“智慧政务”中的文本挖掘应用

摘　要：近年来，随着科学技术与移动互联网的蓬勃发展，微信、微博、市长信箱、阳光热线等网络问政平台也逐步成为了政府了解民意、汇聚民智、凝聚民气的重要渠道。随着大数据、云计算、人工智能等技术的发展，基于自然语言处理技术的智慧政务系统的社会治理创新发展趋势对提升政府的管理水平和施政效率具有极大的推动作用。与此同时，各类社情民意相关的文本数据量不断攀升，给以往主要依靠人工来进行留言划分和热点整理的相关部门的工作带来了极大的挑战。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，除了基础的留言分类处理，进一步了解群众的心声也变得越开越有必要。其中非常重要的方式就是对群众的文本留言数据进行内在信息的数据挖掘分析，而得到的这些信息，并且能够针对相关部门对留言的答复意见多角度对其质量给出评价方案。本文将基于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数据进行内在信息的挖掘与分析。

在本次数据挖掘过程中，我们首先对获取得到的留言数据利用 python 以及 Matlab 工具进行数据预处理、分词以及停用词过滤操作，实现了对留言数据的优化，并提升了其可建模度。接着，采用多种方法来进行数据挖掘模型的构建，为后面的留言分析构建分析的基础。为此我们先利用深度学习的方法，分别用朴素贝叶斯分类器，K 最近邻分类器，线性分类对留言分类；其次构建语义网络，进行基于 K-means 模型的量化分析；再有，结合统计学的角度实现评论主题模型的构建。最后，运用构造出来的多种数据挖掘模型的结果，对这些留言数据进行多方面多角度的留言文本分析，以提取留言中隐藏的信息。无监督聚类技术被用以进行情感倾向性分析，基于 K-means 模型的量化分析，一定程度上得到了扰民因素包括噪音、油烟等信息；TF-IDF 算法则滤取出了从统计学角度上的给予网络问政平台留言的重点信息，以了解群众一般关注的对象及现象。

关键词：留言数据；信息提取；文本分析；K-means 模型

Text Mining Application in "Smart Government Affairs"

Abstract: In recent years, along with the vigorous development of science and te chnology and mobile Internet, WeChat, Weibo, Mayor's Mailbox, Sunshine Hotline a nd other network questioning platforms have gradually become an important channel for the government to understand public opinion, gather people's wisdom and gather p eople's popularity. With the development of big data, cloud computing, artificial intell igence and other technologies, the development trend of social governance innovation of smart government systems based on natural language processing technology has gr eatly promoted the government's management level and governance efficiency. At the same time, the amount of text data related to various social conditions and publ ic opinion has continued to rise, which has brought great challenges to the work of rel evant departments that used to manually divide messages and organize hotspots. At pr esent, most e-government systems still rely on manual processing based on experience . There are problems such as large workload, low efficiency, and high error rate. In ad dition to basic message classification processing, it is necessary to further understand the voices of the people. One of the most important ways is to carry out data mining a nd analysis of the internal information of the text message data of the masses, and the obtained information can also give an evaluation plan for the quality of the response o f the relevant departments to the message from multiple angles. This article will mine and analyze the internal information based on the records of the public's question and answer messages published on the Internet and the response data of some departments ', responses to some of the people's messages. In this data mining process, we first us e python and Matlab tools to perform data preprocessing, word segmentation and stop word filtering operations on the obtained message data, to optimize the message data and improve its modelability . Then, a variety of methods are used to construct the dat a mining model, and the basis for the analysis of the subsequent message analysis is c onstructed. To this end, we first use deep learning methods to classify messages using Naive Bayes classifier, K nearest neighbor classifier, and linear classification; secondl y, construct a semantic network and perform quantitative analysis based on K-means model; From the perspective of statistics, the construction of comment topic model is realized. Finally, using the results of the constructed multiple data mining models, the message text of the message data is analyzed in many aspects and from various angles to extract the hidden information in the message. Unsupervised clustering technology is used to analyze sentiment tendency. Based on the quantitative analysis of K-means model, to a certain extent, information on disturbance factors including noise and oil s moke is obtained.TF-IDF algorithm filters out from a statistical point of view gave the key messages of the online questioning platform to understand the objects and pheno mena that people generally pay attention to.

Keywords: message data,information extraction,text analysis, K-means model

目录

摘　要 1

1. 挖掘目标 4

2. 分析方法与过程 4

2.1 总体流程 4
2.2　具体步骤 5
2.2.1 问题一分析方法与过程 5
2.2.1. 1 流程图 5
2.2.1.2 数据介绍 5
2.2. 1.3 数据整理 5
2.2. 1.4 数据预处理 6
2.2.1.5 TF-IDF 算法 9
2. 2.1.6 留言分类 10
2.2.1.7 模型评估 11
2.2.2 问题二分析方法与过程 15
2.2.2.1 基于 K-means 模型的量化分析 15
2.2.2.2 K-means 模型介绍 16
2.2.2.3 K-means 模型原理 16
2.2.2.4 原始 K-means 算法与本文所用改良后 K-means 算法的对比 17
2.2.2.5 运用 K-means 模型进行量化分析的实现过程 18
2.2.3 问题三分析方法与过程 18
2.2.3.1 答复意见评价的实现 18
2.3 结果分析 21

3.结论 33

4.参考文献 33

1. 挖掘目标

本次建模针对网络问政平台群众的文本留言数据，在对文本进行基本的机器预处理、中文分词、停用词过滤后，通过 TF-IDF 算法、K-means 模型的量化分析，实现对文本留言数据的倾向性判断以及所隐藏的信息的挖掘并分析，以期望得到有价值的内在内容。

2. 分析方法与过程

2.1 总体流程

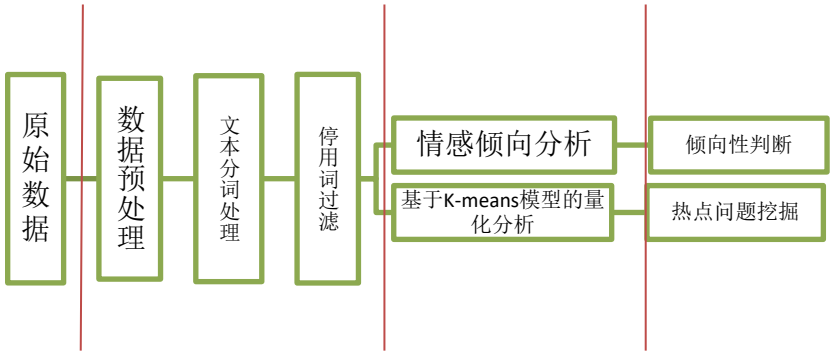


Figure 1 总体流程图

本论文的分析流程可大致分为以下四步：

第一步：获取分析所用的原始数据（文本留言），部分数据自行爬取；

第二步：对获取的数据进行基本的处理操作，包括数据预处理、中文分词处理、停用词过滤等操作；

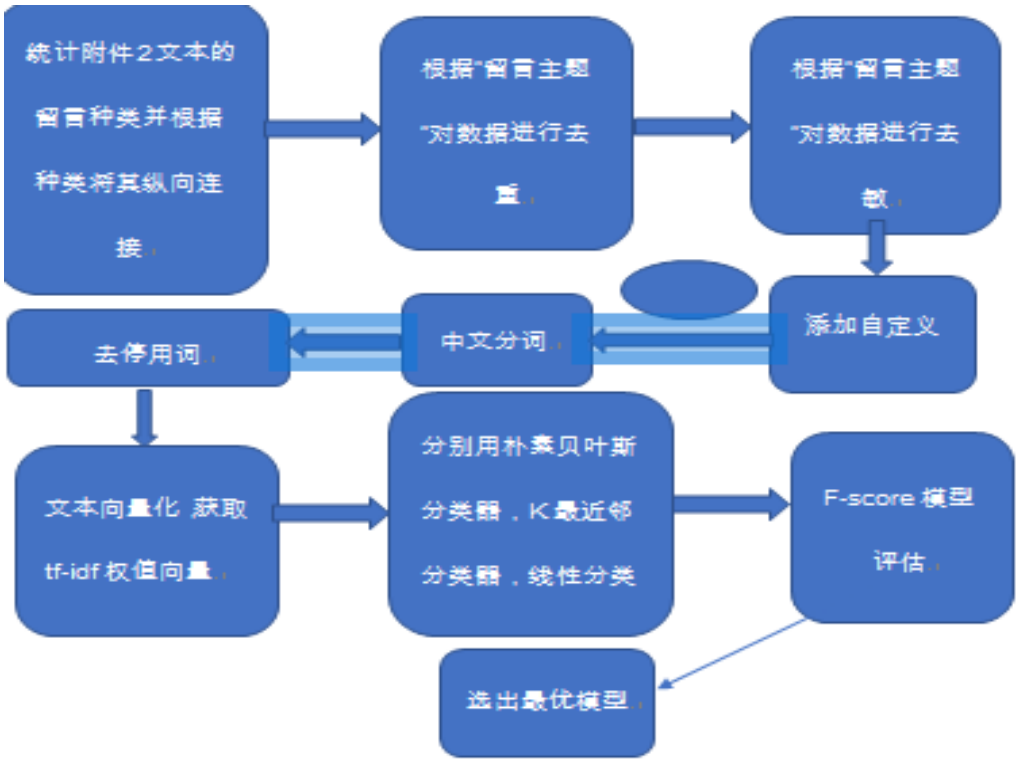
第三步：文本留言数据经过处理后，运用多种手段对留言数据进行多方面的分析；

第四步：从对应结果的分析中获取文本留言数据中有价值的内容。

2.2具体步骤

2.2.1 问题一分析方法与过程

2.2.1.1 流程图



2.2.1.2 数据介绍

本题使用的的数据附件 1 和附件 2 均为来源于互联网公开渠道的文本数据（附件 1 数据具有参照作用，需要我们根据附件 2 中具体数据对附件 1 中的一级标签进行筛选），附件 1 提供的内容为三级分类标签，附件 2 分为留言编号、留言用户、留言主题、留言时间、留言详情和一级标签六列，其中留言编号为数据索引，本题对留言用户和留言时间没有要求，留言详情数据冗杂，因此主要针对留言主题为依据进行处理分类。

2.2．1.3 数据整理

对于题目给出的附件 2，数据杂乱毫无规律，各种分类交错排序，为了使数

据看上去更加直白美观，分别抽取“城乡建设”、“环境保护”、“交通运输”、“教育文体”、“劳动和社会保障”、“商贸旅游”和“卫生计生”七个种类的数据，并进行纵向拼接。

2.2．1.4 数据预处理

得到拼接后的文本后，首先对文本数据进行预处理。文本数据中存在大量重复且没有意义的数据，我们需要对这些数据进行去重、分词、去停用词、词频统计等，若不处理后期对文本的分析挖掘将会产生很大的误差，因此必须先要对文本数据进行预处理。我们分别以去重，去敏感词，分词，去停用词的顺序对文本数据进行预处理。

（1）文本去重

文本去重即去除文本中重复的部分，本题由于留言详情内容冗杂，所以主要研究留言主题，对留言主题内容重复的进行去重，文本去重的主要原因如下：

① 由于本文的数据都来自互联网的公开渠道，不排除在收集数据的时候有将数据重复录入的情况，这样重复的数据会使获得的信息重复，在后续词频统计中对结果分析产生误差。

② 群众留言向政府提交自己的意见，若短时间内得不到答复，不排除同一个用户多次提交重复评论或相近评论，或者复制粘贴他人评论的可能，我们只需提取最有意义的一条便可。

将数据整理之后获得 9210 行的文本数据，运用 drop duplicates 对“留言主题”这一列重复的数据进行去重处理，得到 8905 行数据，可见去重了 305 条数据，可见重复率达到 3.312％。

（2）文本去敏

通过上述文本去重，去掉重复的留言主题，但是在读取数据时能看到存在非汉字的特殊字符，文本中存在许多没有意义的，会影响到数据的质量，因此我们只提取中文汉字部分，这样为后面的分词做准备。

（3）中文分词

中文分词(Chinese Word Segmentation) 指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。现有的分词算法可分为三大类：基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法。

基于字符串匹配的分词方法：这种方法又叫做机械分词方法，它是按照一定的策略将待分析的汉字串与一个“充分大的”机器词典中的词条进行配，若在词典中找到某个字符串，则匹配成功（识别出一个词）

1）正向最大匹配法（由左到右的方向）

2）逆向最大匹配法（由右到左的方向）:

3）最少切分（使每一句中切出的词数最小）

4）双向最大匹配法（进行由左到右、由右到左两次扫描）

基于理解的分词方法：这种分词方法是通过让计算机模拟人对句子的理解，达到识别词的效果。其基本思想就是在分词的同时进行句法、语义分析，利用句法信息和语义信息来处理歧义现象。它通常包括三个部分：分词子系统、句法语义子系统、总控部分。在总控部分的协调下，分词子系统可以获得有关词、句子等的句法和语义信息来对分词歧义进行判断，即它模拟了人对句子的理解过程。这种分词方法需要使用大量的语言知识和信息。由于汉语语言知识的笼统、复杂性，难以将各种语言信息组织成机器可直接读取的形式，因此目前基于理解的分词系统还处在试验阶段。

基于统计的分词方法：给出大量已经分词的文本，利用统计机器学习模型学习词语切分的规律（称为训练），从而实现对未知文本的切分。例如最大概率分词方法和最大熵分词方法等。随着大规模语料库的建立，统计机器学习方法的研究和发展，基于统计的中文分词方法渐渐成为了主流方法。

主要统计模型：N 元文法模型（N-gram），隐马尔可夫模型（Hidden Markov Model ，HMM），最大熵模型（ME），条件随机场模型（Conditional Random Fields，CRF）等。

现在对于中文分词，分词工具有很多种，比如说：jieba 分词、thulac、SnowNLP等。在这篇文档中,我们通常使用jieba库来进行分词,并且基于python3环境，选择 jieba 分词的理由是其比较简单易学，容易上手，并且分词效果还很不错。jieba 分词支持繁体分词，支持自定义词典，juJieba 提供了三种分词模式：

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；

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。

通过初步分词，能看到有部分新型词语，jieba 分词库不能识别，因此需要自定义词典，将“自定义占道、垃圾处理费、不平等、群租房、装监控”加到自定义词典 Cnewwords.txt 中，然后仅从默认精确模式的分词。

分词结果实例：

二胎准生证办理的咨询

二胎，准生证，办理，的，咨询

（4）去停用词

经过中文分词这一步骤，文本中还含有对文本含义表达无意义的词语，应进行删除，以消除它对文本挖掘工作的不良影响，我们把它叫做停用词，因此需要去除停用词。

停用词是指在信息检索中，为节省存储空间和提高搜索效率，在处理自然语言数据(或文本)之前或之后会自动过滤掉某些字或词，这些字或词即被称为Stop Words(停用词)。这些停用词都是人工输入、非自动化生成的，生成后的停用词会形成一个停用词表。但是，并没有一个明确的停用词表能够适用于所有的工具。甚至有一些工具是明确地避免使用停用词来支持短语搜索的。

去停用词结果示例：

二胎，准生证，办理，的，咨询

二胎，准生证，办理,咨询

2.2.1.5 TF-IDF 算法

在对留言主题进行分词后，需要把这些词语转换成向量。此处采用 TF-IDF 算法，TF-IDF 算法的具体原理为：

（1） 计算词频，即 TF 权重

TF = 某词在文本中出现的次数 （1）

对词频进行标准化：

$$TF = \frac{\text{某词在文本中的出现次数}}{\text{文本的总词数}} \quad (2)$$

（2） 计算 IDF 权重，即你文档频率。需要建立一个语料库，用来模拟语言的使用环境，IDF 越大，此特征性在文本中的分布越集中，说明该分词在区分该文本内容属性能力越强。

$$IDF = \log\left(\frac{\text{语料库的文本总数}}{\text{包含该词的文本数}+1}\right) \quad (3)$$

（3） 计算 TF-IDF 值：

TF - IDF = TF * IDF （4）

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的 TF-IDF。因此，TF-IDF 倾向于过滤掉常见的词语，保留重

要的词语。TF-IDF 算法非常容易理解，并且很容易实现，但是其简单结构并没有考虑词语的语义信息，无法处理一词多义与一义多词的情况。

2．2.1.6 留言分类

生成留言主题的 TF-IDF 权重向量后，根据每个留言的 TF-IDF 权重向量对留言进行分类，本文分别采用朴素贝叶斯分类器（NBC），K-最近进分类器（K-NN）和线性分类支持向量机（LinearSVC），然后计算分类的正确率并进行比较，下面分别对三种分类器进行介绍：

（1）朴素贝叶斯分类器

朴素贝叶斯分类器，顾名思义，是一种分类算法，且借助了贝叶斯定理。另外，它是一种生成模型（generative model），采用直接对联合概率 P(x,c)建模，以获得目标概率值的方法。

贝叶斯分类是一类分类算法的总称，这类算法均以贝叶斯定理为基础，故统称为贝叶斯分类。而朴素朴素贝叶斯分类是贝叶斯分类中最简单，也是常见的一种分类方法。朴素贝叶斯的核心便是贝叶斯公式：P(B|A)=P(A|B)P(B)/P(A) 即在 A 条件下，B 发生的概率。

换个角度：P(类别|特征)=P(特征|类别)P(类别)/P(特征)

而我们最后要求解的就是 P(类别|特征)。

在本题中，我们导入 sklearn.naive_bayes 库的 MultinomialNB 模块，根据给定数据与标签返回正确率的均值，可以得出朴素贝叶斯分类器的正确率 model_nb.score = 0.83548568220101072，正确率较高。

（2）K-最近邻分类器

简单地说，K-近邻算法采用测量不同特征值之间的距离方法进行分类。K-近邻算法具有精度高、对异常值不敏感、无数据输入假定的优点，但是又计算复杂度高，空间复杂度高的缺点。

k-近邻算法（kNN）的工作原理是：存在一个样本数据集合，也称作训练样

本集，并且样本集中每个数据都存在标签，即我们知道样本集中每一数据与所属分类的对应关系。输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据（最近邻）的分类标签。一般来说，我们只选择样本数据集中前 k 个最相似的数据，这就是 k-近邻算法中 k 的出处，通常 k 是不大于 20 的整数。最后，选择 k 个最相似数据中出现次数最多的分类，作为新数据的分类。

输入：训练集 $D=\{(x1,y1),(x2,y2),\cdots,(xN,yN)\}$

根据给定的距离度量，在训练集 D 中找出与 x 最近邻的 k 个点，涵盖这涵盖这 k 个点的 x 的领域记作 Nk(x)，在 Nk(x)中根据分类决策规则（如多数表决）决定 x 的类别 y。

本文导入 KNeighborsClassifier，可以得出 K-最近邻分类器的正确率 model_knn.score = 0.47669848399775405，可见 KNN 的正确率大大不如朴素贝叶斯的正确率。

（3）线性分类支持向量机

支持向量机(Support Vector Machine, SVM)是曾经打败神经网络的分类方法，从 90 年代后期开始在很多领域均有举足轻重的应用，近年来，由于深度学习的兴起，SVM 的风光开始衰退，但是其仍然不失为一种经典的分类方法。SVM 最初由 Vladimir N. Vapnik 和 Alexey Ya. Chervonenkis 于 1963 年提出，之后经过一系列改进，现今普遍使用的版本由 Corinna Cortes 和 Vapnik 于 1993 年提出，并在 1995 年发表[1]。深度学习兴起之前，SVM 被认为是机器学习近几十年来最成功、表现最好的方法。SVC=Support Vector Regression.就是支持向量机用于回归分析，本文的 model_svc.score = 0.84222346996069619，可以看出比朴素贝叶斯正确率高一些。

2.2.1.7 模型评估

经过上述三种分类器对文本分类后，下面运用 F-SCORE 进行模型评估，需要涉及到准确率（Precision 查准率）召回率（Recall 查全率）和 F 的概念：

（1）实际上非常简单，精确率是针对我们预测结果而言的，它表示的是预测为正的样本中有多少是真正的正样本。那么预测为正就有两种可能了，一种就是把正类预测为正类(TP)，另一种就是把负类预测为正类(FP)，也就是

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

（2）而召回率是针对我们原来的样本而言的，它表示的是样本中的正例有多少被预测正确了。那也有两种可能，一种是把原来的正类预测成正类(TP)，另一种就是把原来的正类预测为负类(FN)。

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

其实就是分母不同，一个分母是预测为正的样本数，另一个是原来样本中所有的正样本数。

我们当然希望检索的结果 P 越高越好，R 也越高越好，但事实上这两者在某些情况下是矛盾的。比如极端情况下，我们只搜出了一个结果，且是准确的，那么 P 就是 100%，但是 R 就很低；而如果我们把所有结果都返回，那么必然 R 是 100%，但是 P 很低。 因此在不同的场合中需要自己判断希望 P 比较高还是 R 比较高。

下面分别展示三种分类模型的拟合结果：

1）朴素贝叶斯：

	precision	recall	f1-score	support
交通运输	0.85	0.61	0.71	109
劳动和社会保障	0.82	0.91	0.86	394
卫生计生	0.90	0.73	0.81	164
商贸旅游	0.80	0.79	0.79	221

城乡建设	0.80	0.86	0.83	396
教育文体	0.86	0.89	0.88	313
环境保护	0.91	0.82	0.86	184
avg / total	0.84	0.84	0.83	1781

```
[[ 67   9   1  13  15   2   2]
 [  1 359   6   6   6  15   1]
 [  0  28 120   5   6   4   1]
 [  2  11   2 174  26   5   1]
 [  6  16   2   8 340  15   9]
 [  2  13   2   8  10 278   0]
 [  1   3   1   3  23   3 150]]
```

2）K 最近邻分类器

	precision	recall	f1-score	support
交通运输	0.69	0.49	0.57	109
劳动和社会保障	0.80	0.47	0.59	394
卫生计生	0.28	0.66	0.39	164
商贸旅游	0.83	0.35	0.50	221
城乡建设	0.36	0.79	0.49	396
教育文体	0.92	0.31	0.47	313

环境保护	0.93	0.08	0.14	184
avg / total	0.69	0.48	0.47	1781

```
[[ 53   3  15   2  35   1   0]
 [  5 186  69   2 130   2   0]
 [  2  15 108   0  38   1   0]
 [  7   5  38  78  92   1   0]
 [  6  11  58   5 312   3   1]
 [  3   9  46   5 152  98   0]
 [  1   3  52   2 111   1  14]]
```

3）线性分类支持向量机

	precision	recall	f1-score	support
交通运输	0.86	0.84	0.85	109
劳动和社会保障	0.87	0.88	0.88	394
卫生计生	0.86	0.82	0.84	164
商贸旅游	0.81	0.76	0.78	221
城乡建设	0.79	0.86	0.82	396
教育文体	0.86	0.86	0.86	313

环境保护	0.90	0.79	0.84	184
avg / total	0.84	0.84	0.84	1781

```
[[ 92   0   1   6   6   2   2]
 [  2 348   9   4  16  15   0]
 [  1  13 135   5   4   4   2]
 [  5   5   5 168  30   6   2]
 [  6  13   2  12 341  13   9]
 [  1  18   2  11   9 270   2]
 [  0   3   3   2  26   4 146]]
```

由上述结果可以看出，朴素贝叶斯模型和线性支持向量机的查准率和召回率就较高，Knn 模型查准率召回率低很多，查准率和召回率分别为 0.69 和 0.48，所以优先考虑朴素贝叶斯模型和线性支持向量机，平均查准率和召回率都达到 84%，但是支持向量机的算法过程复杂度比朴素贝叶斯都高，因此本题优先选择朴素贝叶斯分类器对文本数据进行分类。

2.2.2 问题二分析方法与过程

2.2.2.1 基于 K-means 模型的量化分析

基于语义网络的评论分析进行初步数据感知后，我们以统计学习的角度出发，对每条评论的主题进行量化表示。本文针对此项问题运用 K-mean 算法为基础，加以改良，用以提取热点问题并进行分类。

文本分析的模型可分为监督学习模型（Naive Bayse、Wordscores、Random Forest 等）、非监督学习模型（Wordfish、Correspondence Analysis、Topic Models 等）、半监督学习模型（Newsmap、Latent Semantic Scaling 等）监督学

习可以比较有效地对结果进行控制，但是需要花费大量的时间和工作来准备学习数据，是一种成本较高的方式。无监督学习虽然成本较低，但是对结果操控的操控性略低。

量化文本分析（Quantitative Text Analysis, QTA）是指在社会科学研究中利用计算机技术来自动并且系统地处理大量文本数据的方法。随着自然语言处理等技术的发展，越来越多的政治学者已经注意到 QTA 方法在社会科学中的应用前景，并且利用这种方法做出了一系列研究成果。

2.2.2.2 K-means 模型介绍

K-means 算法(Lloyod,1982)是简单而又有效的统计聚类算法，使机器能够将具有相同属性的样本归置到一块儿。与分类不同，对于一个分类器，通常需要告诉它“这个样本被分成哪些类”这样一些标签，在最理想情况下，一个分类器会从所得到的训练集中进行“学习”，我们将这种提供训练的过程称为“监督学习”。但是在聚类下，我们并不关心某一类是什么，我们的目的是想将相似的样本归置在一起，这样，一个聚类算法通常只要知道该如何计算样本间的相似度并将相似样本归并到一起就可以操作了，因此聚类通常并不需要使用训练数据进行学习，这在机器学习中被称作“无监督学习”。K-means 算法就是这种用于统计的无监督聚类技术。

2.2.2.3 K-means 模型原理

我们采用 k -均值聚类方法，随机选取以预处理后的文本向量的 k 个点作为质心，对于数据中选取的每个向量，我们计算每个向量到质心的距离(欧几里得距离)，然后将这个点划分到最近的聚类中心，从而形成 K 个聚类。我们重新计算每个点，重复上述步骤数十次，直到质心位置不再变化或达到我们设定的 100 次迭代次数，其数学公式为:

$$\mathrm{d} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

后计算 a（I）=average(I 向量与所属所有簇中其他点的距离)

计算 b（I）= min(从 I 向量到不在自己簇中的所有点的平均距离)

则 I 向量等高线系数为:

$$S(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$

本文使用的 K-means 聚类算法与原始算法有改进之处。

2.2.2.4 原始 K-means 算法与本文所用改良后 K-means 算法的对比

K-means 算法操作简单、运算速度较快，能够有效处理中小型数据集。但同时 K-means 算法也有不足之处，包含以下几点：

(1) 聚类结果不确定

K-means 算法初始聚类中心是随机选择的，初始中心点选取的好坏会导致最终聚类效果。选取不同初始聚类中心，会使得最终聚类得到的类簇发生变化。除此之外，K-means 算法一般采用准则函数为目标函数，准则函数中只存在一个全局最小值和 N 个极小值，这使得算法运算过程中，会陷入局部极小值，导致最终得到的不是全局最优解。

(2) 聚类个数不确定

K-means 算法中 K 表示聚簇个数，K 的取值决定聚类结果。K 值的选取需要根据实际的需求来确定，但通常情况下我们并不知道需将数据集聚为多少个类簇最合适，所以针对 K 值的选取依然有待解决。

(3) 数据量大、算法时间复杂度较高

K-means 算法的计算过程是一个不断迭代的过程，为寻找合适的聚类中心，需要不断的计算和调整才能对数据对象进行有效的聚类。这个过程中反复进行大量的对象间距离的计算，所以 K-Means 聚类过程会消耗大量时间，降低聚类运算效率。

本文采用高斯随机初始中心点方法，传统 K-means 算法在初始化 K 个中心点时使用数据集的前 K 个样本作为中心点或使用默认的随机化方法初始中心点，这里采用高斯随机化方法从数据集中取 K 个点作为中心点，可使得聚簇内更加紧密。从而使得到的结果更加真实可信。

2.2.2.5 运用 K-means 模型进行量化分析的实现过程

在本文有关热点留言的研究中，即对留言中的热点主题进行挖掘（数据预处理过程的原理）在取得了足够可信的向量化留言后我们使用高斯随机初始中心点方法进行 K-means 聚类数据处理得到集群内向量最多的 5 个集群即是我们所要的 5 个热点留言。本文运用 MATLAB R2019a 软件编写 K-means 聚类模型的算法主题个数采用统计语言模型中常用标准困惑度来选取，令 K=15。

2.2.3 问题三分析方法与过程

2.2.3.1 答复意见评价的实现

我们拟定从文本情感分析的角度进行研究处理，按照处理文本的粒度不同，情感分析可分为词语级、短语级、句子级、篇章级这四个层次的研究任务。词语的情感分析是文本情感分析的基础，同时也作为句子和篇章情感分析的前提。基于词的情感分析研究主要有情感词抽取、判定、以及情感语料库与情感词典的研究等；句子的情感分析主要是综合情感词的分析结果，也可视为短篇章的情感分析；而篇章的情感分析则是综合文本各粒度的情感分析结果，再结合上下文和领域知识库做出极性判断。本题所采用的角度应当为篇章级这个层次的研究任务。

在处理每段文字时由于段落过于冗长，其中不乏无实际意义的短语，所以我们首先需要对文本进行预处理，去重、去敏、去停顿词。而后我们主要需要关注其中所携带的情感。

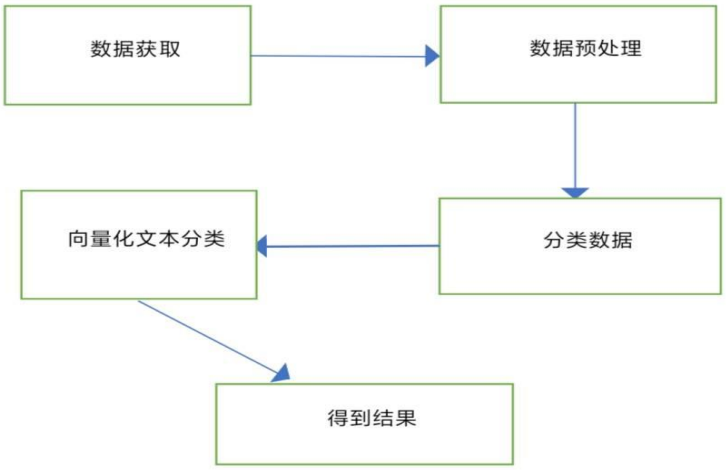
文本情感分析旨在分析出文本中针对某个对象的评价的正负面，比如[中国抗疫工作优秀]就是一个正面评价。情感分析主要有五个要素，（entity/实体，aspect/属性，opinion/观点，holder/观点持有者，time/时间），其中实体和属性合并称为评价对象（target）。情感分析的目标就是从非结构化的文本评论中

抽取出这五个要素。

词语的情感极性判别主要有基于语料库、基于词典、基于词向量这三种方法。要实现短文本的相似度分析，通过关键词的匹配来实现相似度分析是比较困难的，我们认为采用词向量的方法较为出众。

（1）基于词向量的方法

深度学习(Deep Learning)中一般用到的词向量是用词向量(Word Embedding)或布式表达方法（Distributed Representation）所表示的一种低维实数向量。



（2）文本向量化的实现

Word2vec 方法介绍：

Word2vec 是 Google 开源的一款将词表征为实数值向量的高效工具，采用的模型有 CBOW（词袋模型）和 Skip-Gram 两种。Word2vec 通过训练，可以把对文本内容的处理简化为 K 维向量空间中的向量运算，而向量空间上的相似度可以用来表示文本语义上的相似度。因此， Word2vec 输出的词向量可以被用来做很多 NLP 相关的工作，比如聚类、找同义词、词性分析等等。使用谷歌开源的 Word2vec 训练文本后可以得到每个词的向量值。也可以调用相应的包进行短语训练，或者使用多种计算相似度的方法。

（3）对向量化文本的处理

通过使用 CNN 算法对向量化文本进行处理并得到 CNN 的各层运行结果，其中

二层的列表来表示词向量相似度矩阵，经过卷积层和池化层的循环得出 CNN 输出值，这时候是一维的值，最后简单的对这几个输出值求一个平均值就是该项目得出的相似度值。CNN 算法简介如下。

20 世纪 60 年代，Hubel 和 Wiesel 在研究猫脑皮层中用于局部敏感和方向选择的神经元时发现其独特的网络结构可以有效地降低反馈神经网络的复杂性，继而提出了卷积神经网络（Convolutional Neural Networks-简称 CNN）。现在，CNN 已经成为众多科学领域的研究热点之一，特别是在模式分类领域，由于该网络避免了对图像的复杂前期预处理，可以直接输入原始图像，因而得到了更为广泛的应用。 K.Fukushima 在 1980 年提出的新识别机是卷积神经网络的第一个实现网络。随后，更多的科研工作者对该网络进行了改进。其中，具有代表性的研究成果是 Alexander 和 Taylor 提出的“改进认知机”，该方法综合了各种改进方法的优点并避免了耗时的误差反向传播。

其图解如下：

$$h_{w,b}(x)\begin{cases} a \\ b \\ c \\ +1 \end{cases}$$

其表达式如下：

$$h_{W,b}(X) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$$

如此一层一层的加上去，最终就形成了深度神经网络。而本文的卷积神经网络就是一种深度神经网络。卷积神经网络 CNN 的结构一般包含这几个层：

1）输入层：用于数据的输入

2）卷积层：使用卷积核进行特征提取和特征映射

3）激励层：由于卷积也是一种线性运算，因此需要增加非线性映射

4）池化层：进行下采样，对特征图稀疏处理，减少数据运算量。

5）全连接层：通常在 CNN 的尾部进行重新拟合，减少特征信息的损失

2.3 结果分析

2.3.1 Matlab K-mean 算法核心程序

```
function centroids = computeCentroids(X, idx, K)
%计算聚类中心
[~,n] = size(X);
centroids = zeros(K, n);
for i=1:K
    indices = idx == i;
    for j=1:n
        centroids(i, j) = sum(X(:, j) .* indices) / sum(indices);
    end
end
end
function idx = findClosestCentroids(X, centroids)
%对各向量进行分类
K = size(centroids, 1);
idx = zeros(size(X,1), 1);
for i=1:length(X)
    distance = inf;
    for j=1:K
        kDist = norm(X(i, :) - centroids(j, :));
        if (kDist < distance)
```

```
            distance = kDist;
            idx(i) = j;
        end
    end
end
end
function centroids = kMeansInitCentroids(X, K)
%选择聚类中心
centroids = zeros(K, size(X, 2));
randidx = randperm(size(X, 1));
centroids = X(randidx(1:K), :);
end
function [centroids, idx] = runkMeans(X, initial_centroids, ...
                                      max_iters)
%计算聚类中心和分类
[m,n] = size(X);
K = size(initial_centroids, 1);
centroids = initial_centroids;
idx = zeros(m, 1);
for i=1:max_iters
    fprintf('K-Means 迭代 %d/%d...\n', i, max_iters);
    idx = findClosestCentroids(X, centroids);
```

```
    centroids = computeCentroids(X, idx, K);
end
end
clc;
clear;
doc xlsread
data=xlsread('a.xlsx')
k=15;%设置聚类数目
max_iters=100; %设置迭代次数；
initial_centroids = kMeansInitCentroids(data, k);%聚类中心初始化
[centroids, idx] = runkMeans(data, initial_centroids, max_iters);%运行k均值，返回聚类中心和类别

end
[idx2,centroids2]=kmeans(data,k);
```

2.3.2 python 程序

```
import pandas as pd
import re
import jieba
data =  pd.DataFrame(pd.read_excel('附件2.xlsx',header=None,index_col=0))
```

```
data.columns = ['留言用户','留言主题','留言时间','留言详情','一级标签']


a = data[data['一级标签'] == '城乡建设']

b = data[data['一级标签'] == '环境保护']

c = data[data['一级标签'] == '交通运输']

d = data[data['一级标签'] == '教育文体']

e = data[data['一级标签'] == '劳动和社会保障']

f = data[data['一级标签'] == '商贸旅游']

g = data[data['一级标签'] == '卫生计生']

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


data_new = data_new.drop_duplicates(subset='留言主题',keep='first')  #去重


data_qumin = data_new['留言主题'].apply(lambda x:
re.sub('[^\u4E00-\u9FD5]+','',x))#去敏

l

jieba.load_userdict('Cnewwords.txt')  #添加自定义分词字典

data_cut = data_qumin.apply(lambda x: jieba.lcut(x))  #分词


stopWords = pd.read_csv('stopword.txt',encoding =
'GB18030',sep='haha',header=None,engine='python')
```

```
stopWords = [', ','县','A','J','K8','B','？','！','J4','D','M2','区','市'] + list(stopWords.iloc[:,0])


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


labels = data_new.loc[data_after_stop.index,'一级标签']


from matplotlib import pyplot as plt
from wordcloud import WordCloud
import itertools


num = pd.Series(list(itertools.chain(*list(data_after)))).value_counts()


pic = plt.imread('duihuakuan.jpg')
# word_fre = {}
# data_after_stop[labels == '交通运输']
# for i in data_after_stop[labels == '交通运输']:
#     for j in i:
#         if j not in word_fre.keys():
#             word_fre[j] = 1
#         else:
#             word_fre[j] += 1
```

```
wc = WordCloud(mask=pic,
background_color='white',font_path=r'C:\Windows\Fonts\simhei.ttf')
# wc.fit_words(word_fre)
# plt.imshow(wc)
# plt.show(wc)
wc2 = wc.fit_words(num)
plt.imshow(wc2)
plt.axis('off')
plt.show()



tmp = data_after.apply(lambda x:' '.join(x))
from sklearn.feature_extraction.text import
CountVectorizer,TfidfTransformer
cv = CountVectorizer().fit(tmp)
cv_data = cv.transform(tmp)
# cv.vocabulary_
cv_data.toarray().shape                    #文本向量化



from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split


cv_train,cv_test,y_train,y_test = train_test_split(
    cv_data,data_new['一级标签'],
    test_size=0.2,random_state=123
)
model_nb = MultinomialNB().fit(cv_train,y_train)
model_nb.score(cv_test,y_test) #朴素贝叶斯分类器√


model_knn = KNeighborsClassifier().fit(cv_train,y_train)
model_knn.score(cv_test,y_test) # K 最近邻分类器


model_svc = LinearSVC().fit(cv_train,y_train)
model_svc.score(cv_test,y_test)  #线性分类支持向量机


#模型评估
from sklearn.metrics import classification_report, confusion_matrix
y_pre_nb = model_nb.predict(cv_test)
print(classification_report(y_true=y_test,y_pred=y_pre_nb))
print(confusion_matrix(y_true=y_test,y_pred=y_pre_nb))
y_pre_knn = model_knn.predict(cv_test)
```

```
print(classification_report(y_true=y_test,y_pred=y_pre_knn))

print(confusion_matrix(y_true=y_test,y_pred=y_pre_knn))

y_pre_svc = model_svc.predict(cv_test)

print(classification_report(y_true=y_test,y_pred=y_pre_svc))

print(confusion_matrix(y_true=y_test,y_pred=y_pre_svc))


#预处理

import pandas as pd

import re

import jieba

data = pd.DataFrame(pd.read_excel('附件 3.xlsx',header = None,index_col
= 0))

data.columns = ['留言用户','留言主题','留言时间','留言详情','反对数','
点赞数']

data_new = data.drop_duplicates(subset='留言主题',keep='first')  #去重

data_qumin = data_new['留言主题'].apply(lambda x:
re.sub('[^\u4E00-\u9FD5]+','',x))#去敏

data_cut = data_qumin.apply(lambda x: jieba.lcut(x))  #分词

stopWords = pd.read_csv('stopword.txt',encoding =
'GB18030',sep='haha',header=None,engine='python')

stopWords = ['县','区','市'] + list(stopWords.iloc[:,0])

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

print(data_after)
```

```
#文本向量化
tmp = data_after.apply(lambda x:' '.join(x))
from sklearn.feature_extraction.text import
CountVectorizer,TfidfTransformer
cv = CountVectorizer().fit(tmp)     #将文本中的词转换成词频矩阵
print(cv)
cv_data = cv.transform(tmp)         #计算某个词出现的次数
print(cv_data)
print(cv.vocabulary_)      #字典
word = cv.get_feature_names()       #获取词袋中所有文本关键词
print(word)
cv_data.toarray()                   #转成数组
function centroids = computeCentroids(X, idx, K)
%计算聚类中心
[~,n] = size(X);
centroids = zeros(K, n);
for i=1:K
    indices = idx == i;
    for j=1:n
        centroids(i, j) = sum(X(:, j) .* indices) / sum(indices);
    end
```

```
end
end
function idx = findClosestCentroids(X, centroids)

%对各向量进行分类
K = size(centroids, 1);
idx = zeros(size(X,1), 1);
for i=1:length(X)
    distance = inf;
    for j=1:K
        kDist = norm(X(i, :) - centroids(j, :));
        if (kDist < distance)
            distance = kDist;
            idx(i) = j;
        end
    end
end
end
function centroids = kMeansInitCentroids(X, K)
%选择聚类中心
centroids = zeros(K, size(X, 2));
randidx = randperm(size(X, 1));
```

```
centroids = X(randidx(1:K), :);
end
function [centroids, idx] = runkMeans(X, initial_centroids, ...
                                      max_iters)
%计算聚类中心和分类
[m,n] = size(X);
K = size(initial_centroids, 1);
centroids = initial_centroids;
idx = zeros(m, 1);
for i=1:max_iters
    fprintf('K-Means 迭代 %d/%d...\n', i, max_iters);
    idx = findClosestCentroids(X, centroids);
    centroids = computeCentroids(X, idx, K);
end
end
clc;
clear;
doc xlsread
data=xlsread('a.xlsx')
k=15;%设置聚类数目
max_iters=100; %设置迭代次数;
initial_centroids = kMeansInitCentroids(data, k);%聚类中心初始化
```

[centroids, idx] = runkMeans(data, initial_centroids, max_iters);%运行 k 均值，返回聚类中心和类别

end

[idx2,centroids2]=kmeans(data,k);

结果显示：

运用 drop duplicates 对“留言主题”这一列重复的数据进行去重处理，得到 8905 行数据，可见去重了 305 条数据，可见重复率达到 3.312%。某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的 TF-IDF。因此，TF-IDF 倾向于过滤掉常见的词语，保留重要的词语。基于语义网络的评论分析进行初步数据感知后，我们以统计学习的角度出发，对每条评论的主题进行量化表示，最终实现了关于留言内容的分类以及热点问题挖掘。在对向量化的文本进行 CNN 算法运行后对得到的情感分析进行汇总以其强烈程度作为系数例如 0 代表否定 1 代表肯定，以其大小来对大量文本进行判定。结果显示如下。

留言编号	时间	人群/地点	问题核心	热度排名
286572	2018/10/27	梅溪湖	地铁	1
353426	2009年-2019年	工人	社保	2
289408	2017年12月-2018年初	学生	补贴	3
316619	2019/5/14	A市	5G	4
313964	2019/4/23-2019/4/26	学生	驾驶证	5

3.结论

本文通过对处理过的各个网络问政平台的文本留言结论数据利用K-mean算法为基础且加以改良等方法建立多种数据挖掘模型，得到了具有一定价值的结果，实现了对文本评论数据的情感倾向性分析，以及一定程度上的对包括城乡建设、安全生产、事故处理等在内的更细节的文本信息的挖掘与认识，而这些结果对于政府相关部门具有一定性的指导意义。比如，某一时段内群众集中反映的某一问题可称为热点问题，如“某地区某些小区多位业主多次反映入夏以来小区楼下烧烤店深夜经营导致噪音和油烟扰民”，“某校强制学生去定点企业进行实习”。及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务的效率、质量。但是，研究与分析结果显示，总体效果并没有达到预期，比如所谓的情感倾向性分析，由于现如今中文文本挖掘模型不足，而中文的语言结构又必然导致文本留言分析存在缺陷，再加上留言数据本身所具有的问题，其分析结果事实上与真实结果有一定程度上的出入，这便会致使针对相关部门留言答复意见给出的评价方案也存在一定程度上的偏差。这便是需要我们在接下来对中文文本数据的研究过程中进一步学习、探讨的地方。

4.参考文献

[1]李 航.统计学习方法[M].北京:清华大学出版社,2012.3

[2] 周志华.机器学习[M].北京:清华大学出版社,2016.1

[3] PeterHarrington. 机器学习实战[M]. 北京:人们邮电出版社,2013,6

[4]Cao Juan, Xia Tian, Li Jin Tao, A density method for adaptive LDA model selection[J], Neurocomputing2009(72):1775-1781