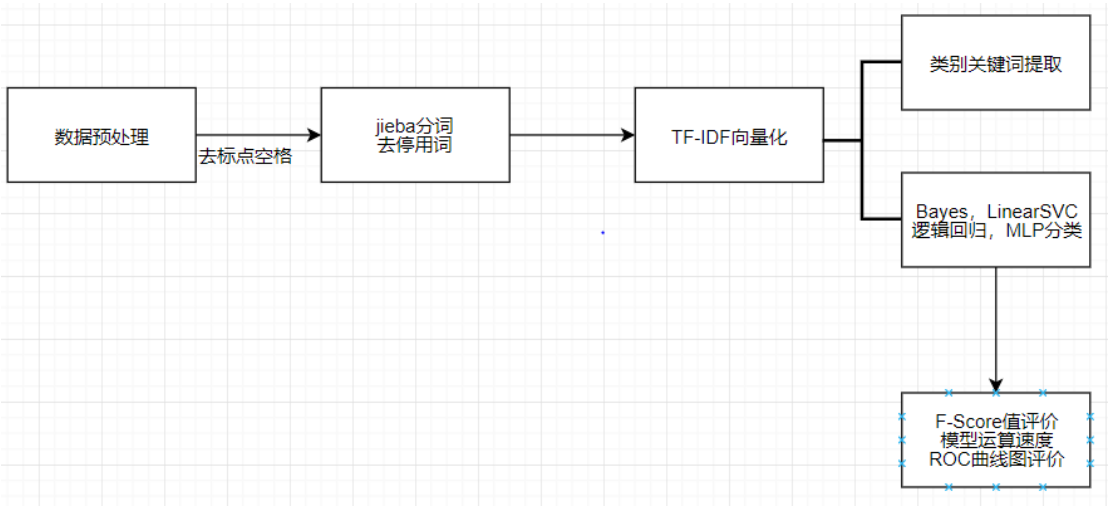

图 2-2 留言分类流程图

2.1.2 数据预处理

2.1.2.1 jieba 中文分词

中文分词是中文文本处理的基本步骤，也是中文人机自然语言交互的基础模块，在进行中文自然语言处理时，通常需要进行分析。Jieba 分词算法使用了高效的词图扫描，生成句子中汉字所有可能生成词情况所构成的有向无环图(DAG)，再采用动态规划查找最大概率路径，找出基于词频的最大切分组合。

Jieba 中文分词支持的三种分词模式包括[1]：

(1) 精确模式：试图将句子最精确地切开，适合文本分析；

(2) 全模式：把句子中所有的可以成词的词语都扫描出来，速度非常快，但是不能解决歧义问题；

(3) 搜索引擎模式：在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

针对附件 2 所给的留言数据，结合附件一三类标签分类，对附件 2 留言详情采用 jieba 分词全模式搜索，便利且快捷。在进行 Jieba 分词之前，首先将所有

数据一级标签与留言详情相对应，将一级标签用具体的数字代替，存在category_id_df（如图3-2所示）中，便于之后的处理计算。针对附件2中的所有留言内容，先考虑到留言内容中存在非中文字符，故先进行去标点、去空和去英文数字的操作。

一级标签	category_id
城乡建设	0
环境保护	1
交通运输	2
教育文体	3
劳动和社会保障	4
商贸旅游	5
卫生计生	6

图2-3 一级标签分类

在完成以上的基础上，调用Python的中文分词包Jieba来进行分词。为节省存储空间和提高搜索效率，对留言内容进行去停用词（出现频率高且对信息含量小的词），例如我、的、了、呢等，最终得到分词的结果并存储在data中。随后利用TF-IDF算法来提取关键词。

本文利用Jieba分词得到的留言一级分类的结果，保存在附件1。

2.1.2.2 TF-IDF 算法[2]

TF-IDF（term frequency-inverse document frequency）是一种用于咨询检索与咨询探勘的常用加权技术，其中TF为词频、IDF为逆向文档频率。其主要思想是：如果某个词或短语在一篇文档中出现的频率TF高，并且在其他文档中很少出现，则认为此词或短语具有很好的类别区分能力，适合用来分类；换言之，如果包含某个词的文档越少，IDF越大，这说明该词具有很好的类别分区能力。

假设某个词为ω，则

$$\mathrm{TF}_{\omega}=\frac{\text{在某一类中词}\omega\text{出现的次数}}{\text{该类中所有的词数目}} \qquad (1)$$

$$\mathrm{IDF}_{\omega}=\log\left(\frac{\text{词库中的文档总数}}{\text{包含词}\omega\text{的文档数}+1}\right) \qquad (2)$$

分母之所以加1，是为了避免分母为0。

某一特定文档内的高词频率，以及该词语在整个文档集合中的低文档频率，会产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要

的词语。

$$\mathrm{TF}-\mathrm{IDF}=\mathrm{TF}*\mathrm{IDF} \quad （3）$$

2.1.2.3 生成 TF-IDF 向量并绘制词云

具体步骤：

（1）使用 TF-IDF 算法将所有词语特征向量化；

（2）将不同类别特征词排序，选出每类前 100 位的关键词，组成词云，保存在附件 2；

如附件 2 所示。明显可以看到卫生计生的得分较高的关键词有：医院、医生、患者等；而旅游商贸的关键词有：旅游、游客、食品等；劳动和社会保障的关键词有：劳动、单位、工资等等；其他类别不再一一赘述。

2.1.3 一级标签分类模型

2.1.3.1 贝叶斯分类器原理[3]

贝叶斯分类器是已知的一种概率分类器，在信息检索中有着非常重要的作用。朴素贝叶斯分算法是基于贝叶斯定理的分类算法，它是机器学习一个普通但实用的算法。如果根据多个特征而非一个特征对数据进行分类时，可以假设这些特征相互独立，然后利用条件概率乘法法则得到每一个分类的概率，然后选择概最大的那个作为机器的判定。

假设某个文本有数据集$\{(x_1,y_1),(x_2,y_2),\dots,(x_m,y_m)\}$，其中m是样本的个数，每个数据集包含n个特征，即$X_i=(x_{i1},x_{i2},\dots,x_{in})$，类标记集合$y_1,y_2,\dots,y_k)$。设$p(y=y_i|X=x)$表示输入的$X$样本为$x$时，输出的y为$y_i$。假设现在给定一个新样本，要判断它的类别可以分别求其条件概率。哪一个值取到最大就属于那一类，即求解最大的后验概率$arg\,max\,p(y|x)$。

贝叶斯定理有：$p(y=y_i|x)=\frac{p(y_i)p(x|y_i)}{p(x)}$ (4)

由于在一般情况下朴素贝叶斯定理假设各个特之间是相互独立的，则上式可以改写为：$p(y=y_i|x)=\frac{p(y_i)p(x|y_i)}{p(x)}=\frac{p(y_i)\prod_{j=1}^{n}p(x|y_i)}{\prod_{j=1}^{n}p(x)}$ (5)

最终判别公式为：

$$y=arg\max_{y_i}p(y_i)p(x|y_i)=arg\max_{y_i}p(y_i)\prod\nolimits_{j=1}^{n}p(x|y_i) \quad (6)$$

其流程图 2-4 如下所示

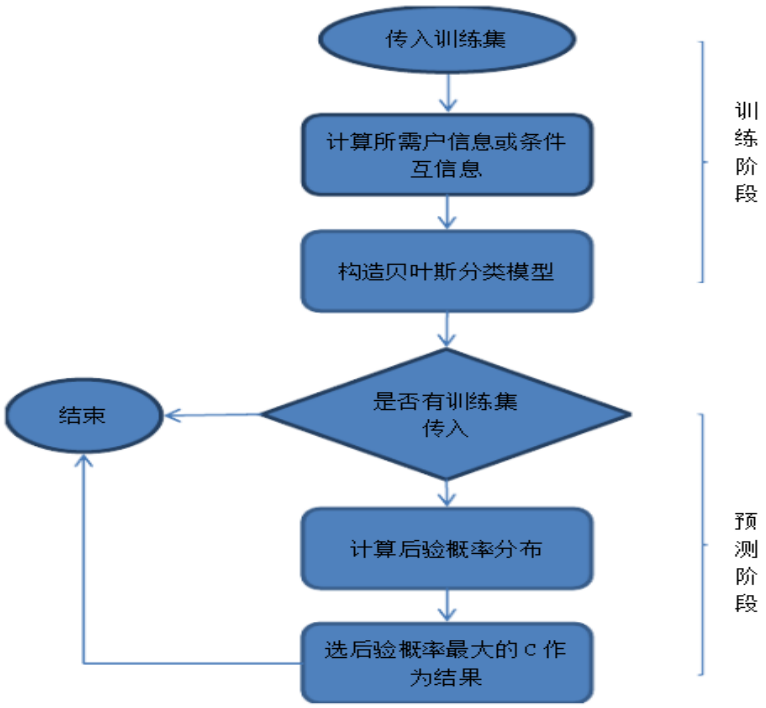

图 2-4 朴素贝叶斯分类器流程图

2.1.3.2 Linear SVM 原理[2]

支持向量机（support vector machine，SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；SVM 还包括核技巧，这使它称为实质上的非线性分类器。SVM 学习的基本思想上是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。

线性支持向量机处理的是线性不可分的数据集。对于线性支持向量机的优化问题，就是在线性可分支持向量机的基础上加了一个松弛变量ξ。

分类超平面：$\omega^* \cdot x + b^* = 0$ (7)

相应的决策函数为：$f(x) = sign(\omega^* \cdot x + b^*)$ (8)

优化问题：$min\frac{1}{2}\|\omega\|^2 + C\sum_i \xi_i$ (9)

$$y_i\left(\omega^t \phi(x_i)\right) \geq 1 - \xi_i \quad (10)$$

$$\xi_i \geq 0, \mathrm{i} = 1, \ldots, \mathrm{n} \quad (11)$$

2.1.3.3 逻辑回归分类器[4]

逻辑回归分类器是通过将训练数据的概率最大化，以产生正确的概率估计，基于正确的概率估计产生正确分类。它主要是以简单回归函数作为主要的学习器来拟合模型，然后再使用交叉验证来决定进行循环的次数。

逻辑回归模型的假设：

$$h_\theta(x) = g(\theta^T x) \quad (12)$$

其中x代表特征向量，g代表一个常用的逻辑函数，公式为：

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (13)$$

分类任务：

$$p(y = 1|x;\theta) = h_\theta(x) \quad (14)$$

$$p(y = 0|x;\theta) = 1 - h_\theta(x) \quad (15)$$

整合：

$$p(y|x;\theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y} \quad (16)$$

决策边界由假设函数中参数决定将样本正确分类的一条边界，在逻辑回归中预测：当$h_\theta(x)$大于等于 0.5 时，预测$y = 1$；当$h_\theta(x)$小于 0.5 时，预测$y = 0$。不同的θ决定了不同的假设函数，决策边界并不是一直都是明确的，特征空间中两类的过度是渐进的，其中若决策平面是超平面那么分类时线性可分的。因策在这里认为$\theta^T x = 0$为决策边界，当大于或者是小于 0 时，根据逻辑回归模型预测不同的分类结果。

2.1.3.4 多层感知机原理[5]

多层感知机是一种前馈网络，具有两个隐层的网络可以得到任何要求的判别边界以实现分类，输入层的节点数一般由输入的特征多少来决定。输出层节点数目选取有两种方法;一种是根据输出的个数来决定，另一种是将输出按二进制编码。如图 2-5 是多层感知机的神经网络图。

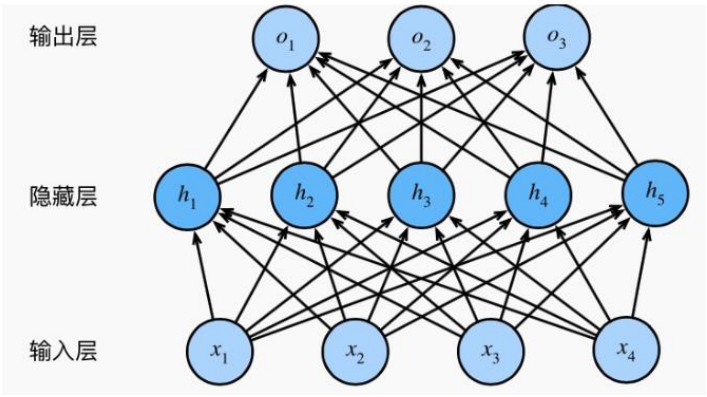

图 2-5 多层感知机网络图

给定一个小批量样本$\mathrm{X} \in \mathrm{R}^{n \times d}$, 其批量大小为$n$，输入个数为$d$。假设多层感知机只有一个隐含层，其中隐藏单元个数为h。假设隐含层的输出是H，有$\mathrm{H} \in \mathrm{R}^{n \times h}$。因为隐含层和输出层都是全连接，可以假设隐含层的权重参数和偏差

参数分别为$W_h \in \mathrm{R}^{d\times h}$和$b_h \in \mathrm{R}^{1\times h}$，输出层的权重和偏差参数分别为$W_o \in \mathrm{R}^{h\times q}$和$b_o \in \mathrm{R}^{1\times q}$。

$$H = \phi(XW_h + b_h) \quad (17)$$

$$O = HW_o + b_o \quad (18)$$

其中ϕ表示激活函数。在分类问题中，可以对输出O做 softmax 运算，且使用 softmax 回归中的交叉熵损失函数。

2.1.4 分类结果评价

将数据集划分为 4:1，分为训练数据集与测试数据集，利用上述模型分别在测试集上测试，针对运行时间，F1-Score 值以及 ROC 曲线给出模型评价。

2.1.4.1 运行时间

不同分类器在训练集上运行的时间（单位 ：秒）如下图 2-6 所示，可以看到贝叶斯分类器的运行时间最短而多层感知机的运行时间最长约为 15 秒，与其他方法相差明显。相对而言，贝叶斯和线性支持向量机分类器的运行时间较好，若在数据量庞大的实验中可以节约时间来完成更多的任务。

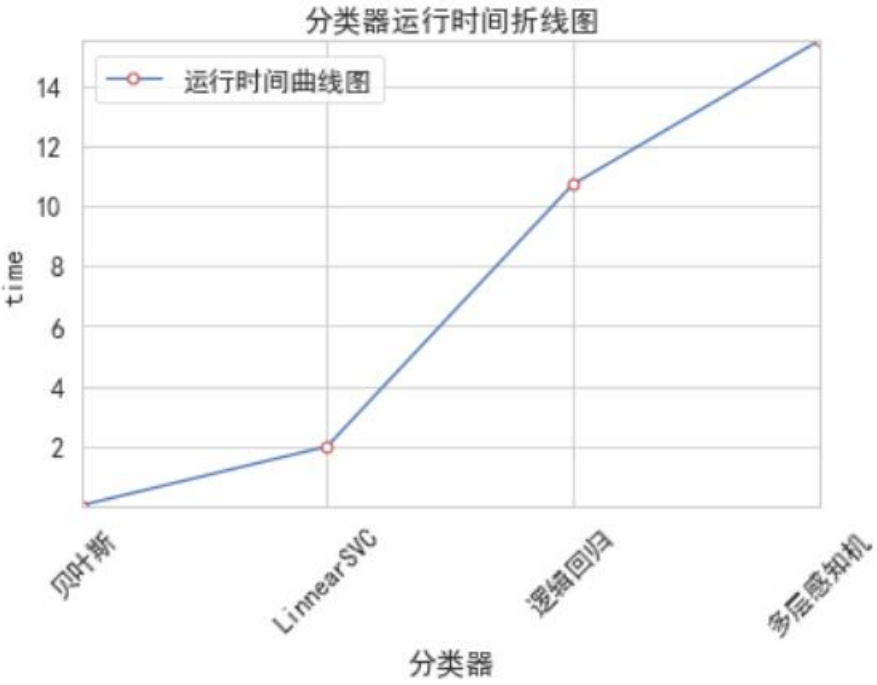

图 2-6 不同预测方法计算时间

2.1.4.2 F1-Score 值

F-Score 是一种用于评价分类模型分类好坏的方法，其中其中P_i为第i类的查准率，R_i为第i类的查全率。查准率是针对预测结果而言，计算的是所有“正确被检索的项（TP）”占所有“实际被检索到的项（TP+FP）”的比例。查全率是针对原样本而言，计算的是“正确被检索的项（TP）”占所有“应该检索到的项（TP+FN）”的比例。

从某一类别发角度考虑，F1 值就是查准率和查全率的调和平均：$\frac{2}{F1}=\frac{1}{P}+\frac{1}{R}$。而从所有类别考虑，能写成$F1=\frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$ (19) 的形式。

一般来说，查准率和查全率之间是矛盾的，引入 F-Score 作为综合指标，就是为了平衡准确率和查全率的影响，较为全面地评价一个分类器。

本文连续调用多个库模型，利用训练集数据来训练模型，并对测试集做一个预测。期间利用到了 Bayes 模型、线性支持向量机、逻辑回归模型和多层感知机分类器，并计算各个模型的 F-Score 值，最后得出线性支持向量机的 F-Score 值最大，值约为 0.9117。如下表 3-1 所示即为不同预测模型的 F-Score 值。而图 3-5 是不同模型的 F-Score 值的曲线图。

表 3-1 不同预测方法的 F-Score 值

预测方法	查准率	查全率	F-Score
Bayes 预测	0.8591	0.8588	0.8568
线性支持向量机	0.9119	0.9120	0.9117
逻辑回归	0.8871	0.8811	0.8801
多层感知机分类器	0.8325	0.8246	0.8271

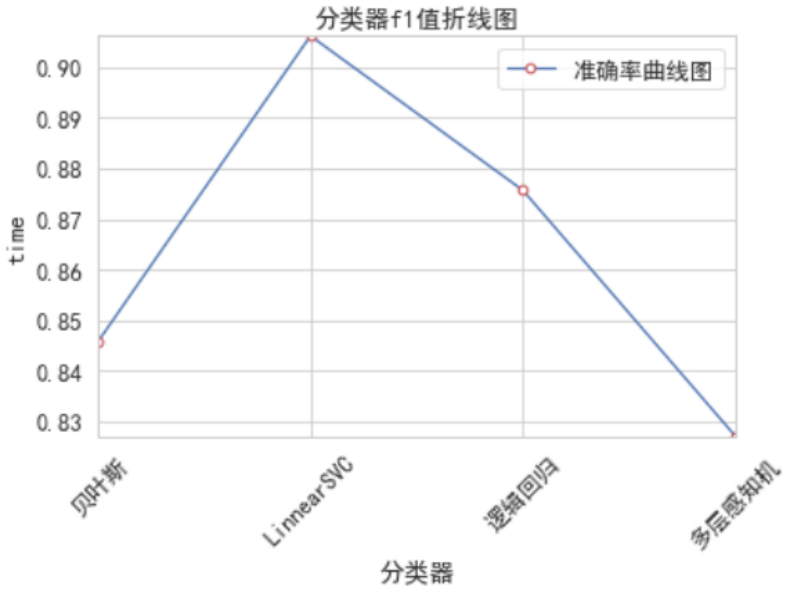

图 2-7 不同模型的 F-Score 值

2.1.4.3 绘制 ROC 曲线[6]

ROC 是一个用于度量分类中的非均衡性的工具，ROC 曲线及 AUC 常被用来评价一个二值分类器的优劣。曲线图中横坐标为 FPR（False positive rate 假阳率），纵坐标为真阳率 TPR（True postive rate）。其中，$TPR = TP/P$，$FPR = FP/N$，ROC 曲线越接近左上角，该分类器的性能越好即：

FPR：所有负例中有多少被预测为正例；

TPR：有多少真正的正例被预测出来

AUC（Area Under Curve） 被定义为 ROC 曲线下的面积，因为 ROC 曲线一般都处于 y=x 这条直线的上方，所以取值范围在 0.5 和 1 之间，使用 AUC 作为评价指标是因为 ROC 曲线在很多时候并不能清晰地说明哪个分类器的效果更好，而 AUC 作为一个数值，其值越大代表分类器效果更好。

下图为不同上述分类器的 ROC 曲线 AUC 图：

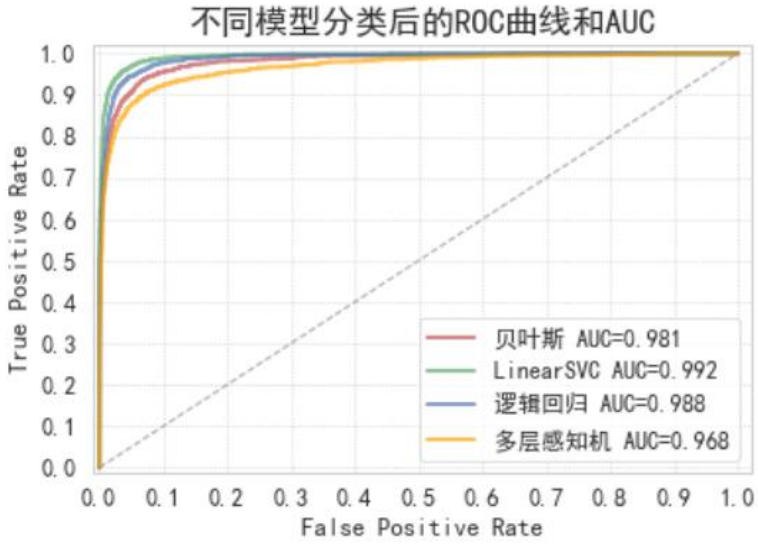

图 2-8ROC 曲线

综合上述三点评价可知，线性支持向量机的分类效果最好。

2.2 问 2 热点问题挖掘

思路流程

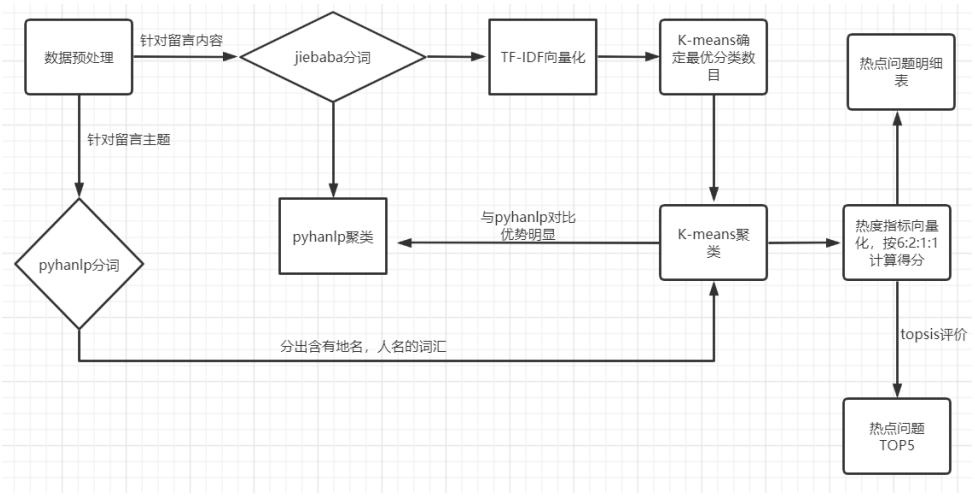

图 2-9 热点问题分析流程图

2.2.1 数据预处理

对留言内容采取与第一问相同的操作，即 jieba 分词和 TF-IDF 向量化，这里不展开赘述。

2.2.1.1 分词[7]

针对留言内容，本文仍采用 jieba 分词方式，但在热点问题地点或人群描述上，本文采用 pyhanlp 分词。

pyhanlp 获取 hanlp 中分词器有两种，第一种是直接从封装好的 hanlp 类中获取，这种获取方式一共可以获取五种分词器，而现在默认的就是第一种维特比分词器

1.维特比(viterbi)：效率和效果的最佳平衡。也是最短路分词，HanLP 最短路求解采用 Viterbi 算法。

2.双数组 trie 树 (dat)：极速词典分词，千万字符每秒（可能无法获取词性，此处取决于你的词典）。

3.条件随机场(crf)：分词、词性标注与命名实体识别精度都较高，适合要求较高的 NLP 任务。

4.感知机(perceptron)：分词、词性标注与命名实体识别，支持在线学习。

5.N 最短路 (nshort)：命名实体识别稍微好一些，牺牲了速度。

第二种方式是使用 JClass 直接获取 java 类，然后使用。这种方式除了获取上面的五种分词器以外还可以获得一些其他分词器，如 NLP 分词器，索引分词，快速词典分词等等，本文采用第二种分词器，使用 segment（）对留言主题分类，便于提取地点人名。

在留言主题处理上，自定义函数提取词语的词性，统计每个热点问题的留言主题分词后词性为 n，nx，ns，s 的词语从中找到出现频率最高的词作为该热点问题类别的地点或人群描述。

2.2.1.2 TSNE 降维[8]

由于直接用 TF_IDF 向量化后的数据代入 K-means 聚类器中计算速度非常慢，因此采用 TSNE 降维方式。

TSNE 是由 T 和 SNE 组成，也就是 T 分布和随机邻近嵌入，它是一种用于探索高位数据的非线性降维算法。它将多维数据映射到适合于人类观察的两个或多个维度。TSNE 主要步骤包括两个：第一，TSNE 构建一个高维对象之间的概率分布，是的相似的对象有更高的概率被选择，而不相似的对象有较低的概率被选择。第二，TSNE 在低维空间里在构建这些点的概率分布，使得这两个概率分布之间尽可能的相似。

高位数据用x表示，x_i表示第i个样本，低维数据用$\mathcal{Y}$来表示，则高维中的分布概率矩阵 P 定义如下：

$$p_{j|i} = \frac{\exp(-\left\|x_i - x_j\right\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\left\|x_i - x_j\right\|^2/2\sigma_i^2)} \quad (20)$$

低维中的分布概率矩阵计算如下：

$$q_{j|i} = \frac{\exp(-\left\|y_i - y_j\right\|^2)}{\sum_{k\neq i}\exp(-\left\|x_i - x_j\right\|^2)} \quad (21)$$

随机给定一个初始化的$\mathcal{Y}$，进行优化，使得$\mathcal{Y}$的分布矩阵逼近x的分布矩阵。给定目标函数，用KL散度来定义两个不同分布之间的差距：

$$\mathrm{C} = \sum_i KL\left(\mathrm{P}_i||\mathrm{Q}_i\right) = \sum_i\sum_j p_{j|i} log\frac{p_{j|i}}{q_{j|i}} \quad (22)$$

则可以计算梯度为：

$$\frac{\partial C}{\partial y_i} = 2\sum_j\left(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j}\right)\left(y_i - y_j\right) \quad (23)$$

TSNE 对高维中的分布采用对称 SNE 的做法，低维中的分布则采用更一般的 T 分布，也是对称的，具体算法流程如下：

Algorithm1：Simple version of t-Distributed Stochastic Neighbor Embedding

Data: data set $\mathrm{x} = \{x_1, x_2, \ldots, x_n\}$
cost function parameters: perplexity Perp
optimization parameters: number of iterations T, learning rate η ,momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$
begin

compute pairwise af nities $P_{j|i}$ with perplexities Perp (using Equation 9)

set $P_{ij} = \frac{P_{j|i} + P_{i|j}}{2n}$

sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$

for $t = 1$ **to** T **do**

compute low-dimensional af nities q_{ij} (using Equation 10)

compute gradient $\frac{\partial C}{\partial y}$ (using Equation 12)

set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\partial C}{\partial y} + \alpha(t)(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$

end

end

如图 2-10，降维前词语向量的位=维数为：41244，降维后的维数为 2，明显得到降低。

```
#数据降维前
print("降维前{}".format(X.shape))
tsne = TSNE(n_components=2)
X = tsne.fit_transform(X)
```

降维前(4326, 41244)

```
#降维后
print("降维后{}".format(X.shape))
```

降维后(4326, 2)

图 2-10 数据降维

2.2.2 文本聚类

2.2.2.1 K-means 聚类[9]

K-Means 聚类算法属于无监督机器学习，是典型的基于原型的目标函数聚类方法的代表，它以数据点到原型的某种距离作为优化的目标函数，利用函数求极值的方法得到迭代运算的调整规则，其采用距离作为相似性评价的指标，即认为两个对象的距离越近，其相似度就越大。

K-Means 算法流程如下：

1、 首先确定一个 K 值，即希望将数据集经过聚类得到 K 个集合；
2、 从数据集中随机选择 K 个数据点作为质心；
3、 对数据集中每个点，计算其余每一个质心的距离（如欧氏距离），离哪个质心近，就划分到哪个质心所属的集合；
4、 把所有数据归好集合后，一共有 K 个集合，然后重新计算每个集合的质心。
5、 如果新计算出来的质心和原来的质心之间的距离小于某一个设置的阈值（表示重新计算的质心的位置变化不大，趋于稳定，或者说收敛），可以认为聚类已经达到期望的结果，算法终止。
6、 如果新质心和原质心距离变化很大，需要迭代 3~5 步。

K-Means 数学原理：

假设原始数据集合为$(x_1, x_2, \dots, x_n)$，并且每个人x_i为d维的向量，K-Means 聚类算法要把这n个数据对象划分到k个簇中$(\mathrm{k} \le \mathrm{n})$，即找到使得下式满足的$S_i$：

$$\underset{S}{\operatorname{argmax}} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - u_i\|^2 \qquad (24)$$

其中u_i是S_i中所有点的平均值。

$$u_j = \frac{1}{|c_j|} \sum_{i \in c_j} x_i \qquad (25)$$

本文通过 K-Means 聚类把留言内容进行分类，由于不知道聚成多少类合适，因此从 2-2500 类分别计算得分。由于类别众多，在聚类前进行了 TSNE 数据降维，最后发现 1561 类的得分最高，故最后选择 1561 类对数据聚类，得分数值见下图 2-11：

	共分了几类	分数	最大类包含的个数	聚类的名称
61	1561	0.556678	9	168
41	1541	0.556639	10	158
29	1529	0.555883	9	777
102	1602	0.555350	9	605
1	1501	0.555192	9	255
...	...	...	...	...
490	1990	0.533339	8	23
496	1996	0.532962	7	679
498	1998	0.532654	7	161
487	1987	0.532049	7	129

图 2-11K-Means 聚类图

2.2.2.2 pyhanlp 文本聚类[10]

本文采用的是 HanLP 中的 repeated bisection 算法，该聚类算法的优势在于其聚类模块可以接受任意文本作为文档，而不需要用特殊分隔符隔开单词。另外，该模块还接受单词列表作为输入，用户可以将英文、日文等预先切分为单词列表后输入本模块。统计方法适用于所有语种，不必拘泥于中文。

在 K-means 中通常聚类个数 K 这个超参数难以准确指定，然而在 repeated bisection 算法中，有一种变通的方法，那就是通过给准则函数的增幅设定阈值 beta 来自动判断 k。此时算法的停机条件为，当一个簇的二分增幅小于 beta 时不再对该簇进行划分，即认为这个簇已经达到最终状态，不可再分；当所有簇都不可再分时，算法终止，此时产生的聚类数量就不再需要人工指定了。

2.2.2.3 聚类对比

通过对留言内容的文本聚类，K-means 算法共聚成 1561 类，而 repeated bisection 算法聚成 671 类，在聚类性能上前者要优于后者，但在运行速度以及文本预处理上，后者运行速度快得多切处理方式要比前者简单。

最后热点问题挖掘采用 K-means 聚类结果。

2.2.3 热点问题

2.2.3.1 热度评价指标

该问关键要定义合理的热点问题评价指标，本文采用对留言用户数量，留言时间跨度以及点赞数和反对数处理的方式，结合 Topsis 评价方法给出热点问题评价指标。

Topsis[11]

Topsis 法亦称为理想解法，是一种有效的多指标评价方法。这种方法通过构造评价问题的正理想解和负理想解，即各指标的最优解和最劣解，通过计算每个方案到理想方案的相对贴近度，即靠近正理想解和远离负理想解的程度，来对方案进行排序，从而选出最优方案。

Topsis 方法和原理：

设多属性决策方案集为D $= \{d_1, d_2, \dots, d_m\}$，衡量方案优劣的属性变量为，$x_1, x_2, \dots, x_n$，这时方案集 D 中的每个方案$d_i(i = 1, \dots, m)$的n个属性值构成的向量是$[a_{i1}, \dots, a_{in}]$，它作为n维空间中的一个点，能唯一地表征方案$d_i$。

正理想解C^*是一个方案集 D 中并不存在的虚拟的最佳方案，它的每个属性值都是决策矩阵中该属性的最好值；而负理想解C^0则是虚拟的最差方案，它的每个属性值都是决策矩阵中该属性的最差值。在n维空间中，将方案集 D 中的各备选方案d_i与正理想解C^*和负理想解C^0的距离进行比较，既靠近正理想解又远离负理想解的方案集 D 中的最佳方案；并可以据此排定方案集 D 中各备选方案的优先序。

Topsis 法所用的是欧式距离。至于既用正理想解又用负理想解是因为在仅仅使用正理想解时有时会出现某两个备选方案与正理想解的距离相同的情况，为了区分这两个方案的优劣，引入负理想解并计算这两个方案与负理想解的距离，与正理想解的距离相同的方案离负理想解远者为优。

Topsis 法的具体算法如下：

（1） 用向量规划的方法求得规范决策矩阵

设多属性决策问题的决策矩阵A $=$（a_{ij}）$_{m\times n}$，规范化决策矩阵

B $=$（b_{ij}）$_{m\times n}$，

其中$b_{ij} = a_{ij} \Big/ \sqrt{\sum_{i=1}^{m} a_{ij}^2}$，i $= 1,2, \dots,$ m，j $= 1,2, \dots,$ m　　（26）。

（2） 构成加权规范阵C $=$（c_{ij}）$_{m\times n}$

设由决策人给定各属性的权重向量为w $= [w_1, w_2, \dots, w_n]^T$，则

$c_{ij} = w_j \cdot b_{ij}$，$i = 1,2, \dots, m$，$j = 1,2, \dots, n$　　(27)

（3） 确定正理想解C^*和负理想解C^0

设正理想解C^*的第j个属性值为c_j^*,负理想解C^0第j个属性值c_j^0，则

$$\text{正理想解}c_j^* = \begin{cases}\max_i c_{ij}\text{，j 为效益型属性，}\\ \min_i c_{ij}\text{，j 为成本型属性，}\end{cases} \quad j = 1,2,\dots,n \quad (28)$$

$$\text{负理想解}c_j^0 = \begin{cases}\min_i c_{ij}\text{，j 为效益型属性，}\\ \max_i c_{ij}\text{，j 为成本型属性，}\end{cases} \quad j = 1,2,\dots,n \quad (29)$$

（4） 计算各方案到正理想解与负理想解的距离

备选方案d_i到正理想解的距离为

$$s_i^* = \sqrt{\sum_{j=1}^{n}(c_{ij} - c_j^*)^2}\text{，}\ i = 1,2\dots,m \quad (30)$$

备选方案d_i到负理想解的距离为

$$s_i^0 = \sqrt{\sum_{j=1}^{n}(c_{ij} - c_j^0)^2}\text{，}\ i = 1,2\dots,m \quad (31)$$

（5） 计算各方案的排队指标值（即综合评阶指数）

$$f_i^* = s_i^0/(s_i^0 + s_i^*)\text{，}\ i = 1,2,\dots,m \quad (32)$$

（6） 按f_i^*由大到小排列方案的优劣次序。

本文按上述方法对不同热点问题计算最终得分，发现得分受点赞数的影响较大，因此为了削弱点赞数的影响，将用户留言数量、时间跨度、点赞数、反对数按照 6:2:1:1 比例重新计算得分，得到得分 2.0，具体数值见下图：

	问题ID	留言用户数量	留言时间跨度	反对数	点赞数	得分	得分2.0
0	119	0.064088	0.053416	0.024978	0.813808	0.988247	0.131948
4	1133	0.054933	0.142127	0.019794	0.000512	0.033145	0.072135
1	1240	0.027466	0.050379	0.025449	0.404599	0.484438	0.071852
7	536	0.073244	0.054790	0.025449	0.001536	0.011492	0.055757
11	398	0.064088	0.052180	0.025449	0.001536	0.009476	0.050397
...	...	...	...	...	...	...	...
1545	1116	0.009155	0.000000	0.024978	0.000000	0.000908	0.007075
1544	869	0.009155	0.000000	0.024978	0.000000	0.000908	0.007075
1543	1399	0.009155	0.000000	0.024978	0.000000	0.000908	0.007075
1558	1383	0.009155	0.000000	0.021207	0.002049	0.000664	0.006903
1560	75	0.009155	0.000007	0.021207	0.000000	0.000655	0.006700

1561 rows × 7 columns

图 2-12 热点问题得分图

注：1. 留言用户数量，留言时间跨度，点赞数与反对数为 topsis 处理后的结果

2. 该表按得分 2.0 排序，可以看到得分高低对应的问题 ID

将得分 2.0 作为热度评价指标对热点问题排序，即可得到热点问题明细表。

2.2.3.2 热点问题地点或人名提取与问题描述

2.2.3.2.1 地点或人名提取

利用 pyhanlp 提取热点问题对应留言主题的地点或人名词，前文在介绍 pyhanlp 分词时已做过解释，这里不展开赘述。

2.2.3.2.2 问题描述

TextRank 自动摘要原理[12]

TextRank 是一种基于图的无监督方法，通过对图结构的迭代计算实现词语的重要性然后进行文本摘要和关键词生成。其基本原理如下：

设$G(V,E)$是由给定文本的词汇所构成的图结构，V为图节点集合，E为图边集合。对于文本中的任意V_i，基于 TextRank 算法得到的权重W_i计算公式为：

$$W_i = (1-d) + d \times \sum_{V_j \in ln(V_i)} \frac{w_{ji}}{\sum_{V_k \in out(V_j)} w_{jk}} W_j \qquad (33)$$

式中 d 为阻尼系数，取值范围是 0-1；$ln(V_i)$为指向节点V_i的所有节点的集合；$out(V_j)$为节V_j指向所有的节点的集合；w_{ji}为节点V_j到V_i的边的权重。

对热点问题留言主题采用 TextRank 自动摘要，得到问题描述。取出排名前五的热点问题及其描述，得到热点问题图表。

	热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述
0	1	1	0.12465499	2019/01/20至2019/11/06	A市A4区	58车贷案件进展
1	2	2	0.080563483	2017/06/08至2019/10/15	A市经济学院学生	强制学生外出实习
2	3	3	0.065450132	2019/01/08至2019/12/29	A市A6区A7县中学	强制学生有偿补课
3	4	4	0.056519968	2019/08/23至2019/09/06	渝长厦高铁A4区绿地海外滩二期业主	渝长厦高铁路线对小区噪音影响严重
4	5	5	0.056328473	2019/02/10至2019/11/04	西地省A市雨敞坪镇雨敞坪快递驿站	快递驿站乱收费

图 2-13 部分热点问题

2.3 问 3 答复意见评价

思路流程

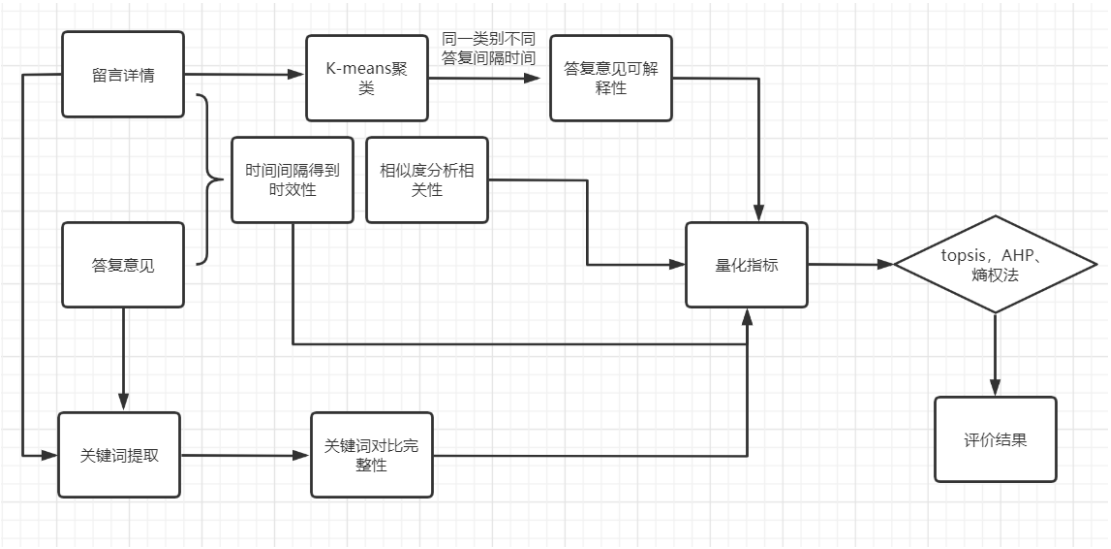

图 2-14 答复意见评价流程图

2.3.1 答复意见特征量化

1. **相关性：**利用附件 4 留言详情与答复意见的文本相似度量化该特征，相似度计算我采用的是 text2vec 库中的 Similarity 函数，可直接对两个文本计算相似度，较为准确且方便；

2. 完整性：通过对留言详情以及答复意见的关键词提取，提取方法为 Jieba 分词 TF-IDF 向量化排序，选出前 30 个关键词，对比答复意见与留言详情的关键词，计算答复意见涵盖留言详情关键词的比率作为答复意见完整性；

3. 时效性：通过留言与答复的时间间隔处理，计算留言详情与答复意见的时间差，作为时效性；

4. **可解释性：**对于不同答复意见，我们认为利用同类问题下的不同答复意见时间间隔可作为衡量指标，因此采用 K-means 聚类算法对留言详情聚类，得到相同问题下的不同时间的答复意见，将留言时间作为标准，对相同问题下的不同答复意见做时间排序，将相邻两个答复意见的时间间隔作为后者问题的可解释性指标，在通过总类别数量化统一，即：

$$x = \frac{L}{N}\left(\frac{time_i - time_{i-1}}{time_{end} - time_0}\right) \qquad (34)$$

其中L为某类问题答复意见数，N为总答复意见数，$\text{time}_{end} - \text{time}_0$为答复意见的最大时间差。

2.3.2 评价计算方法

2.3.2.1 AHP（层次分析法）[13]

层次分析法(Analytic Hierarchy Process,AHP)这是一种定性和定量相结合的、系统的、层次化的分析方法。这种方法的特点就是在对复杂决策问题的本质、影响因素及其内在关系等进行深入研究的基础上，利用较少的定量信息使决策的思维过程数学化，从而为多目标、多准则或无结构特性的复杂决策问题提供简便的决策方法。**是对难以完全定量的复杂系统做出决策的模型和方法。**
由于我们的数据以文本处理为主，其量化存在不容易被察觉的误差，所以初步采用 AHP 方法进行分析。

（1）建立层次结构模型

将决策的目标、考虑的因素(决策准则)和决策对象按他们之间的相互关系分成最高层、中间层和最低层，绘制层次结构图。

. 最高层(目标层)：决策的目的、要解决的问题；
. 中间层(准则层或指标层)：考虑的因素、决策的准则；
. 最低层(方案层)：决策时的备选方案；

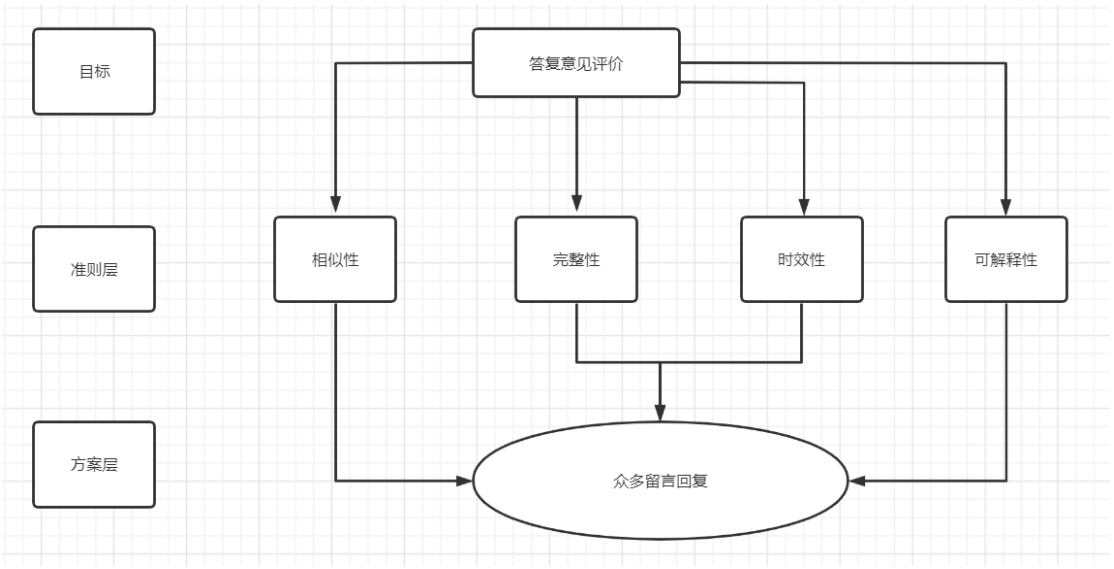

图 2-15 层次结构模型流程图

在这里，我们的准则层既是相关性，完整性，可解释性和时效性四者。每一条留言的回复则是方案层，而我们的目标就是给留言回复的质量进行排序，作为评估的一大因素。

（2） 构造判断(成对比较)矩阵

这里我们的判断矩阵使用一致矩阵法，早时由 santy 等人提出。

（3）一致矩阵法

（1） 不把所有因素放在一起比较，而是两两比较；

（2）对此时采用相对尺度，以尽可能减少性质不同的诸因素相互比较的困难，以提高准去性；

成对比较矩阵是表示**本层所有因素针对上一层某一个*因素(准侧或目标)*的相对重要性**的比较。成对比较矩阵的元素a_{ij}表示的是第i个因素相对于第j个因素的比较结果，这个值使用的是 Santy 的 1-9 标度方法给出,如下表。

标度	含义
1	表示两个因素相比，具有同样重要性
3	表示两个因素相比，一个因素比另一个因素稍微重要
5	表示两个因素相比，一个因素比另一个因素明显重要
7	表示两个因素相比，一个因素比另一个因素强烈重要
9	表示两个因素相比，一个因素比另一个因素极端重要
2,4,6,8	上述两个相邻判断的中值
倒数	因素i与j比较的判断a_{ij}，则因素j与i比较的判断$a_{ji} = 1/a_{ij}$

（4）层次单排序及一致性检验

两两比较之后，为了得出下层各因素对上层某因素的影响程度的排序结果，我们需要对其进行层次单排序和一致性检验

其元素为同一层次因素对于上一层次因素某因素相对重要性的排序权值，这一过程称为层次单排序，之后需要进行一致性检验来确认层次单排序。

注：一致阵的性质

1. $a_{ij} = 1/a_{ji}$，$a_{ii} = 1(i,j = 1,2,\ldots,n)$;
2. A^T也是一致阵；
3. A的各行成比例，则A的矩阵转秩为 1；
4. A的最大特征根（值）为$\lambda = n$,其余$n-1$个特征根均为 0；
5. A的任一行（列）都是对应于特征根n的特征向量，$AW = nW$。

（1）如果成对比较矩阵是一致阵，则我们自然回取对应于最大特征根 n 的归一化特征向量$\{w_1, w_2, \ldots, w_n\}$，且$\sum_{i=0}^{n} w_i = 1$，$w_i$表示下层第i个因素对上层某个因素影响程度的权值。

（2）若成对比较矩阵不是一致阵，我们用其最大特征根对应的归一化特征向量作为权向量 W，即特征根法。

（3）**定义一致性指标：**

$$\mathrm{CI} = \frac{\lambda - n}{n-1} \quad (35)$$

若$\mathrm{CI}=0$，则有完全的一致性，CI越接近 0，一致性越强，反之，不一致性越强。为了衡量CI 的大小，引入随机一致性指标 RI。

（4）**定义一致性比率：**$\mathrm{CR}=\frac{CI}{RI}$ (36)

若$\mathrm{CR}<0.1$，则一致性检验通过，之后按照总排序权向量进行决策。

（5）AHP 的优势和缺点[13]

优势：1.可以对难以完全定量的事物进行有效的评估并作出决策；
2.分层确定权重，相比而言，减少了传统的主观判断所带来的误差；
3.纵向横向均可比较，为以后改进提供依据。

缺点：1.判断矩阵的构造方式受主观影响较大；
2.九级指标进行两两判断，容易产生误差与矛盾。

Topsis 在前文中已经给过介绍，故这里不再展开赘述。

2.3.2.2 熵权法[14]

在前面的基础上，我们已经有了 AHP 和 TOPSIS 得出的数据。之后，我们使用熵权法给此二者赋予权重

若某个指标的信息熵越大，表明指标值得变异程度越大，提供的信息量越多，在综合评价中所能起到的作用也越大，其权重也就越大。相反，某个指标的信息熵越大，表明指标值得变异程度越小，提供的信息量也越少，在综合评价中所起到的作用也越小，其权重也就越小。

（1）标准化数据

假设给定了k个指标$X_1,X_2,\dots,X_k$，其中$X_i=\{x_1,x_2,\dots,x_n\}$。假设对各个指标数据标准化后的值为$Y_1,Y_2,\dots,Y_k$，那么$Y_{ij}=\frac{X_{ij}-min(X_i)}{max(X_i)-min(X_i)}$ (37)。

（2）信息熵

根据信息论中信息熵的定义，一组数据的信息熵：

$$E_j=-\ln(n)^{-1}\sum_{i=1}^{n}p_{ij}\,ln(p_{ij}) \quad (38)$$

其中$\mathrm{p}_{ij}=Y_{ij}/\sum_{i=1}^{n}Y_{ij}$ (39)，如果$\mathrm{p}_{ij}=0$，则定义$lim_{p_{ij}\to 0}\,p_{ij}\,ln(p_{ij})=0$ (40)。

（3）确定权重

根据信息熵的计算公式，计算出各个指标的信息熵为$E_1, E_2, \dots, E_k$通过信息熵计算各指标的权重：$w_i = \frac{1-E_i}{k-\sum E_i}\ (i = 1,2, \dots, k)$ (41)。

3. 结论

总结这次比赛，本文针对围绕用户留言，分别利用 Jieba 和 pyhanlp 对其分词处理，基于 TF-IDF 权重法提取特征词，构造词汇-文本矩阵，在其基础上通过贝叶斯模型，LinearSVC 模型，逻辑回归模型以及多层感知机模型对测试集做分类，得到最优分类器；利用 TSNE 降维方法对词汇-文本矩阵降维，并用 K-means 聚类法以及 HanLP 聚类算法确定最优簇数 K，利用相关指标结合 Topsis 评价方法得出留言中反应的热点问题，利用 TextRank 自动摘要得到热点问题合理的关键词描述；本文还通过对留言详情与答复意见的相关性、完整性、时效性和可解释性指标向量化，利用 Topsis 和 AHP 结合评价方法对答复意见的质量给出了评价。

本文**创新之处**在于利用 AHP 与 Topsis 相结合的熵权法评价答复意见，取得了良好的成果。

本文最后在 K-means 的聚类效果上表现欠佳，观察分类后热点问题，虽然调低了点赞数对值排序的比例，但受高点赞数的影响，排序结果仍然不是完美的。

在分词方面本文也可以利用 word2vec 分词方式，或者利用想词典中加入诸如“魅力小城”地点名来改善分词结果，进一步提高分类和聚类的准确度，后期也会对文本挖掘做近一步的思索与探讨。

参考文献

[1]简书.结巴中文分词介绍[OL].https://www.jianshu.com/p/1d525c86515d.

[2]杨锋.基于线性支持向量机的文本分类应用研究[J].信息技术与信息化.2020.

[3]庞博,成东坡.利用朴素贝叶斯分类器的视频分类方法[J].武汉工程职业技术学报.2019.

[4]毛林，陆全华，程涛.基于高维数据的集成逻辑回归分类算法的研究与应用.科技通报.[J].2013.

[5]博客园.多层感知机概述[OL].
https://www.cnblogs.com/somedayLi/p/12313804.html.

[6]简书.ROC 曲线原理及 Python 实现[OL].
https://www.jianshu.com/p/4b616c7838a3.2017.

[7]简书.pyhanlp 中文词性标注于分词简介[OL].
https://www.jianshu.com/p/145ba6722cbb.

[8]博客园.TSNE——目前最好的降维方法[OL].
https://www.cnblogs.com/bonelee/p/7849867.html.2017.

[9]肖艳炜，叶效威，曹坚成，李英.基于 K-means 算法的电力监控信息系统[R].中国会议.2017.

[10]简书.pyhanlp 文本聚类详细介绍[OL].
https://www.jianshu.com/p/f440e65a1791.2018.

[11]百度文库.topsis[OL].
https://wenku.baidu.com/view/05a44b63af1ffc4ffe47aca0.html.2012.

[12]陈志泊，李钰曼，许福. 基于 TextRank 和簇过滤的林业文本关键信息抽取研究[J].农业机械学报.2020.

[13]知乎.层次分析法(AHP)[OL]. https://zhuanlan.zhihu.com/p/38207837.

[14]CSDN.指标权重确定方法之熵权法[OL].
https://blog.csdn.net/qq_32942549/article/details/80019005.2018.