“智慧政务”中的文本挖掘应用

摘要

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。

在本次数据挖掘过程中，对于问题1，对留言数据进行预处理、文本分词、停用词过滤、文本向量化表达后，通过朴素贝叶斯模型来建立关于群众留言内容的一级标签分类模型，再用F-Score进行模型评价，以便后续将群众留言分派至相应的职能部门处理。

对于问题2，通过对群众留言进行基本操作后，再进行词性标注、文本向量化表达，实现对留言数据的优化，提升其可建模度。紧接用K-means聚类构建数据挖掘模型，对群众留言反映的特定地点或特定人群问题进行归类，定义合理的热度评价指标，并给出评价结果，制作出热点问题表和热点问题留言明细表，以便及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。

对于问题3，通过对群众留言原数据进行预处理、文本分词、停用词过滤和文本向量化，再基于numpy.Corrcoef实现相关性分析来构造模型。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，从而提升政府的管理水平和施政效率。

关键词：智慧政务　文本分析　中文分词　TF-IDF 算法　K-means 聚类

Application of text mining in "intelligent government affairs"

Abstract

In recent years, with WeChat, such as weibo, mayor mailbox, sun hotline network asks the political platform gradually become the government understand the importance of public opinion, gathering intelligence, condensed bull channels, all kinds of public opinion related text data quantity rising, leave a message to past mainly rely on artificial to divide and hot spots of relevant departments work has brought great challenge. At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of intelligent government affairs system based on natural language processing technology has become a new trend of social governance innovation and development, which plays a great role in promoting the management level and efficiency of the government.

In the process of the data mining, for question 1, the message data preprocessing text participle stop words filtering after vectorization expression, through the simple bayesian model to establish the level about the message content label classification model, then use the F - Score model evaluation, so that subsequent message assigned to the corresponding functional departments to deal with the masses.

For question 2, through the study of the basic operation of the message, and then to part-of-speech tagging text to the quantitative expression, realize the optimization of data message, promotes its modeling followed with K means clustering data mining model was constructed, the mass comments reflect a specific location or specific problems are classified, define the reasonable heat evaluation indexes, and gives the evaluation results, make a hot issue list and hot issues the message list, so as to find hot issues in a timely manner, targeted help relevant departments, and improve service efficiency.

For question 3, the model is constructed by preprocessing, text segmentation, stop word filtering and text vectorization of the original data of public comments, and then correlation analysis based on numpy.corrcoef. From the perspective of relevance, completeness and interpretability of replies, this paper gives a set of evaluation scheme for the quality of replies, so as to improve the management level and efficiency of government.

Keywords: Wisdom government affairs The text analysis Chinese word-segmentation TF - IDF algorithm K - means clustering

目录

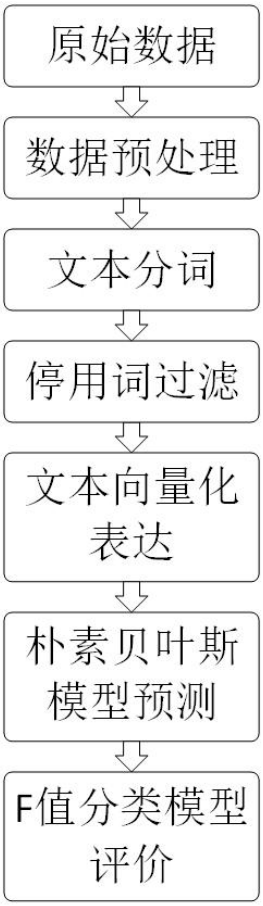
1. 挖掘目标

本次建模针对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的文本数据，对留言数据进行预处理、文本分词、停用词过滤、文本向量化表达后，通过朴素贝叶斯模型来建立关于群众留言内容的一级标签分类模型，再用F-Score进行模型评价解决第一问。通过对群众留言进行基本操作后，再进行词性标注、文本向量化表达，用K-means聚类构建数据挖掘模型，实现对群众留内容反映的热点问题进行归类，最后通过文本向量化，再基于numpy.Corrcoef实现相关性分析来构造模型对留言内容进行相对应的答复意见，从而提升政府的管理水平和施政效率，使人民生活更加便利。

2. 分析方法与过程

2.1. 问题 1 分析方法与过程

2.1.1. 流程图



2.1.2. 数据预处理

得到文本数据后，首先需要对文本留言进行预处理。文本留言数据里面含有大量价值不高的条目，倘若将这些条目进行接下来的操作流程，必将会对分析过程乃至最终的结果产生不可估量的影响，最终可能导致结果不够精确。因此为了使接下来的分析更加有效和流畅，我们先将这些价值含量不高的留言去掉。

对这些文本留言数据进行数据预处理主要有两个部分：数据筛选、文本去重。按照各个部分的特性，我们就按这个顺序进行文本留言数据的预处理。

2.1.2.1. 数据筛选

数据中包含一些价值不大的信息，例如留言编号，留言用户等，这些信息在后续的文本分析中作用不大，根据题目需求，只需将留言主题，留言详情等信息筛选出来即可，此外，后续还需进行分词，机器学习等，因此我们将这些有意义的信息筛选出来，简化处理操作。

2.1.2.2. 文本去重[1]

文本去重，顾名思义就是对文本中重复的数据删除掉，无论获取到什么数据，一般都会包含一些重复的数据，例如在本题中，重复的原因可能有：

① 用户不熟悉智慧政务平台的操作，留言多次提交，造成重复。

② 留言未及时处理导致用户多次发表留言。

③ 文本数据中包含重复且无意义的词，比如：哈哈哈哈，好好好好。

2.1.3. 文本分词

在中文中，只有字、句和段落能够通过明显的分隔符进行简单的划界，而对于词或词组来说，它们的边界模糊，没有一个形式上的分隔符，不同于英文中每个单词之前都会有空格相连，所以在处理中文文本挖掘时首先要进行分词，即连续的字符串按照一定的规范重新组合成词序列的过程[2]。

分词的结果的准确性也会对后续对文本做数据分析也有着一些不可忽视的影响，例如词的特征选择，因此，在进行中文文本分词时，应当选用合适的分词模型。

对这些文本留言数据进行中文文本分词主要有两个步骤：加载新词、jieba 分词。

2.1.3.1. 加载新词

由于数据中存在着数字连着中文的地名，例如 A 市，A1 区等，因此分词常常将这些词分开来，如果一旦分开，两个词的意义会完全不同，所以我们将所有地名筛选出来，并保存在 TXT 中，通过 jieba 库的 load_userdict 导入包含所有地名的 TXT 文档。

2.1.3.2. jieba 分词

中文分词是中文文本处理的一个基础性工作，结巴分词利用进行中文分词。其基本实现原理有三点：

① 基于 Trie 树结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG）

② 采用了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

③ 对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法

首先我们导入 jieba 库，使用 apply 与 lambda 函数去重后的数据进行 jieba 分词。

2.1.4. 去除停用词

经过分词操作后，我们得到了结果为一个词的集合，即 d=（a1,a2,a3…），其中包括换行符，制表符在内的多个标点符号以及一些停用词，这些词在文本数据表达中是毫无意义的，因此在这步操作中我们将一些停用词（包括一些标点符号、换行符以及空格等）筛选并删除。这个过程被称为去除停用词，停用词有两个特征：一是比较普遍，在文本中出现的次数较高，例如文本中的：的，了，啊，嗯等。在后续特征选取的操作过程中，这些停用词的介入可能会降低特征的准确度，从而使分析出来的结果不够精准。

本题中采用基于停用词表的停用词过滤方式，将分词的结果与停用词表中的词进行匹配，若匹配成功，则去除。

2.1.4.1. 划分数据和标签

对分词后的数据划分出留言详情的分词数据并用空格将各个分好的词连接起来，同时也划分出每条数据的对应的标签，分别保存在两个变量中，以便后续模型的建立。

2.1.5. 模型的建立

2.1.5.1. 数据库的欠抽样

为了平衡数据中一级标签的分配比例，以便对数据进行模型预测，提高有监督学习的训练准确度，所以就对每种一级标签的数据进行抽样，在本题中，我们采取欠抽样的方式，对每一条一级标签的数各取 600 条，并使用 concat 将抽样后的数据进行拼接，形成一个每种一级标签比例相同的数据。

2.1.5.2. 划分训练集和测试集

通过有监督学习的方式，对数据进行模型预测，所以我们首先划分出数据以及标签的训练集作为学习的依据，然后划分出测试集作为预测的依据。

2.1.5.3. 词向量转换（TF-IDF）

TF-IDF（term frequency–inverse document frequency）是一种用于资讯检索与资讯探勘的常用加权技术。TF-IDF 是一种统计方法，用以评估一字词对于一个文件集或一份文件对于所在的一个语料库中的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。TF 意思是词频(Term Frequency)，IDF 意思是逆向文件频率(Inverse Document Frequency)。FIDF 的主要思想是：如果某个词或短语在一篇文章中出现的频率 TF 高，并且在其他文章中很少出现，即 IDF 低，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

可以将一个词对的 TF-IDF 值表示为公式-1，该值明确定义了对于分类的重要性，值越大，说明越有益于分类；值越小，说明越无益于分类。

分别实例化一个 CountVectorizer() 和 TfidfTransformer() 对象，调用 CountVectorizer()对象的 fit_transform()方法对留言分词数据的训练集转换为词频向量，并转化为 array 数组的形式，然再调用 TfidfTransformer()对象的 fit_transform()方法对词频向量的数据进行 tf-idf 权值向量的转化。需要注意的是，留言数据中的测试与训练集词语长度不匹配，导致预测的结果不准确。因此我们对测试集数据的 CountVectorizer()对象传入 vocabulary= vectorizer.vocabulary_参数，使得训练集与测试局的词语长度一致。

2.1.5.4. 建立模型

朴素贝叶斯模型（Naive Bayes Model），是一种基于贝叶斯定理与特征条件

独立假设的分类方法，与决策树模型（Decision Tree Model）同为目前使用最广泛的分类模型之一，在各个领域都有广泛的应用，例如我们经常会用到的垃圾邮件的分类功能。

使用朴素贝叶斯模型[3]，传入 tf-idf 转化后的留言数据的训练集以及对应的标签，最后将 tf-idf 转化后的测试集传入 predict 函数中预测出留言数据测试集所对应的标签。

2.1.6. 模型的评价

通常使用 F-Score 对分类方法进行评价：

$$F_1 = \frac{1}{\mathrm{n}} \sum_{i=1}^{n} \frac{2P_i R_i}{P_i + R_i}, \qquad (1)$$

其中 Pi 为第 i 类的查准率， Ri 为第 i 类的查全率。

真实情况	预测情况	
	正例	反例
正例	TP	FN
反例	FP	TN

查准率：TP/TP+FP

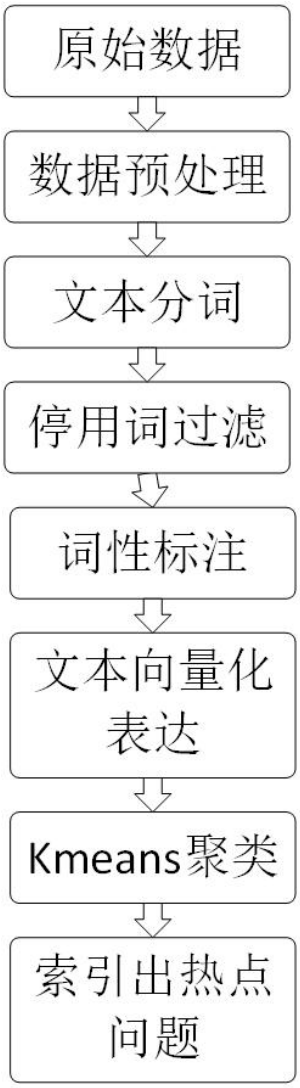
查全率：TP/TP+FN

由于预测出的数据是一个多分类模型，所以通过将多分类转化为多个二分类问题分别计算。

分别计算每个标签真实情况与预测情况的的查准率与查全率，最后通过公式（1）对每个查准率和查全率进行计算得出预测模型的 F-score 大约为 0.70。

2.2. 问题 2 分析方法与过程

2.2.1. 流程图



2.2.2. 数据预处理

2.2.2.1. 数据筛选

观察问题以及所需要得到的结果，热点问题挖掘的对象是留言主题，我们首先对留言主题进行提取，筛选出留言主题的文本，其次筛选出留言详情，时间以及点赞数，反对数，通过 pandas 的 loc 函数去筛选，以便后续对文本的操作以及向量化。

2.2.2.2. 去除无意义词语

由于文本是留言主题，其中存在一些无意义的词语，例如）等其他标点符号，我们使用正则表达式 re.sub 方法将无意义的词语替换成空字符串。

2.2.2.3. 数据去重，去空[1]

考虑到所给数据可能会有空文本，以及存在一些完全相同的文本，我们使用 drop_duplicates()dropna()去除重复和空的文本行。

2.2.3. 文本分词[2]

2.2.3.1. 新词载入 jieba 词库

由于 jieba 分词不能识别出文本数据中的地名，例如 A 市，A1 区等，分词往往将它们拆分开来，拆分后形成的词语就能构成地名，对特征词的提取以及词性标注会产生影响，所以我们事先提取出所有地名的词语，保存在 TXT 文档中，并通过 load_userdict() 的方法载入到 jieba 词库。

2.2.3.2. Jieba 分词

普通的分词方法是遍历所有行，并对每行进行分词。这种做法需要使用到 for 循环，增大计算机计算负担，所以我们改进此方法，通过 dataframe 内置的 apply 函数以及 lambda 函数对文本数据进行操作，这样可以省去遍历的步骤，减少计算机的计算量。

2.2.3.3. 去除停用词

分词后每行都得到了由词语构成的列表，观察这些词语，发现存在\n, \t 等符号，除此之外，一些在文本中无意义的词，例如语气助词等我们需要将他们去除，一是为了缩短分词的长度，二十能对后续的预测提高精确度。

在网上查找一些常用的停用词表 stopwords，对整个数据使用 apply 函数以及 lambda 函数从而对美整个数据的每行数据进行操作，与停用词表中的词进行匹配，提取出不在停用词的分词结果。apply(lambda x: [i for i in x if i not in stopWords]).

2.2.4. 文本向量化表达

首先实例化两个对象 vectorizer=CountVectorizer()，transformer= TfidfTransformer().

Vertorizer 用于将文本转换为词频向量，transformer 用于将文本转换成 TF-IDF 权值向量，最后通过 toarray 的方式转换为矩阵 array([0,0,0…,0,0,0])的形式。

2.2.5. Kmeans 聚类

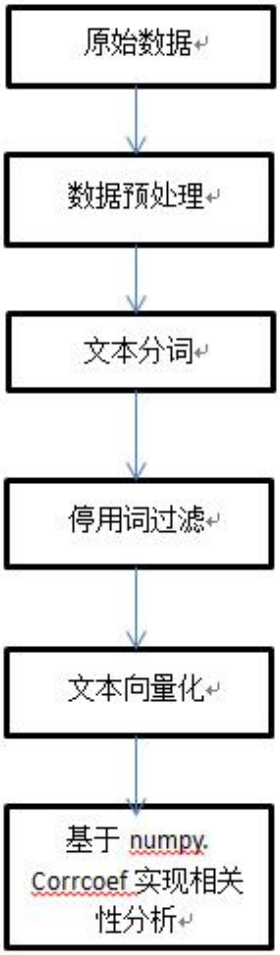
由于具有出色的速度和良好的可扩展性，Kmeans 聚类算法[3]算得上是最著名的聚类方法。Kmeans 算法是一个重复移动类中心点的过程，把类的中心点，也称重心(centroids)，移动到其包含成员的平均位置，然后重新划分其内部成员。k 是算法计算出的超参数，表示类的数量；Kmeans 可以自动分配样本到不同的类。

在本题中，我们将附件 3 数据的留言主题、留言详情、留言时间、点赞数以及反对数筛选出来，将 TF-IDF 权值转化过的留言主题向量进行数组化 array([0,0,0,…0,0,0]) 输入，进行 K 值为 1000 的聚类，得到一个长度为数据集长度的一维数组 array([x1,x2,x3,x4…])，一维数组中数值相同的类别为一类，一共有 1000 类。

通过字典统计词频的方式，使用 sorted 函数统计出现次数最多的 5 个类别，这些类别分别对应一系列热点问题，热度指数即为出现的次数，通过这些类别在一维数组中的索引作为文本数据的索引索引出相对应的热点问题，在通过 jieba 分词中标注的词性索引出热点问题的地名人名。

2.3. 问题 3 分析方法与过程

2.3.1. 流程图



2.3.2. 数据预处理

2.3.2.1. 数据筛选

根据问题以及所要得到的结果，从相关性角度考虑，问题 3 需要的是留言主题和答复意见两者之间的相关系数，我们首先对留言主题进行提取，筛选出留言主题的文本，其次对答复意见的提取，筛选出答复意见的文本，通过 pandas 的 loc 函数去筛选，以便后续对文本的操作以及向量化。

2.3.2.2. 去除无意义词语

无意义的词语会使操作更加复杂，花很多时间做无用功。由于文本数据是留言主题，其中存在一些无意义的词语，例如）等其他标点符号，我们使用正则表达式 re.sub 方法

将无意义的词语替换成空字符串。

2.3.3. 文本分词[2]

2.3.3.1. 新词载入 jieba 词库

由于 jieba 分词不能识别出文本数据中的地名，分词能将它们拆分开，拆分后形成的词语就能构成地名，对特征词的提取以及词性标注会产生影响，所以我们事先提取出所有地名的词语，保存在 TXT 文档中，并通过 load_userdict()的方法载入到 jieba 词库。

2.3.3.2. jieba 分词

普通的分词方法是遍历所有行，并对每行进行分词。这种做法需要使用到 for 循环，增大计算机计算负担，所以我们改进此方法，通过 dataframe 内置的 apply 函数以及 lambda 函数对文本数据进行操作，这样可以省去一些步骤，减少计算机的计算量。

2.3.3.3. 去除停用词

分词后每行都得到了由词语构成的列表，观察这些词语，发现存在\n, \t 等符号，除此之外，还有一些在文本中无意义的词，例如语气助词等。我们需要将他们去除，一是为了缩短分词的长度，二十能对后续的预测提高精确度。

在网上查找一些常用的停用词表 stopwords，对整个数据使用 apply 函数以及 lambda 函数从而对美整个数据的每行数据进行操作，与停用词表中的词进行匹配，提取出不在停用词的分词结果。

2.3.4. 文本向量化表达

首先实例化两个对象 vectorizer = CountVectorizer(),transformer = TfidfTransformer().

Vertorizer 用于将文本转换为词频向量，transformer 用于将文本转换成 TF-IDF 权值向量，最后通过 toarray 的方式转换为矩阵的形式。

2.3.5. 相关性分析

导入 numpy 库，调用 numpy.corrcoef()函数[5]，然后通过 for 循环语句对附件 4 中每一行的“留言主题”和“答复意见”进行遍历，最后得到结果 array［］矩

阵，矩阵中的数字即为两者间的相关系数。

3. 结果分析

3.1. 问题 1 结果分析

通过对留言数据以及标签划分的训练集进行朴素贝叶斯模型训练，放入留言数据的测试集得到标签列表，该标签列表为留言数据测试集预测的标签，我们通过对留言数据测试集的真实标签与预测出来的标签进行评价：

将留言数据的测试集重新索引后取出得到真实的标签数据，然后分别取出 7 个标签对应的位置，与预测出来的标签数据分别计算每个标签的 TP,FP,TN,FN 的值，在通过：

P=TP/(TP+FP)

R=TP/(TP+FN)

计算每个标签对应的查准率与查全率，得到结果如下：

标签为交通运输的查准率 P=0.75，查全率 R=0.53

标签为商贸旅游的查准率 P=0.53，查全率 R=0.66

标签为城乡建设的查准率 P=0.54，查全率 R=0.53

标签为教育文体的查准率 P=0.72，查全率 R=0.79

标签为环境保护的查准率 P=0.75，查全率 R=0.84

标签为劳动社会保障的查准率 P=0.76，查全率 R=0.62

标签为卫生计生的查准率 P=0.76，查全率 R=0.81

最后通过

F-Score：得出该分类模型的 F-Score 值为 0.68

3.2. 问题 2 结果分析

留言主题的 TF-IDF 权值向量经过 KMeans 聚类后，得到一个包含 4209 个数据的一维数组：

array([976, 99, 818, ..., 651, 651, 651])，

观察可以发现其中一些数值相等，用字典统计词频的方式得到所有标签的出现次数，经过 sorted 函数排序后并提取前 5 个标签以及对应出现的次数：

[(25, 31), (563, 23), (84, 22), (22, 22), (177, 17)]

索引出每个标签在矩阵中对应的位置：其中：

25：[[33, 68, 77, 98, 105, 205, 631, 847, 1047, 1069, 1086,

1162, 1203, 1233, 1312, 1917, 1964, 2072, 2114, 2132, 2291, 2325,

2898, 3210, 3317, 3491, 3840, 3863, 3933, 3995, 4097]]，31

563：[[318, 701, 729, 785, 1128, 1291, 1477, 1538, 1933, 1993, 2100,
2316, 2620, 2781, 2887, 2998, 3327, 3658, 3954, 4049, 4058, 4180,
4202]]，23

84：[[121, 522, 533, 690, 1166, 1283, 1398, 1549, 1970, 2098, 2101,
2149, 2161, 2380, 2455, 2499, 2803, 2865, 3018, 3360, 4022, 4140]]，
22

22：[[173, 278, 302, 397, 619, 632, 797, 953, 1131, 1220, 1299,
1494, 2099, 2115, 2370, 2942, 2947, 3065, 3336, 3623, 3716, 4141]]，
22

177：[[61, 792, 1070, 1117, 1201, 1431, 1647, 1885, 1937, 2102, 2182,
2586, 2818, 3021, 3109, 3243, 3349]]，17

通过这些索引：使用 loc 的方法在处理后的留言数据中索引出特定列，每一类标签代表了一个簇，数量最大的前 5 个簇即为所要求得热点问题。对应的热度指数即为每一个标签所对应在矩阵出现的次数。

3.3. 问题 3 结果分析

留言主题和答复意见的词向量矩阵经过 numpy.corrcoef() 函数后，得到每一行数据的相关系数，通过索引取出前 20 列的相关系数矩阵为：

[array([[1. , 0.22776429],
[0.22776429, 1.]]),
array([[1. , 0.33490663],
[0.33490663, 1.]]),
array([[1. , 0.39632404],
[0.39632404, 1.]]),
array([[1. , 0.40456729],
[0.40456729, 1.]]),
array([[1. , 0.18028606],
[0.18028606, 1.]]),
array([[1. , 0.43539545],
[0.43539545, 1.]]),
array([[1. , 0.50444298],
[0.50444298, 1.]]),
array([[1. , 0.25534212],
[0.25534212, 1.]]),

```
array([[1.        , 0.2299883],
       [0.2299883, 1.       ]]),
array([[1.        , 0.27943995],
       [0.27943995, 1.       ]]),
array([[1.        , 0.0691635],
       [0.0691635, 1.       ]]),
array([[1.        , 0.61472068],
       [0.61472068, 1.       ]]),
array([[1.        , 0.32581195],
       [0.32581195, 1.       ]]),
array([[ 1.0000000e+00, -3.8502629e-04],
       [-3.8502629e-04,  1.0000000e+00]]),
array([[1.        , 0.15570247],
       [0.15570247, 1.       ]]),
array([[1.        , 0.27997626],
       [0.27997626, 1.       ]]),
array([[1.        , 0.06180513],
       [0.06180513, 1.       ]]),
array([[1.        , 0.37363232],
       [0.37363232, 1.       ]]),
array([[1.        , 0.18344499],
       [0.18344499, 1.       ]]),
array([[ 1.00000000e+00, -4.11232896e-04],
       [-4.11232896e-04,  1.00000000e+00]])]
```

通过观察结果可看出，相关部门的答复意见基本上能解决群众提出的问题。

4. 总结

智慧政务可以实现政务服务高效化，数据实时化，响应及时化的政务工具，可以简单、高效的解决百姓的各类问题，数据统计清晰明了化。其中智慧政务的主要作用有这四点：建立百多元化政务服务供给渠度道，创新国家治理方式；加快新一代信息技术的应用推广，建设智问慧政府答；法律保障体系，改善促进国家治理现代化的网络环境回；解决老百姓办事难的问题（信息答多跑路，群众少跑路）。如今随着大数据、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，相信这种趋势很快

就会成为一种大流行趋势，当然，智慧政务还需继续完善，从而更好地提升政府的管理水平和施政效率。

5. 参考文献

[1]杨虎.面向海量短文文本去重技术的研究与实现.国防科学技术大学.2007

[2]曹卫峰.中文分词关键技术研究[D].南京理工大学.硕士学位论文.2009

[3]史努.机器学习数学原理——朴素贝叶斯模型[Z].2018

[4]王千，王成，冯振元，叶金凤.K-means聚类算法研究综述.2012

[5]Python Numpy库 numpy.corrcoef()函数讲解[Z].https://blog.csdn.net/qq_39514033/article/details/88931639,2019-03-31