“智慧政务”中的文本挖掘应用研究

摘要

近年来，数据挖掘的运用范围日趋变大，许多传统的需要人工做的事情也可以通过数据挖掘让机器“学习”去处理一些事情，在数据挖掘方面，文本挖掘是其中及其重要的一个方面。利用文本挖掘让机器“看懂”人类的语言，可以帮助我们处理很多事情，例如在政务处理中对留言的划分和热点整理。随着大数据和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地推动作用。本文根据收集自互联网公开来源的群众问政留言记录及相关部门对部分群众留言的答复意见，利用自然语言处理和文本挖掘的方法解决下面三个问题。

针对问题一，我们首先对附件一和附件二的文本进行初步合并、处理，接着对总的文本进行预处理，预处理的过程中我们逐步完善以找到适合这些留言记录的方法，接下来对预处理好的内容进行向量化处理，最后尝试多种文本分类的方法找到较为适合的群众留言分类方法。

针对问题二，采用 Python 语言对附件三进行读取，用 jieba 库对留言主题进行清洗、去停用词、分词，对分词结果采用 K-Means 算法进行聚类，对聚类结果进行分析、评估，并进行制表。

针对问题三，我们首先读取附件四的留言和答复，其次从答复的相关性、完整性、可解释性等设置一系列的答复意见评价指标，例如答复与留言的相关性分析、回复与留言的时间差、回复的完整度等，接下来利用模糊数学的层次分析法确定每一个指标所占的权重，最后给出一个总的评价标准。

关键词：自然语言处理技术；朴素贝叶斯；KNN；支持向量机；K-means

Abstract

In recent years, the application range of data mining is becoming larger and larger. Many traditional things that need to be done manually can also be done through data mining to let the machine "learn" to deal with some things. In data mining, text mining is one of the most important aspects. Using text mining to make machines "understand" human language can help us deal with many things, such as the division of messages and hot spot sorting in government affairs processing. With the development of big data, artificial intelligence and other technologies, the establishment of smart government system based on natural language processing technology has become a new trend of social governance innovation and development, which has a great role in promoting the management level and governance efficiency of the government. In this paper, according to the records of the public political messages collected from the open sources of the Internet and the response opinions of the relevant departments to some of the public messages, we use natural language processing and text mining methods to solve the following three problems.

To solve the problem 1, we first merge and process the text in Annex 1 and Annex 2, then preprocess the general text. In the process of preprocessing, we gradually improve to find a method suitable for these message records. Next, we vectorize the preprocessed content, and finally try various text classification methods to find a more suitable mass message score Class method.

In view of the second problem, we use Python language to read the third attachment, use the Jieba library to clean the message subject, remove the stop words, and segment words. We use k-means algorithm to cluster the segmentation results, analyze, evaluate and tabulate the clustering results.

In response to question 3, we first read the message and response in Annex 4, and then set a series of response evaluation indicators from the relevance, integrity, and interpretability of the response, such as the correlation analysis between the response and the message, the time difference between the response and the message, the integrity of the response, etc., and then use the analytic hierarchy process of fuzzy mathematics to determine the weight of each indicator, Finally, a general evaluation standard is given.

Keywords: natural language processing technology;naive Bayes;KNN;support vector m achine;K-means

目录

一、 问题分析

在问题一“群众留言分类”中，我们一开始是对附件二的“留言详情”进行分词，后续的实验中我们又将附件二的“留言主题”加入进去且利用了附件一所给的三级标签，分词我们尝试了 jieba 分词和 THULAC，同时，我们在分词是也加入了停用词词典并将一些有关留言内容的词语加入用户自定义词典。接着是对文本的向量化和 TF-IDF 处理，最后我们分别建立了朴素贝叶斯、KNN、决策树、支持向量机的分类模型，运用 F-Score 比较这些分类效果的优良。

在问题二“热点问题挖掘”中，热点问题主要体现在一段时间集中爆发的问题或多人反映同一问题，我们将附件三的留言答复进行分词、K-means 聚类的方法找出相似留言，再把相似留言归类为同一问题，再将某一时间段反映特定地点或特征人群问题的留言寻找出来，将留言条数与点赞数综合起来作为热度评价指标，最后按照热度指标给出排名前 5 的热度问题和具体留言信息。

在问题三“答复意见的评价”中，我们分别从相关性、完整性、可解释性等角度来进行评价，最后通过熵权法，对回复的评论进行综合评价。评价指标中我们利用了文本挖掘中的相似度分析，向量化文本以确定相似度分析的具体实施步骤，关于权重的确定我们则采用模糊数学的层次分析法来确定各个指标的权重，结合得出的结果实例分析，我们不断调整权重来得到较为适合的评价方案。最后，我们给出最终评价出来的优秀回复。

二、 数据准备

2.1 预处理数据

检查是否有数据缺失与数据异常，消除重复数据并去除敏感字符，如把数据中的类似于身份证号码*****,把*替换成’’（空）。把清洗结果以 csv 的格式保存，以便后续其他程序处理。

2.2 添加词典停用词

将一些在文本中出现频率高但是含义虚泛的词放入停用词表，保证出现在停用词表中的词不能选作文档特征。如数字、符号、“的”、“而且”等文字和词语。

在本文中特定的地点、日期、线路等不能拆分，是作为分类的重要条件之一，建立新的用户自定义词典能更加准确的进行分类。

2.3 构造分析需要的指标

构建模型之前对文本分析，构造指标，图 2.3.1 为指标说明。

data	原始文本	classification_1s_tr	data_tr 对应的一级标签
data_after_stop	分词后的文本	classification_1s_te	data_te 对应的一级标签
classification	一级标签	pre	data_te 预测的一级标签
data_tr	训练集留言	f1_score	F-Score 对分类方法进行的评价
data_te	测试集留言	Wi	第 i 个因素所占权重

图 2.3.1 指标说明

2.4 分词

对附件二的“留言详情”进行分词，尝试通过 jieba 库、THULAC 和已添加的停用词字典，把文本精确分开。

2.5 文本向量化和 TF-IDF 处理

通过统计每个词在文本中出现的次数，得到该文本基于词的特征，将各个文本样本的这些词与对应的词频放在一起，进行向量化。用 TF-IDF 算法权衡某个分词是否为关键词的指标，该值越大是关键词的可能性越大。

2.6 绘制词云

将 jieba 分词后的结果用 WordCloud 进行分词可视化，可迅速得到文本中最突出、重要的词，词频越高的词，字体越大。图 2.6.1-2.6.7 分别对应每一分类。

图 2.6.1 城乡建设

图 2.6.2 环境保护

图 2.6.3 交通运输

图 2.6.4 教育文体

图 2.6.5 劳动和社会保障

图 2.6.6 商贸旅游

图 2.6.7 卫生计生

三、模型假设

为了便于问题的研究，对题目种某些条件进行简化及合理假设。

1) 附件三中由于一人给多个留言点赞情况无法得知，忽略该情况。
2) 在去重中，仅除去账号相同且留言相同的情况，允许账号不同但留言内容相似的情况出现。
3) 时间跨度过大、地点跨度大、人群分散但留言相似的情况不作为同一热点挖掘。

四、 任务一

4.1 分类模型分析

4.1.1 KNN 算法

KNN 算法是一种分类算法，它根据某个数据点周围的最近 k 个邻居的类别标签情况，赋予这个数据点一个类别。过程是：给定一个测试数据点，计算它于数据集中其他数据点的距离；找出距离最近的 k 个数据点，作为该数据点的近邻数据点集合；根据这 k 个近邻所归属的类别，来确定当前数据点的类别。该流程如下：

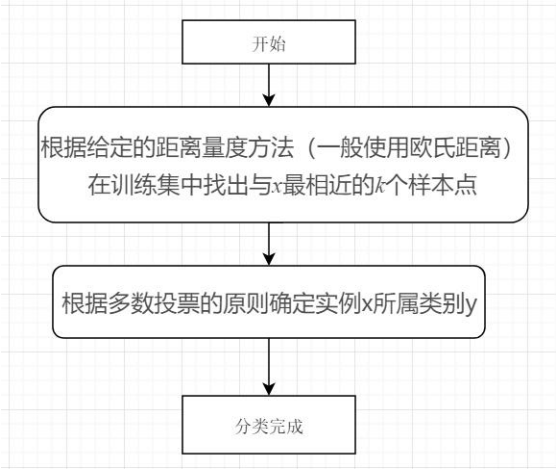

图 4.1.1 KNN 算法流程图

KNN 算法有以下优缺点：

a.优点：

① 该算法容易理解，容易实现，无须进行参数估计与训练过程，标注数据之后，直接进行分类即可；

② 可以对稀有的事件进行分类，使用于多类别分类。

b.缺点：

① 进行数据点分类时计算量大，内存开销大，执行速度慢；

② 无法给出类似决策树的规则，结果的可解释性差；

③ k 值的选择重要，k 值太小，则分类结果容易受到噪声数据点影响；k 值太大，则近邻中可能包含太多其他类别的数据点。

4.1.2 朴素贝叶斯算法

朴素贝叶斯分类是运用贝叶斯定理，并假设特征属性是条件独立的一种分类方法，即朴素贝叶斯分类器假设样本的每个特征与其他特征都不相关。准备阶段：为朴素贝叶斯分类做必要的准备，主要工作是根据具体情况确定特征属性，并对每个特征属性进行适当划分，然后对一部分待分类进行人工分类，进行训练样本集合，第一个阶段的数据质量对整个过程将有重要影响，分类器的质量很大程度上有特征属性、特征属性划分以及训练样本质量决定。训练阶段：主要任务是生成分类器，主要工作是计算每个类别在训练样本中出现频率，以及每个特征属性划分，对每个类别的条件概率估计，并且记录结果。应用阶段：主要任务是使用分类器对待分类项进行分类，也就是对新数据进行分类。假设有类别集合 $x=\{a_1,a_2,\cdots,a_m\}$，其中a_i是 x 的一个分类项，$i=1,2,\cdots,m$，此外有类别集合 $C=\{y_1,y_2,\cdots,y_n\}$，则朴素贝叶斯算法流程图如下：

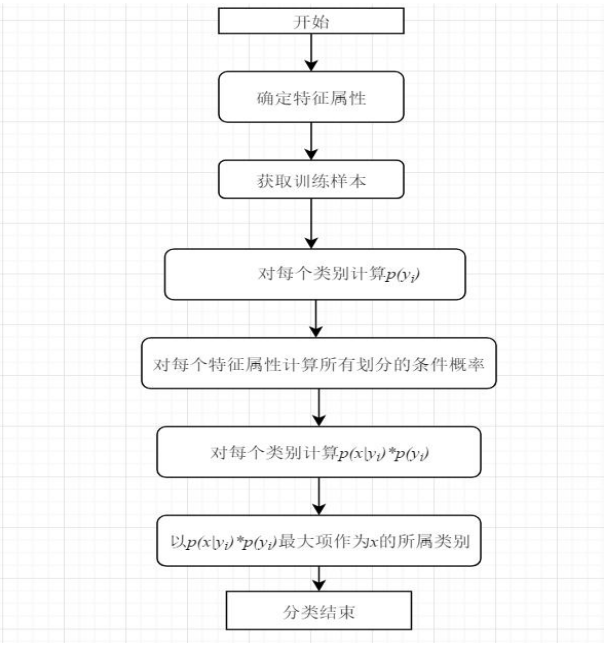

图 4.1.2 朴素贝叶斯算法流程图

朴素贝叶斯算法有以下优缺点：

a. 优点：

① 有稳定的分类效率，对小规模的数据表现很好，能够处理多分类任务；

② 适合增量式训练，尤其是数据量超出内存时，可以一批批的去增量训练；

③ 对缺失数据不太敏感，算法比较简单，常用于文本分类。

b.缺点：

① 在属性个数比较多或者属性之间相关性大时分类效果不好；

② 会由于假设的先验模型的原因导致预测的效果不佳；

③ 由于通过先验和数据决定后验概率，从而决定分类，分类决策存在一定的错误率，对输入数据的表达形式很敏感。

4.1.3 决策树

决策树学习本质是从训练集中归纳出一组分类规则。决策树的构造分两步进行。第一步决策树的生成：由训练样本集生成决策树的过程。一般情况下，训练样本数据集根据实际需要有历史的、有一定综合程度的，用于数据分析处理的数据集。第二步决策树的剪枝：决策树的剪枝是对上一阶段生成的决策树进行检验、矫正和修下的过程，主要是用新的样本数据集的数据校验决策树生成过程中产生的初步规则，将那些影响预测准确性的分支剪除。该算法流程大致如下：

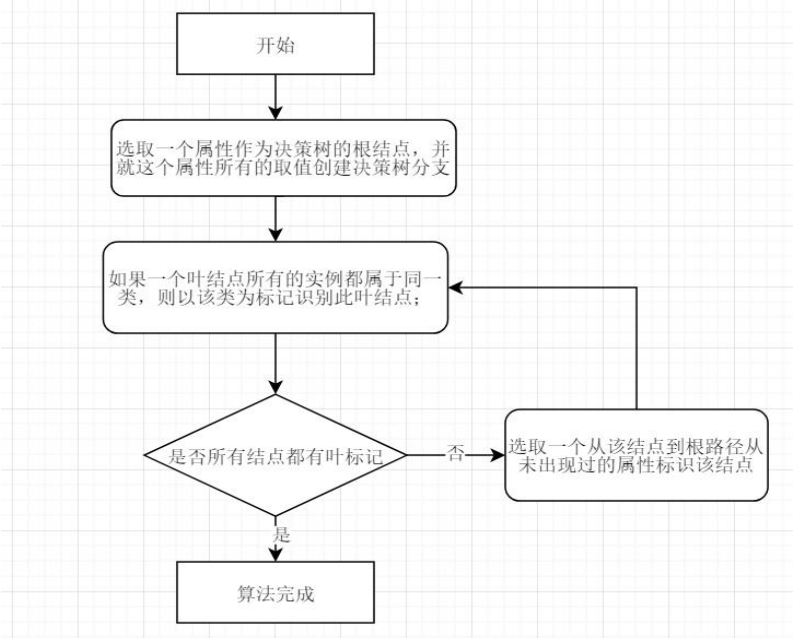

图 4.1.3 决策树算法流程图

决策树算法优缺点：

a.优点：

① 模型具有可读性，分类速度快，分类规则准确性高；

② 便于理解，不需要任何领域知识和参数假设。

b.缺点：

① 对于各类别样本数量不一致的数据，信息增益偏向于那些更多数值的特征；

② 容易过拟合，容易忽略属性之间的相关性。

4.1.4 支持向量机

支持向量机是一种二分类模型，它的基本模型是定义在特征空间上间隔最大的线性分类器。SVM 的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM 的学习算法就是求解凸二次规划的最优化算法。该算法流程图大致如下：

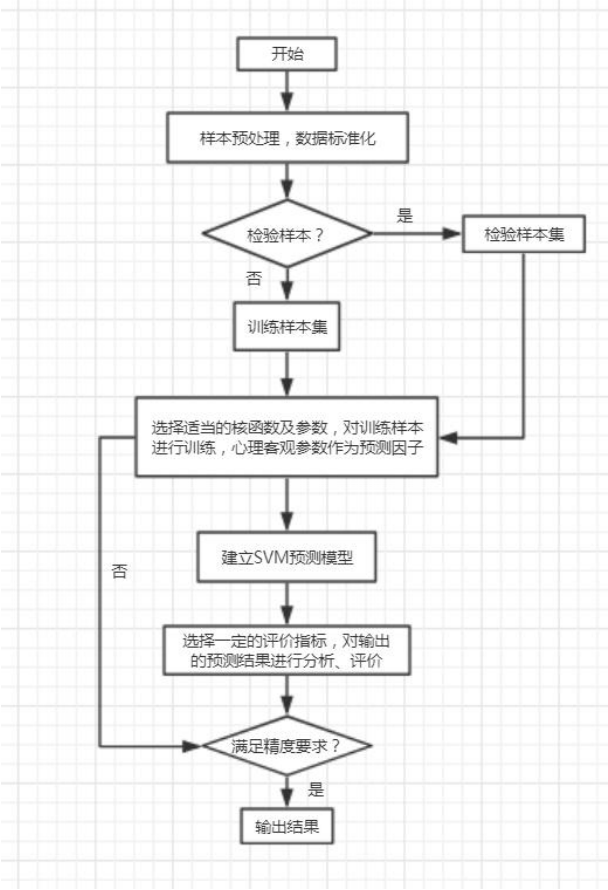

图 4.1.4 SVM 算法流程图

支持向量机算法优缺点：

a.优点：

① SVM 是一种有坚实理论基础的新颖的试用小样本学习方法，不涉及概率测度及大数定律等，简化了通常的分类和回归问题；

② 计算的复杂度取决于支持向量的数目，而不是样本空间的维数，避免了维数灾难；

③ 少数支持向量决定了最终结果，对异常值不敏感，可剔除大量冗余样本。

b.缺点：

① 对大规模训练样本难以实施，SVM 是借助二次规划来求解支持向量，而求解二次规划将涉及 m 阶矩阵的计算，当 m 数目很大时，该矩阵的存储和计算将消耗大量机器内存和运算时间；

② 解决多分类问题困难；

③ 对参数和核函数选择敏感，支持向量机性能的优劣主要取决于核函数的选取。

4.2 分类算法评价方式

F-Score 评价方法计算公式 $F-Score=(1+\beta^2)*\frac{\Pr ecision*\mathrm{Re}call}{(\beta^2*\Pr ecision)+\mathrm{Re}call}$，

精确率 $\Pr ecision=\frac{TP}{TP+FP}$，召回率 $\mathrm{Re}call=\frac{TP}{TP+FN}$

β 是用来平衡 Precision,Recall 在 F-Score 计算中的权重，取值情况有以下三种：

a. 如果取 1，表示 Precision 与 Recall 一样重要

b. 如果取小于 1，表示 Precision 比 Recall 重要

c. 如果取大于 1，表示 Recall 比 Precision 重要

在本文中，精确率与召回率同样重要，于是取 β 值为 1.

则得到得分公式 $\mathrm{F}=\frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$，将四个算法分别带入计算得分即可。

4.3 四种分类算法实证分析

我们将四种算法的实现过程写进一个工程文件，对每一类数据分别抽取若干条数据作为测试样本进行学习。这些学习的数据包括留言标题，内容以及附件 1 中的三级标签。剩下的数据则用于测试检验。从结果来看，随着样本数据的不断增大，各方法的准确率都有小幅度提高。除决策树外，其他方法效果都比较好。在测试样本达到 800 条时，采用 F-score 对测试数据检验的效果如下

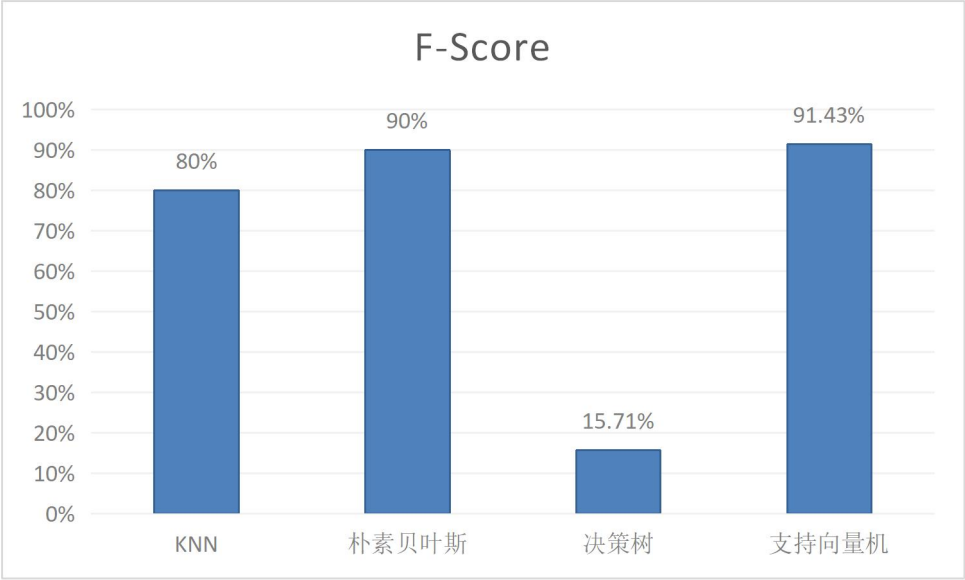

图 4.3.1 F-score 检验下的四种算法准确率对比

可以看出，这三种方法中，决策树的效果远不如其他三种，我们对效果较好的三种方法针对每一个具体的分类效果作图如下：

这三种方法中，不同的方法对不同类别的有不同的效果，例如朴素贝叶斯对“城乡建设”这一类的准确率是、支持向量机对“交通运输”和“卫生计生”的准

确率均为 100%。在后续的实际操作中，我们在文本挖掘进行分类时也可以考虑不同的类别采用不同的方法，这样也是一种提高整体准确率的方法。下面分别给出了三种分类方法的测试准确率。

图 4.3.2 KNN 分类测试数据准确率

图 4.3.3 朴素贝叶斯分类测试数据准确率

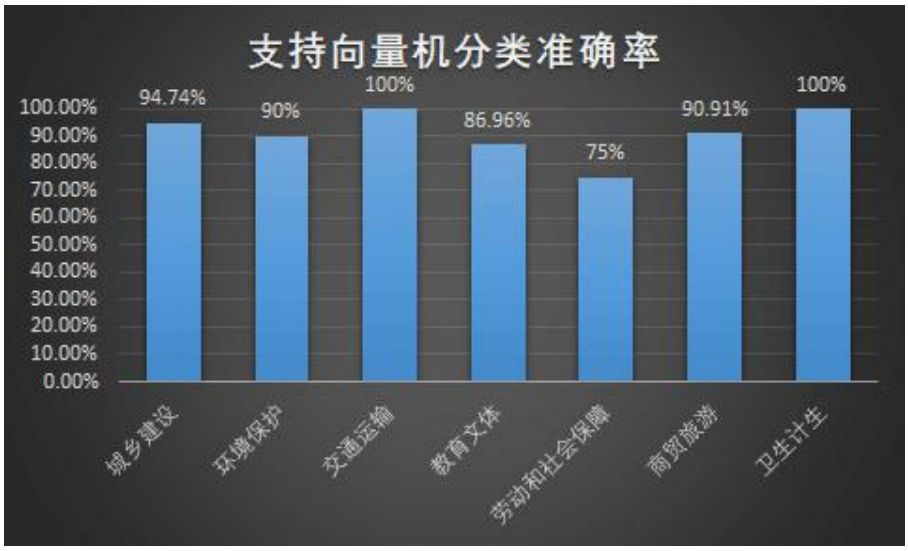

图 4.3.4 支持向量机分类测试数据准确率

4.4 小结

本题我们先对读入数据进行预处理、去停用词、分词处理，再分别采用上述的四种算法分别对处理后的数据进行一定量的学习，最后对分类结果进行评估。从结果来看，支持向量机的分类方法准确率最高。相比之下，由于决策树在处理模式中是基于二叉树的，在分类数较多，数据量较大时，其判断准确率与其他方法相比就显得比较低。

五、 任务二

本题的实现过程主要有语句清洗、去停用词、文本分词、中心聚类、分析校正等。技术路线流程图如下：

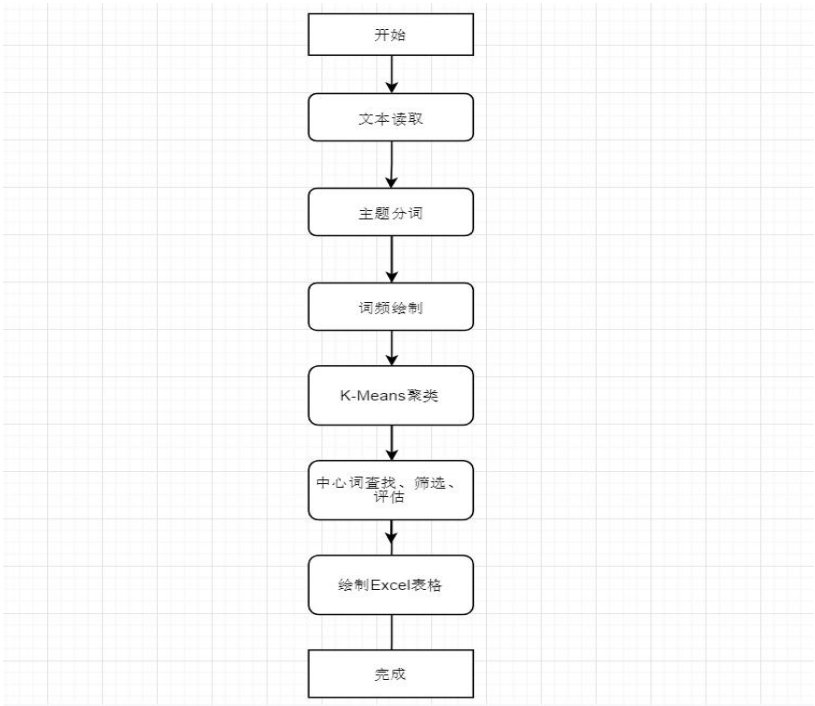

图 5 问题二：热点问题挖掘流程图

5.1 聚类模型分析

K-means 算法是硬聚类算法，是典型的基于原型的目标函数聚类方法的代表，它是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。K-means 算法以欧式距离作为相似度测度，它是求对应某一初始聚类中心向量 V 最优分类，使得评价指标最小。算法采用误差平方和准则函数作为聚类准则函数。其实现过程大致如下：

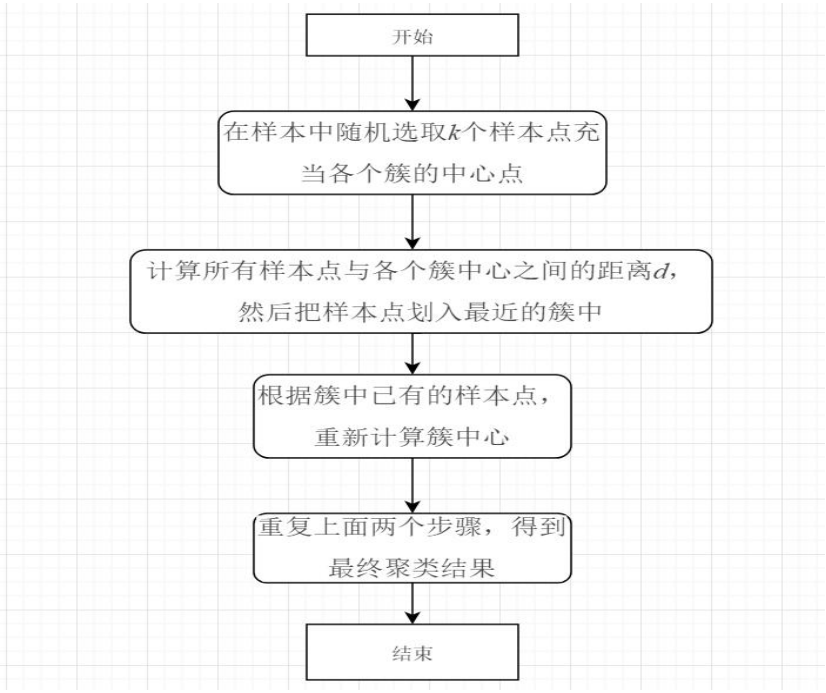

图 5.1 k-means 流程图

a.优点：

① 算法简单、快速，是解决聚类问题的一种经典算法；

② 对处理大数据集，该算法保持可伸缩性和高效性；

③ 当簇接近高斯分布时，它的效果较好。

b.缺点：

① 在 K-means 算法中 K 值的选定是非常难以估计的；

② 在 K-means 算法中，首先需要根据初始聚类中心来确定一个初始划分，然后对初始划分进行优化。这个初始聚类中心的选择对聚类结果有较大的影响；

③ 若簇中含有异常点，将导致均值偏离严重；

④ 不适用于发现凸形状的簇或者大小差别很大的簇。

5.2 留言主题分词

本部分与 4.1-4.3 中的过程类似。由于数据不同，自定义词典在内容上有所差异，但对于程序和算法整体没有太大影响。故这里不再赘述。本题尝试使用了清华大学自定义 thulac 的第三方库标记词性，最终经过对比权衡选择，仍采用 jieba 进行分词。

5.3 留言主题聚类分析

承接上文，已经分词的留言主题是列表形式。现采用 K-Means 算法对全体留言主题进行聚类分析，结果分别令 k 值取 150,200,250，经对比认为分为 200 类时效果较好。以下是分类效果展示。

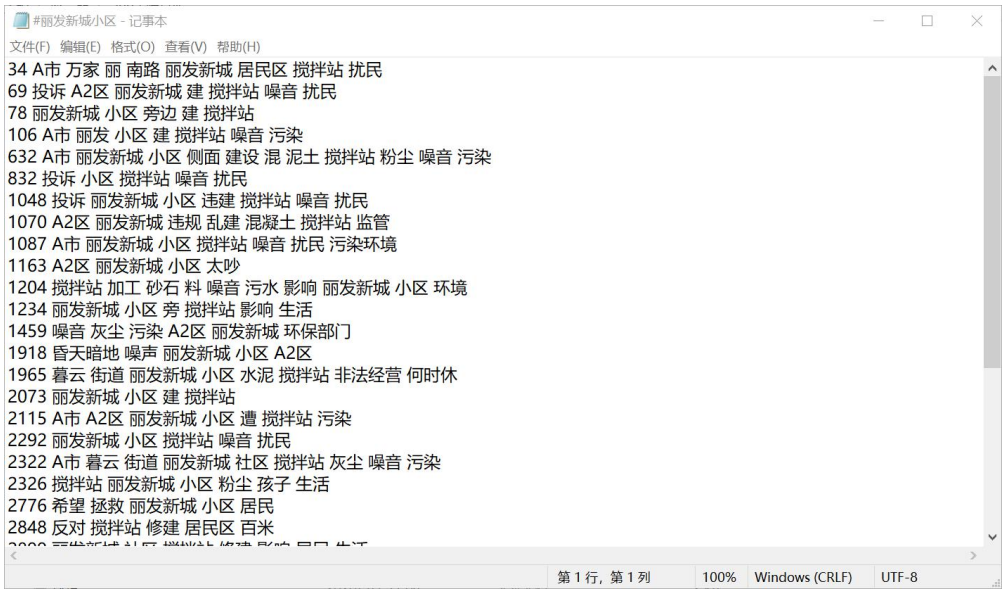

图 5.3.1 K-Means 聚类算法效果展示

此外，我们还制作了一个词频的 Excel 表格，用以辅助检查分类的准确程度。将上述分类结果进行对比、分析、检验，辅以关键词筛选，最终绘制出了热点问题表和热点问题留言明细表（见附件）。

5.4 小结

本题的最终目标是最热点问题进行挖掘、排序。因此第一步——进行合理有效的分词显得尤为重要。为此我们尝试了 jieba、thulac 等对比分词效果。命名实体识别也是一个较为不错的方法，但考虑到文本中存在大量特殊地点，这些地点（如小区名）由于是较小的地点而难以被识别，从而我们放弃了尝试这种做法。K-Means 聚类算法是一个门槛低，适用性强，应用范围广的方法，这里比较好地承接了分词结果经过取值的放缩，我们得到了一个较为不错的聚类数。此时根据聚类结果代入原文件中查找对比，辅以词频，最终有效解决了热点问题挖掘问题。不足之处则是未能找到最优聚类，例如可采用相似度分析对结果进行进一步优化，此外，还可以进行文本特征向量的构造。

六、 任务三

本题的实现过程主要有确定评价指标、确定各指标权重、给出评价、得出优质回复等、分析校正等。技术路线流程图如下：

图 6 任务三流程图

6.1 确定评价指标

6.1.1 相关性

进行留言与留言回复的相似度分析，相似度越大，相关性越好。

6.1.2 完整性

答复留言的字数以及格式，如“网友您好”、“感谢对我们工作的支持理解”等，格式越完整，评价越好。

6.1.3 可解释性

在这里为解释质量，是否有效地帮助用户，解决了用户的问题。可解释性越好，最终评价得分越好。

6.1.4 回复与留言的时间差

当答复时间与留言时间的时间差越短时，回复的就越及时，该答复留言的评价就越好。

6.2 确定各指标权重

6.2.1 层次分析法

层次分析法是对难以完全定量的复杂系统做出决策的模型和方法。层次分析法根据问题的性质和要达到的总目标，将问题分解为不同的组成因素，并按照因素间的相互关联影响以及隶属关系将因素按不同的层次聚集组合，形成一个多层次的分析结构模型，从而最终使问题归结为最低层（供决策的方案、措施等）相对于最高层（总目标）的相对重要权值的确定或相对优劣次序的排定。

基本步骤：

1、建立层次结构模型。在深入分析实际问题的基础上，将有关的各个因素按照不同属性自上而下地分解成若干层次，同一层的诸因素从属于上一层的因素或对上层因素有影响，同时又支配下一层的因素或受到下层因素的作用。最上层为目标层，通常只有 1 个因素，最下层通常为方案或对象层，中间可以有一个或几个层次，通常为准则或指标层。当准则过多时（譬如多于 9 个）应进一步分解出子准则层。

2、构造成对比较阵。从层次结构模型的第 2 层开始，对于从属于（或影响）上一层每个因素的同一层诸因素，用成对比较法和 1—9 比较尺度构造成对比较阵，直到最下层。

3、计算权向量并做一致性检验。对于每一个成对比较阵计算最大特征根及对应特征向量，利用一致性指标、随机一致性指标和一致性比率做一致性检验。若检验通过，特征向量（归一化后）即为权向量：若不通过，需重新构造成对比较阵。

4、计算组合权向量并做组合一致性检验。计算最下层对目标的组合权向量，并根据公式做组合一致性检验，若检验通过，则可按照组合权向量表示的结果进行决策，否则需要重新考虑模型或重新构造那些一致性比率较大的成对比较阵。

层次分析法的优点：

适用于存在不确定性和主观信息的情况，如本题情况。

6.3 给出评价

时间差最大有相隔三年后回复，最小仅 30 分钟，平均在 20 天左右会对网友进行答复；回复字数最多 7883 字，有些没有正面回复用户留言内容，平均回复字数在 360 字左右；相似度分析方面，最匹配的可以高达 90%，而有一些甚至直接回复“您所反映的问题，已转交相关部门调查处置。”。在权重的确定是，我们经过在网上调查网友对政务回复质量看重的方面，结合层次分析法进行计算，最终我们得到相似度：时间差：留言字数=0.5:0.3:0.2。将权重乘以每一项对应的百分制分数，我们最终得到一个总的分数，通过从高到低进行排序，我们可以定位出对应的答复，从而找到优质的答复意见。

6.4 得出优质回复

经过上面的步骤，我们筛选出评分前五的回复如下：

留言编号	留言用户	留言主题	留言时间	留言详情	答复意见	答复时间
88359	UU0082390	咨询J9县的油茶种植政策	2019/1/5 15:52:14	机农产品品牌建设的决定和农	为目标，以品牌建设为动力，以“原生态、	2019/1/7 17:00:37
88612	UU0081032	家庭大病医疗慈善救助相关	2018/8/9 17:05:43	于申请特困家庭大病医疗慈善	疾人。4、突发性灾害导致家庭特困和人员	2018/8/17 9:29:45
96762	UU0082390	农产品品牌建设扶持政策的	2018/12/15 23:28:40	生态有机农产品品牌建设的决	组织为主体，以产业发展为目标，以品牌建设	2019/1/3 10:53:46
96757	UU0082390	咨询J9县的油茶种植政策	2019/1/5 15:56:23	机农产品品牌建设的决定和农	以新型组织为主体，以产业发展为目标，以	2019/1/16 8:38:52
8380	UU00878	议A市强化对共享单车的管	2017/6/22 7:42:41	单车的投放方，一定要到城市	停放提供了硬件支持。二是从市政管理及服	2017/7/12 10:02:06

图 6.4 优质回复

具体看第一个，我们可以看到，留言开头有“首先感谢您对 J9 县扶贫工作的支持与关注！针对您咨询的关于加快推进原生态有机农产品品牌建设的决定和农产品品牌建设扶持政策，现回复如下：”这样较为完整的格式，而且后面的回答内容不仅与留言详细的内容十分符合，也引用了多个关于这个问题的一些条例，准确有力地阐释了自己的观点。同时，这条留言的长度比较长，回复与留言的时间差也较短。后面几条的回复也是如此，再具体问题分析时，我们也可以通过层次分析法及时得出最为适合的权重。

总的来说，这种方式能够结合答复的相关性、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。

七、 参考文献

[1]运输车辆安全驾驶行为分析.第七届泰迪杯挑战赛

[2]周志华.机器学习[M].北京：清华大学出版社，2016

[3]张倩.用于网络评论文本挖掘的主题模型研究[D].2014.

[4]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,2017,038(004):47-51.

[5]李廷辰，杨艳.基于分词聚类技术的微博热点问题挖掘[J].教学与科技,2013(1):8-13.

八、附录

1. 建模所用软件

写作：Word2016，Excel2016

编程软件：Python3.6.8，MATLAB 2018A，Pycharm64

2. 问题一核心代码

①数据处理

```
import pandas as pd
import re
import jieba

def data_process(file='fujian2.csv'):
    data = pd.read_csv('fujian2.csv', encoding="gbk", header=None, index_col=0)
    data.columns = ['user', 'theme', 'time', 'detail', 'classification_1','classification_3','theme_detail']
    n = 50

    a = data[data['classification_1'] == '城乡建设'].sample(n)
    b = data[data['classification_1'] == '环境保护'].sample(n)
    c = data[data['classification_1'] == '交通运输'].sample(n)
    d = data[data['classification_1'] == '教育文体'].sample(n)
    e = data[data['classification_1'] == '劳动和社会保障'].sample(n)
    f = data[data['classification_1'] == '商贸旅游'].sample(n)
    g = data[data['classification_1'] == '卫生计生'].sample(n)

    data_new = pd.concat([a, b, c, d, e, f, g], axis=0)

    data_dup = data_new['theme_detail'].drop_duplicates()
    data_qumin = data_dup.apply(lambda x: re.sub('#', '', x))

    jieba.load_userdict('newdic.txt')
    data_cut = data_qumin.apply(lambda x: jieba.lcut(x))

    stopWords = pd.read_csv('stopword.txt', encoding='utf-8', sep='hahaha', header=None)
    stopWords = ['\n', '\t', ' ', '  ', '   ', '    ', '\r\n'] + list(stopWords.iloc[:, 0])

    data['classification_1'].value_counts()
    data_after_stop = data_cut.apply(lambda x: [i for i in x if i not in stopWords])
    print(data_after_stop)
    classification_1s = data_new.loc[data_after_stop.index, 'classification_1']
    adata = data_after_stop.apply(lambda x: ' '.join(x))

    return adata, data_after_stop, classification_1s
```

②模型构建

```
from data_process1 import data_process1
from data_process2 import data_process2
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.svm import LinearSVC
from xgboost import XGBClassifier
from sklearn.metrics import f1_score

adata1, data_after_stop1, classification_1s1 = data_process1()
adata2, data_after_stop2, classification_1s2 = data_process2()
data_tr, data_te, classification_1s_tr, classification_1s_te = adata1, adata2, classification_1s1, classification_1s2

countVectorizer = CountVectorizer()
data_tr = countVectorizer.fit_transform(data_tr)
X_tr = TfidfTransformer().fit_transform(data_tr.toarray()).toarray()

data_te = CountVectorizer(vocabulary=countVectorizer.vocabulary_).fit_transform(data_te)
X_te = TfidfTransformer().fit_transform(data_te.toarray()).toarray()

n = int(input("input:"))

if n==1:
    model = KNeighborsClassifier(n_neighbors=5)
elif n==2:
    model = GaussianNB()
elif n==3:
    model = DecisionTreeClassifier(max_depth=5, random_state=8)
elif n==4:
    model = LinearSVC().fit(X_tr, classification_1s_tr)

model.fit(X_tr, classification_1s_tr)

model.score(X_te, classification_1s_te)
print(model.score(X_te, classification_1s_te))

pre = model.predict(X_te)
print(f1_score(classification_1s_te,pre,average=None))
```

3. 问题二核心代码

①模型构建

```
import pandas as pd
import re
import jieba
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.cluster import KMeans
from sklearn.metrics import f1_score

data = pd.read_csv('fujian3.csv', encoding="gbk", header=None, index_col=0)
data.columns = ['user', 'theme', 'time', 'detail', 'against', 'agree']

data_dup = data['theme'].drop_duplicates()
data_qumin = data_dup.apply(lambda x: re.sub('#', '', x))

jieba.load_userdict('newdic.txt')
data_cut = data_qumin.apply(lambda x: jieba.lcut(x))

stopWords = pd.read_csv('stopword.txt', encoding='utf-8', sep='hahaha', header=None)
stopWords = ['\n', '\t', ' ', '  ', '   ', '\r\n'] + list(stopWords.iloc[:, 0])

data_after_stop = data_cut.apply(lambda x: [i for i in x if i not in stopWords])

adata = data_after_stop.apply(lambda x: ' '.join(x))

countVectorizer = CountVectorizer()
adata_ = countVectorizer.fit_transform(adata)
X_te = TfidfTransformer().fit_transform(adata_.toarray()).toarray()

model = KMeans(n_clusters=500).fit(X_te)

for i in range (0,len(model.labels_)):
    num = model.labels_[i]
    name = '%04d' % (num+1)
    filename = str(name) + '.txt'
    f = open(filename, "a+", encoding = "utf8")
    f.write(str(i)+' ')
    f.write(str(adata[i])+'\n')

print(model.labels_)
print(len(model.labels_))
```

4. 问题三核心代码

①总评价

```
from xiangsidu import xiangsidu
from time import time
from zishu import zishu
import pandas as pd
import jieba
import re

data = pd.read_csv('fujian4.csv', encoding="gbk", header=None, index_col=0)
data.columns = ['user', 'theme', 'time1', 'detail', 'opinion', 'time2', "time", "time3", "zishu", "time4"]

detail = list(data["detail"])
del detail[0]
texts = detail[1]

opinion = list(data["opinion"])
del opinion[0]
keyword = opinion[1]

data_detail = data['detail']
data_opinion = data['opinion']

data_qumin1 = data_detail.apply(lambda x: re.sub('#', '', x))
data_qumin2 = data_opinion.apply(lambda x: re.sub('#', '', x))

jieba.load_userdict('newdic.txt')
data_cut1 = data_qumin1.apply(lambda x: jieba.lcut(x))
data_cut2 = data_qumin2.apply(lambda x: jieba.lcut(x))

stopWords = pd.read_csv('stopword.txt', encoding='utf-8', sep='hahaha', header=None)
stopWords = ['\n', '\t', ' ', '  ', '   ', '    ', '\r\n'] + list(stopWords.iloc[:, 0])

data_after_stop1 = data_cut1.apply(lambda x: [i for i in x if i not in stopWords])
data_after_stop2 = data_cut2.apply(lambda x: [i for i in x if i not in stopWords])

a1 = []
for i in range (1,len(data_after_stop1)):
    a1.append(data_after_stop1[i])
a2 = []
for i in range (1,len(data_after_stop2)):
    a2.append(data_after_stop2[i])

all_doc_list_ = []
for i in range (0,len(a1)):
    all_doc_list_.append(a1[i])
all_test_list_ = []
for i in range (0,len(a2)):
    all_test_list_.append(a2[i])

t = list(data["time"])
del t[0]
for i in range (0,len(t)):
    t[i] = float(t[i])

z = list(data["zishu"])
del z[0]
for i in range (0,len(z)):
    z[i] = int(z[i])

w1, w2, w3 = 0.5, 0.3, 0.2

r1 = xiangsidu(all_doc_list_, all_test_list_)

fenshu = []
for i in range (0,len(t)):
    x = t[i]
    if x <= 5:
        f = 100
    else:
        f = 100/(1+0.02*(x-5)*(x-5))
    fenshu.append(f)

r3 = zishu(z)

R = []
for i in range(0,len(r1)):
    x = w1*r1[i]+w2*fenshu[i]+w3*r3[i]
    R.append(x)
print(len(R))

result = open('fenshu.xls', 'w', encoding='gbk')
result.write('留言\t分数\n')
for i in range(len(R)):
    result.write(str(i+1))
    result.write('\t')
    result.write(str(R[i]/max(R)*100))
    result.write('\n')
result.close()
```

②层次分析法

```
function [W,lambda] = cengcifenxi(A)
[m,n]=size(A);
lmdmax=0;
if (m~=n || n<3)
    W=[];
    return;
end % 输入数据有误
for (i=1:n)  % 检验A是否为判断矩阵
    if (A(i,i)~=1)
        W=[];
        return;
    end  % 输入数据有误
    for (j=i+1:n)
        x=A(i,j);
        if (x<=0 || x*A(j,i)~=1)
            W=[];
            return;
        end   % 输入数据有误
    end
end
for (j=1:n)
    x=0; % 列向量归一化
    for (i=1:n)
        x=x+A(i,j);
    end
    for (i=1:n)
        B(i,j)=A(i,j)/x;
    end
end
for (i=1:n)
    W(i,1)=0;  % 计算最大特征值相应的特征
    for (j=1:n)
        W(i)=W(i)+B(i,j);
    end % 行和
end
W=W/n;  % 特征向量归一化
for (i=1:n)
    lmdmax=lmdmax+A(i,:)*W/W(i);
end
lmdmax=lmdmax/n;
```