第八届“泰迪杯”数据挖掘挑战

——“智慧政务”中的文本挖掘应用

摘要

随着信息时代的发展，互联网的应用越来越广泛，政府通过互联网做宣传、做决策，以达到取之于民，用之于民。从而实现科学决策、民主决策，真正做到全心全意为人民服务。随着网络的日益普及，互联网在中国民众的政治、经济和社会生活中扮演着日益重要的角色，中国公民通过它行使知情权、参与权、表达权和监督权。这也使产生很多网络数据需要处理和归类。

本文基于这类问题做了综合评价模型。

针对问题一，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，所以请采用合适的算法建立关于留言内容的一级标签分类模型，以便节省大量的人力物力，帮助问政平台快速处理留言问题，方便人们的生活。因为数据过多，现将数据进行概率抽样，再用 KNN 方法对数据进行分类。

针对问题二，挖掘热点问题，及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。将某一时段内反映特定地点或特定人群问题的留言进行归类并排名前 5 的热点问题，用简单地 EXEL 表就可以解决问题。

针对问题三，对留言的答复意见，从答复的相关性、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，根据评价留言答复处理体系，得到一系列数值，根据从大到小分为三个等级，分别为：优秀，良好，一般。

关键词：KNN 算法，热点挖掘，TOPSIS 综合评价

Abstract

With the development of the information age, the application of the Internet is more and more extensive. The government makes propaganda and decision-making through the Internet, so as to obtain and use it for the people. So as to realize scientific and democratic decision-making and truly serve the people wholeheartedly. With the increasing popularity of the Internet, the Internet plays an increasingly important role in the political, economic and social life of the Chinese people, through which Chinese citizens exercise the right to know, participate, express and supervise. This also makes a lot of network data need to be processed and classified.

Based on this kind of problems, this paper makes a comprehensive evaluation model.

In view of problem 1, most e-government systems still rely on manual processing according to experience, which has the problems of heavy workload, low efficiency and high error rate. Therefore, please use the appropriate algorithm to establish a level-1 label classification model for message content, so as to save a lot of human and material resources, help the platform to deal with message problems quickly and facilitate people's life. Because there are too many data, the data are sampled by probability and classified by KNN.

In view of question 3, for the reply to the message, a set of evaluation scheme is given for the quality of the reply from the perspective of relevance, integrity and interpretability. According to the reply processing system of the evaluation message, a series of values are obtained, which are divided into three grades from large to small: excellent, good and general.

Key words: KNN algorithm, hotspot mining, TOPSIS comprehensive evaluation

目录

1. 问题重述

1.1 问题背景

近年来，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道，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挑战。同时，随着大数据、云计算、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的推动作用。

附件给出了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。请利用自然语言处理和文本挖掘的方法解决下面的问题。

1.2 问题概述

1、群众留言分类

在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系（参考附件 1 提供的内容分类三级标签体系）对留言进行分类，以便后续将群众留言分派至相应的职能部门处理。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。请根据附件 2 给出的数据，建立关于留言内容的一级标签分类模型。

通常使用 F-Score 对分类方法进行评价：

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i},$$

其中 P_i 为第 i 类的查准率,R_i 为第 i 类的查全率。

2、热点问题挖掘

某一时段内群众集中反映的某一问题可称为热点问题，如“XXX 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”。及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。请根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，定义合理的热度评

价指标，并给出评价结果，按表 1 的格式给出排名前 5 的热点问题，并保存为文件“热点问题表.xls”。按表 2 的格式给出相应热点问题对应的留言信息，并保存为“热点问题留言明细表.xls”。

3、答复意见的评价

针对附件 4 相关部门对留言的答复意见，从答复的相关性、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，并尝试实现。

2. 问题分析

2.1 问题一

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，所以请采用合适的算法建立关于留言内容的一级标签分类模型，以便节省大量的人力物力，帮助问政平台快速处理留言问题，方便人们的生活。因为数据过多，现将数据进行概率抽样，再用 KNN 方法对数据进行分类。

2.2 问题二

挖掘热点问题，及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。将某一时段内反映特定地点或特定人群问题的留言进行归类并排名前 5 的热点问题，用简单地 EXEL 表就可以解决问题。

2.3 问题三

对留言的答复意见，从答复的相关性、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，并尝试实现。

3. 挖掘目标与数据预处理

3.1 挖掘背景

随着网络的发展，大数据的应用我们正处于并将长期处于信息时代，在这个

信息爆炸的时代，每天就会产生将近百亿条数据，而在这数据的海洋背后，是数以万计的数据分析师的辛劳才会使呈现在我们眼前的数据是如此清晰有条理。在这近百亿条数据中，又有几十亿的文本数据，这个时候文本挖掘就对这些数据起到了决定性作用，它可以帮助我们提取文本数据中的有效信息。

所谓文本挖掘也被称为文字探勘、文本数据挖掘等，大致相当于文字分析，一般指文本处理过程中产生高质量的信息。高质量的信息通常通过分类和预测来产生，如模式识别。文本挖掘通常涉及输入文本的处理过程，产生结构化数据，并最终评价和解释输出。'高品质'的文本挖掘通常是指某种组合的相关性，新颖性和趣味性。典型的文本挖掘方法包括文本分类，文本聚类，概念/实体挖掘，生产精确分类，观点分析，文档摘要和实体关系模型。

近年来，随着微信、微博、市长信箱、阳光热线等网络问政平台逐步成为政府了解民意、汇聚民智、凝聚民气的重要渠道。政府通过互联网做宣传、做决策，以达到取之于民，用之于民。从而实现科学决策、民主决策，真正做到全心全意为人民服务。随着网络的日益普及，互联网在中国民众的政治、经济和社会生活中扮演着日益重要的角色，中国公民通过它行使知情权、参与权、表达权和监督权。

这也导致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挑战。同时，随着大数据、云计算、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的推动作用。

3.2 挖掘目标

本次实验希望可以通过已给数据，完成对于以下的问题的分析：

1、根据附件 2 给出的数据完成对于留言内容的分类，以提高分类的准确率，提高工作效率，以解决分类工作量大和出错率高等问题。

2、请根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，并给出评价结果定义合理的热度评价指标，对热点问题进行提取，了解群众对于哪类问题比较关心，方便日后的群众工作。

3、针对附件 4 相关部门对留言的答复意见，对于每一条留言工作人员给予了答复，但是答复质量参差不齐，我们需要对这些答复从不同角度（如：相关性、完整性、可解释性等）进行评价。

3.3 数据预处理

检查数据集的各变量概况，对数据结构有一个整体的把握，获取变量详情；本题用到的数据量非常大，需要检查数据稀疏性，以更好的了解数据特征，用于选择合适的数据分析算法进行进一步信息挖掘，检查数据是否有缺失值，避免缺失值对后续分析造成影响，数据经过这些预处理后，基本已经排除内部不良数据影响分析结果的可能，就可以进行分析了。

4. 数据挖掘方法及解释

4.1 分层抽样

分层抽样法，也叫类型抽样法。就是将总体单位按其属性特征分成若干类型或层，然后在类型或层中随机抽取样本单位。分层抽样的特点是：由于通过划类分层，增大了各类型中单位间的共同性，容易抽出具有代表性的调查样本。该方法适用于总体情况复杂，各单位之间差异较大，单位较多的情况。

这种方法的优点是，样本的代表性比较好，抽样误差比较小。缺点是抽样手续较简单随机抽样还要繁杂些。定量调查中的分层抽样是一种卓越的概率抽样方式，在调查中经常被使用。

4.2 KNN 算法

KNN 将测试集的数据特征与训练集的数据进行特征比较，然后算法提取样本集中特征最近邻数据的分类标签，即 KNN 算法采用测量不同特征值之间的距离的方法进行分类。KNN 的思路很简单，如果一个样本在特征空间中的 K 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。也就是说，该方法在定类决策上只依据最邻近的一个或者几个样本的类别来

决定待分样本所属的类别。KNN 具有精度高、无数据输入假定、简单有效的特点，所以在本次建模中，选择 KNN 算法对数据建立分类模型。

4.3 TOPSIS 综合评价法

TOPSIS 综合评价法是进行多个对象选择排序的方法，其通过计算各对像与最优方案、最劣方案的相对距离进行排序，它用于解决多对象多属性决策问题，根据有限个评价对象与理想化目标的接近程度进行排序，适用于多项目标、对多个方案进行比较选择的分析方法。

5. 数据挖掘内容及过程

5.1 群众留言分类

5.1.1 建模过程及结果

1、准备安装包

```
install.packages("corpus")
install.packages("Rwordseg")
install.packages("tmcn")
install.packages("tm")
install.packages("HMM")
library(HMM)
library(Rwordseg)
library(tmcn)
library(tm)
library(NLP)
library(corpus)
library(class)
library(sampling)
```

图 1

2.数据读取

```
> d<-read.csv("D:/cd/rs.csv")
> d$留言详情<-as.character(d$留言详情)
> d$一级标签<-as.character(d$一级标签)
> length(d)
[1] 6
> table(d$一级标签)

      城乡建设       环境保护       交通运输       教育文体 劳动和社会保障
          2009            938            613           1589           1969
      商贸旅游       卫生计生
          1215            877
```

图 2

3、分层抽样

因为数据过多，所以选择分层抽样的方法，每层抽取 300 个样本。

```
> head(sub)
  一级标签 ID_unit Prob Stratum
1 城乡建设       1    1       1
2 城乡建设       2    1       1
3 城乡建设       3    1       1
4 城乡建设       4    1       1
5 城乡建设       5    1       1
6 城乡建设       6    1       1
```

图 3

4、分词操作

去除文本中的数字，删除停止词。

```
temp<-gsub('[0-9]','',d$留言详情)
segwords1<-segmentCN(temp)
stopwords<-stopwordsCN()
segwords2<-lapply(segwords1,removeWords,stopwords)
```

图 4

5、文本的特征提取

```
> corpus_d<-Corpus(VectorSource(segwords2))
> dtm<-DocumentTermMatrix(corpus_d,control=list(wordLengths=c(2,Inf)))
> dtm
<<DocumentTermMatrix (documents: 2100, terms: 7265)>>
Non-/sparse entries: 20983/15235517
Sparsity           : 100%
Error in nchar(Terms(x), type = "chars") :
  invalid multibyte string, element 1
> dtm_matrix<-as.matrix(dtm)
> dim(dtm_matrix)
[1] 2100 7265
```

图 5

6、对矩阵建立统计模型

选择 2/3 的数据作为训练样本，1/3 的数据作为测试样本。

7、分类效果

通过混淆矩阵来看分类效果。

表 1

	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	卫生计生
城乡建设	79	2	4	6	2	0	0
环境保护	0	85	0	0	1	2	0
交通运输	0	4	78	0	1	0	0
教育文体	1	5	8	80	0	3	0
劳动和社会保障	9	0	0	8	89	1	0
商贸旅游	5	0	0	3	5	91	0
卫生计生	6	4	10	3	2	3	100

5.1.2 模型评估

使用 F-Score 对分类方法进行评价：

1、计算查准率

城乡建设：0.79，环境保护：0.85，交通运输：0.78，教育文体：0.80，劳动和社会保障：0.89，商贸旅游：0.91，卫生计生：1

2、计算查全率

城乡建设：0.77，环境保护：0.77，交通运输：0.77，教育文体：0.77，劳动和社会保障：0.77，商贸旅游：0.77，卫生计生：0.77

3、带入公式

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

其中 P_i 为第 i 类的查准率，R_i 为第 i 类的查全率，n=7.

计算得 F_1=0.811，可以看出模型良好，本分类模型有效。

5.2 热点问题挖掘

5.2.1 热点问题理解

社会热点应该是“社会问题”。什么是“问题”？按照决策学的解释：问题是理想与现实的差距,即人们对目前某一事物的现状感到不满意,需要加强管理,达到理想状态。社会问题就是某一方面存在的状况已经令人感到不解决就会出现矛盾，甚至引起社会混乱，导致社会停滞不前或社会严重失调。不是任何一个新事物的出现都会成为社会问题，不是任何热点都会成为命题的依据。正面的、个体的、局部的、意识形态的都不可能成为社会热点，不可能以此来命题。所以，社会热点对于应试者来说，不是新鲜事物，不是没有定性的问题，不是务虚的意识形态问题。既然是热点，就应该是已经被人认识和了解的矛盾，是积累到一定程度的“热”问题。

5.2.2 数据处理步骤

1、排序数据

从 4326 个数据中心挑选出有用的数据，本组认为无论点赞数还是反对数，都是说明有人在关注着此类问题,故将附件 3 中反对数和点赞数求和得出总关注数。再将各个问题的总关注数进行降序排列，得到最高和最低总关注数。

2、筛选数据

因为社会热点应该是“政府问题”。即某一方面存在的问题，必须是政府所要着力解决的，而且也必须与政府的各个职能部门紧密相关。所以，如果发生的某一社会热点与政府的关联性不大，或是不是政府所要面临解决的问题，一般就不会成为申论的命题范围。故由右上一步骤的结果，可将总关注数为 0，没有得到大众的关注的问题排除；剩下主要问题 33 个。

3、数据分类

社会热点应该具有普遍性。即所存在的社会热点问题不是刚刚出现，也不是还没有出现，是已经社会各界的广泛议论，对于该问题的解决，政府已经出台了一些措施，人们对此出现，而且问题已经积累到比较严重的地步，已经引起了政府高度重视，引起了也形成了一些科学的认识和看法，有些地方对解决此类问题的做法也有值得借鉴的正面意义。将有第二步筛选后剩余的数据中，热度较高的问题，从中找到共性并将问题进行分类汇总，找到排名前五的问题。

4、根据题目要求做出表格（文件）。

5.2.3 数据处理过程及结果

1、分类指标

将筛选后的 33 个数据按地区/人群分类，得到 5 个主要分类指标：A 市 58 车贷案、A 市 A5 区汇金路、A 市房屋购买用户、A 市高铁附近、A 市各小区住户。

2、热度指数

将各个问题的总关注数求和得到热点问题关注总数为 13022，再将 5 个指标下的各个问题的关注总数分别求和，分别为 2386、2097、1829、767、319。热度指数公式为：

$$\text{热度指数}=\frac{\text{各指标指标下关注数}}{\text{热点问题关注总数}}\times 100\%$$

得到排名前五的热度指数分别为 18%、16%、14%、62%。

3、得到热点问题表

表 1 热度问题表

热度排名	问题 ID	热度指数	时间范围	地点/人群	问题描述
1	1	18%	2019/1/4 至 2019/5/28	A 市 58 车贷案	案情通报完成进度情况
2	2	16%	2019/8/19	A 市 A5 区汇金路	五矿万境 K9 县存在一系列问题
3	3	14%	2019/2/11 至 2019/6/12	A 市房屋购买用户	学区房产生的一系列问题
4	4	6%	2019/2/15 至 2019/9/6	A 市高铁附近	由高铁引起的经济及环境等问题
5	5	2%	2019/1/7 至 2019/12/25	A 市各小区住户	小区住户反映物业服务及管理差问题

4、按要求得到热点问题留言明细，见附录。

5.2.3 热点问题总结

因为社会热点应该是“政府问题”。即某一方面存在的问题，必须是政府所要着力解决的，而且也必须与政府的各个职能部门紧密相关。所以，由表 1 可知，

热点问题都是民生问题，主要集中在了社会中一直都存在的几类问题，如：在我们身边息息相关的小区物业服务及管理问题、为了小孩子上学引起的购买学区房问题、购买车辆的贷款问题、高铁站附件引起的居民用户坏境和经济问题等。这些问题年复一年、“生生不息”的在我们身边没有彻底的解决，但是希望政府可以每年做到改善。

5.3 答复意见的评价

评价方案将根据以下几个维度制定：

1. 相关性：根据回答内容与问题的相关性进行判断。
2. 完整性：根据回答内容的完整度进行判断。
3. 可解释性：根据回答内容的可理解度进行判断。
4. 时效性：根据回答问题的间隔时间进行判断。
5. 处理结果：根据问题的解决结果进行判断。

5.3.1 处理时效计算

```
import datetime

oldtime=datetime.datetime.now()

print oldtime;

x=1

while x<10000000:

x=x+1

newtime=datetime.datetime.now()

print newtime;

print u'相差：%s'%(newtime-oldtime)

print u'相差：%s 微秒'%(newtime-oldtime).microseconds

print u'相差：%s 秒'%(newtime-oldtime).seconds
```

得出答复时间与留言时间间隔，间隔较短较好，得到最优解。

5.3.2 TOPSIS 综合评价

本方法根据有限个评价对象与理想化目标的接近程度进行排序，适用于多项目标、对多个方案进行比较选择的分析方法。

1、构造初始矩阵 A（n 个评价指标，m 个目标）

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

式中，aij 表示第 i 个目标的第 j 项指标值(1≤i≤m, 1≤j≤n)

```
#获取数据

data=pd.read_excel('附件 4.xlsx').values

A=data[:,1:]  #获取初始矩阵

A=np.array(A)  #转为数组

m,n=A.shape[0],A.shape[1] # m,n 为行,列数
```

2、标准化处理

$$A'=\begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ a'_{21} & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{m1} & a'_{m2} & \cdots & a'_{mn} \end{bmatrix}$$

```
#标准化处理

A1=np.ones([m,n],float)

for i in range(n):

 mu=np.power(np.sum(np.power(A[i],2)),0.5)

 A1[i]=A[i]/mu
```

3、构造加权标准化矩阵 Z

$$Z=A'W=\begin{bmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ a'_{21} & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{m1} & a'_{m2} & \cdots & a'_{mn} \end{bmatrix}\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}=\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \cdots & z_{mn} \end{bmatrix}$$

式中，wj 是第 j 个指标的权重。权重确定方法：Delphi 法，对数最小二乘法，层次分析法等

```
W=[w1,w2,w3,w4,w5]

W=np.array(W)

Z=np.ones([m,n],float)

for i in range(len(W)):

        for j in range(len(W)):

            if i==j:

                W[i,j]=WO[j]

            else:

                W[i,j]=0
```

4、根据加权矩阵判断正负理想解 Z+ ,Z-

$$z_j^+ = \begin{matrix} \max\limits_i(z_{ij}),\ j\in J^* \\ \min\limits_i(z_{ij}),\ j\in J' \end{matrix},\ z_j^- = \begin{matrix} \min\limits_i(z_{ij}),\ j\in J^* \\ \max\limits_i(z_{ij}),\ j\in J' \end{matrix}$$

式中，J*是效益性指标集（指标值越大越好）；J’是成本型指标集（指标值越小越好）。

```
Zmax=np.ones([1,n],float)

Zmin=np.ones([1,n],float)

for j in range(n):

Zmax[0,j]=max(Z[:,j])

Zmin[0,j]=min(Z[:,j])
```

5、计算各个方案的到正理想点的距离 Si+和负理想点的距离 Si-

$$S_i^+ = \sqrt{\sum_{j=1}^{n}\left(z_{ij}-z_j^+\right)^2},\ i=1,2,\cdots,m$$

$$S_i^- = \sqrt{\sum_{j=1}^{n}\left(z_{ij}-z_j^-\right)^2},\ i=1,2,\cdots,m$$

```
for i in range(m):

 Smax=np.sqrt(np.sum(np.square(Z[i,:]-Zmax[0,:])))

 Smin=np.sqrt(np.sum(np.square(Z[i,:]-Zmin[0,:])))v
```

6、计算各个方案的相对贴近度 Ci

$$C_i = S_i^- \Big/ \left(S_i^+ + S_i^-\right) \ , \ i = 1,2,\cdots,m$$

```
C=Smin/(Smin+Smax)

C=pd.DataFrame(C,index=['院校' + i for i in list('12345')])
```

按每个方案的相对贴近度的大小进行排序，值越大越好，找出最优解。

5.3.3 数据标准化处理

1、效益型指标处理：指标值越大越好

$$a'_{ij} = \begin{cases} \left(a_{ij} - a_{j\min}\right) \Big/ \left(a_{j\max} - a_{j\min}\right), & a_{j\max} \neq a_{j\min} \\ 1, & a_{j\max} = a_{j\min} \end{cases}$$

```
#归一化处理

A1=np.ones([A.shape[0],A.shape[1]],float)

for i in n:

    if max(A[:,i])==min(A[:,i]):

        A[:,i]=1

    else:

        for j in m:

            A1[j,i]=(A[j,i]-min(A[:,i]))/(max(A[:,i])-min(A[:,i]))
```

2、成本型指标：指标值最小最好

$$a'_{ij} = \begin{cases} \left(a_{j\max} - a_{ij}\right) \Big/ \left(a_{j\max} - a_{j\min}\right), & a_{j\max} \neq a_{j\min} \\ 1, & a_{j\max} = a_{j\min} \end{cases}$$

```
#归一化处理

A1=np.ones([A.shape[0],A.shape[1]],float)

for i in n:

    if max(A[:,i])==min(A[:,i]):

        A[:,i]=1

    else:

        for j in m:
```

```
A1[j,i]=(max(A[:,i])-A[i,j])/(max(A[:,i])-min(A[:,i]))
```

3、中间型指标：指标最优值在某一点取得，值越靠近该点越好

$$a'_{ij}=M\Big/\left(M+\left|a_{ij}-M\right|\right)$$

式中，M 为取到最优值的点。

```
#归一化处理

A1=np.ones([A.shape[0],A.shape[1]],float)

for i in n:

    for j in m:

        A1[j,i]=M/(M+abs(A[j,i])-M))
```

```
#归一化处理

A1=np.ones([A.shape[0],A.shape[1]],float)

for i in n:

    for j in m:

        A1[j,i]=M/(M+abs(A[j,i])-M))
```

4、区间型指标：指标最优值落在一个区间范围呢。

设指标取值在区间[a,b]是最优的，最差下限为 lb,最差上限为 ub.

$$a'_{ij}=\begin{cases}\left(a_{ij}-lb\right)\Big/\left(a-lb\right), & lb\le a_{ij}<a\\ 1, & a\le a_{ij}<b\\ \left(ub-a_{ij}\right)\Big/\left(ub-b\right), & b\le a_{ij}\le ub\\ 0, & a_{ij}<lb\ or\ a_{ij}>ub\end{cases}$$

```
#归一化处理

A1=np.ones([A.shape[0],A.shape[1]],float)

for i in n:

    for j in m:

        if lb <= A[j,i]<= a:

            A1[j,i]=(A[j,i]-lb)/(a-lb)

        elif a <= A[j,i]<= b:

            A1[j,i]=1

        elif b <= A[j,i]<= ub:
```

```
        A1[j,i]=(ub-A[j,i])/(ub-b)
    else :  #A[j,i]< lb or A[j,i]>ub
        A1[j,i]=0
```

5.3.4 留言答复处理等级分类

根据评价留言答复处理体系，得到一系列数值，根据从大到小分为三个等级，分别为：优秀，良好，一般。分类详细见附录。

6. 总结

综上述研究内容和结果，我们对留言内容建立分类模型，方便留言内容分类，减少了工作量，节省了人力物力，并且对分类模型进行评价，效果良好；对于人们关心的话题，找到合理的热度评价指标，对热点问题进行评价，发现人们民生问题有着较高的关注度；针对留言回复内容，我们从相关性、完整性、可解释性、时效性和处理结果五个方面来评价，答复意见的质量根据从大到小分为三个等级，分别为：优秀，良好，一般。对于整个问题来说它是一个文本挖掘以及对于信息的评价问题，在本次实验中，给出了较为理想的评价模型，对于整体数据的挖掘十分有效，本次挖掘结果理想。

7. 参考文献

[1]EchoCaiCai.R 语言做文本挖掘 part1 安装依赖包 https://blog.csdn.net/cl1143015961/article/details/44082731?utm_source=app.2015 年 3 月 5 日发布。

[2]EchoCaiCai.R 语言做文本挖掘 part2 分词处理.http://blog.csdn.net/cl1143015961/article/details/44108143.2015 年 3 月 6 日发布。

[3]EchoCaiCai.R 语言做文本挖掘 part4 文本分类.https://blog.csdn.net/cl1143015961/article/details/44413631?utm_source=app.2015 年 3 月 18 日发布。

[4]宗成庆，夏睿，张家俊.文本数据挖掘[M].北京：清华大学出版社，2019.

[5]Julia Silge，David Robinson.文本挖掘——基于 R 语言的整洁工具[M].北京：机械工业出版社，2018。

8. 附录

8.1 问题一程序

```
install.packages("corpus")
install.packages("Rwordseg")
install.packages("tmcn")
install.packages("tm")
install.packages("HMM")
library(HMM)
library(Rwordseg)
library(tmcn)
library(tm)
library(NLP)
library(corpus)
library(class)
library(sampling)
d<-read.csv("D:/cd/rt.csv")
d$留言主题<-as.character(d$留言主题)
d$一级标签<-as.character(d$一级标签)
length(d)
table(d$一级标签)
sub=strata(d,stratanames="一级标签
",size=c(300,300,300,300,300,300,300),method="srswor")
head(sub)
temp<-gsub('[0-9]','',d$留言主题)
segwords1<-segmentCN(temp)
stopwords<-stopwordsCN()
segwords2<-lapply(segwords1,removeWords,stopwords)
corpus_d<-Corpus(VectorSource(segwords2))
dtm<-DocumentTermMatrix(corpus_d,control=list(wordLengths=c(2,Inf)))
```

```
dtm
dtm_matrix<-as.matrix(dtm)
dim(dtm_matrix)
data.train<-dtm_matrix[c(1:200,301:500,601:800,901:1100,1201:1400,150
1:1700,1801:2100),]
data.test<-dtm_matrix[c(201:300,501:600,801:900,1101:1200,1401:1500,1
701:1800,2001:2100),]
row.names(data.train)<-d$一级标签
[c(1:200,301:500,601:800,901:1100,1201:1400,1501:1700,1801:2100)]
row.names(data.test)<-NULL
d_class<-as.factor(row.names(data.train))
predict<-knn(train=data.train,test=data.test,cl=d_class)
shiji<-d$一级标签
[c(201:300,501:600,801:900,1101:1200,1401:1500,1701:1800,2001:2100)]
table(predict,shiji)
sum(diag(table(predict,shiji)))/sum(table(predict,shiji))
```

8.2 问题三程序

程序一：

```
import numpy as np
import pandas as pd

#TOPSIS 方法函数
def Topsis(A1):
    W0=[0.2,0.3,0.4,0.1] #权重矩阵
    W=np.ones([A1.shape[1],A1.shape[1]],float)
    for i in range(len(W)):
        for j in range(len(W)):
```

```
            if i==j:
                W[i,j]=WO[j]
            else:
                W[i,j]=0
    Z=np.ones([A1.shape[0],A1.shape[1]],float)
    Z=np.dot(A1,W) #加权矩阵

    #计算正、负理想解
    Zmax=np.ones([1,A1.shape[1]],float)
    Zmin=np.ones([1,A1.shape[1]],float)
    for j in range(A1.shape[1]):
        if j==3:
            Zmax[0,j]=min(Z[:,j])
            Zmin[0,j]=max(Z[:,j])
        else:
            Zmax[0,j]=max(Z[:,j])
            Zmin[0,j]=min(Z[:,j])

    #计算各个方案的相对贴近度 C
    C=[]
    for i in range(A1.shape[0]):
            Smax=np.sqrt(np.sum(np.square(Z[i,:]-Zmax[0,:])))
            Smin=np.sqrt(np.sum(np.square(Z[i,:]-Zmin[0,:])))
            C.append(Smin/(Smax+Smin))
    C=pd.DataFrame(C,index=['院校' + i for i in list('12345')])
    return C

#标准化处理
def standard(A):
```

```
    #效益型指标
    A1=np.ones([A.shape[0],A.shape[1]],float)
    for i in range(A.shape[1]):
        if i==0 or i==2:
            if max(A[:,i])==min(A[:,i]):
                A1[:,i]=1
            else:
                for j in range(A.shape[0]):

A1[j,i]=(A[j,i]-min(A[:,i]))/(max(A[:,i])-min(A[:,i]))

    #成本型指标
        elif i==3:
            if max(A[:,i])==min(A[:,i]):
                A1[:,i]=1
            else:
                for j in range(A.shape[0]):

A1[j,i]=(max(A[:,i])-A[j,i])/(max(A[:,i])-min(A[:,i]))

    #区间型指标
        else:
            a,b,lb,ub=5,6,2,12
            for j in range(A.shape[0]):
                if lb <= A[j,i] < a:
                    A1[j,i]=(A[j,i]-lb)/(a-lb)
                elif a <= A[j,i] < b:
                    A1[j,i]=1
                elif b <= A[j,i] <= ub:
```

```
                    A1[j,i]=(ub-A[j,i])/(ub-b)
                else:  #A[i,:]< lb or A[i,:]>ub
                    A1[j,i]=0
    return A1

#读取初始矩阵并计算
def data(file_path):
    data=pd.read_excel(file_path).values
    A=data[:,1:]
    A=np.array(A)
    #m,n=A.shape[0],A.shape[1] #m 表示行数,n 表示列数
    return A

#权重
A=data('附件 4.xlsx')
A1=standard(A)
C=Topsis(A1)
print(C)
```

程序二：

```
import datetime
oldtime=datetime.datetime.now()
print oldtime;
x=1
while x<10000000:
  x=x+1
newtime=datetime.datetime.now()
print newtime;
```

```
print u'相差：%s'%(newtime-oldtime)
print u'相差：%s 微秒'%(newtime-oldtime).microseconds
print u'相差：%s 秒'%(newtime-oldtime).seconds
```