文本挖掘在智慧政务建设中的应用

摘要

智慧政务的建设是建设智慧城市的重要策略之一，也是提升政府的管理水平和施政效率的有效方法。随着智慧应用的日趋多元化、大数据等技术的逐渐发展，加强对于智慧政务系统建设情况的检测与理解、提供基于数据的精准化规划指导变得更加重要。在此背景下，本文结合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，基于自然语言处理和文本挖掘，通过讨论，归纳总结出建设政府政务工作的创新发展的最佳方案。

对于问题一，利用 Python 从附件 2 中原有的群众留言数据对其进行数据预处理之后，对于得到的处理过的数据，通过 TF-IDF 将其进行向量化后，构建高斯朴素贝叶斯模型的分类方法，根据 F-Score 对该分类方法进行评价，准确率在 69%到 71%之间。

对于问题二，利用 Python 对附件 3 中的留言主题进行数据预处理之后，通过 TF-IDF 将其进行向量化，再通过 K-均值聚类算法对留言向量进行聚类，最后找出最大的 5 类并筛选出距离 5 个聚类中心最近的留言向量即排名前 5 位的热点问题。

对于问题三，利用 Python 对附件 4 中的留言详情和答复进行数据预处理过滤掉无意义的词，取两类数据预处理后得到所有词的词合并成为关键词，通过 TF-IDF 将其进行向量化，找出留言相对应的答复中两者词的相同性和相似性，再通过两个向量的余弦相似度分析，得到相关部门对相应留言的答复的相关性约为 0.746。

关键词：Python TF-IDF 高斯朴素贝叶斯 K-均值聚类 余弦相似度

The application of text mining in the construction of smart government

Abstract

In this context, this paper combines the records of the public political message from the open sources of the Internet and the response opinions of relevant departments to some of the public message.Based on natural language processing and text mining, this paper sums up the best scheme for the innovation and development of government affairs through discussion.

For question 1, After Python preprocesses the original mass message data in Annex 2, the obtained processed data is vectorized by TF-IDF, and then a classification method of Gaussian naive Bayesian model is constructed. According to F-score, the classification method is evaluated, with an accuracy of 69% to 71%.

For question 2, After preprocessing the message subject in attachment 3 by Python, it is vectorized by TF-IDF, and then clustering the message vector by K-means clustering algorithm. Finally, the largest five categories are found and the message vector closest to the five clustering centers, i.e. the top five hot issues, is screened.

For question 3, Python is used to preprocess the message details and replies in Annex 4 to filter out meaningless words. After preprocessing two types of data, the words of all the words are combined into keywords. TF-IDF is used to vectorize them and find out the similarity and similarity of the two words in the corresponding replies of the message. Then through cosine similarity analysis of the two vectors, relevant departments can get the corresponding The relevance of the response to the message is about 0.746.

Key words:Python、TF-IDF、Gaussian Naive Bayes、K-Means、Cosine similarity

目录

一、挖掘目标

本次数据挖掘的目标是利用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 Python 等软件并使用了高斯朴素贝叶斯模型、K-均值聚类算法的数学分析方法得出当中隐含的信息，并通过这些得出的信息解决下列问题：

1. 群众留言分类。根据附件 1 提供的内容分类三级标签体系，分析其体系构成，并利用附件 2 所给出的数据，建立关于其留言内容的一级分类标签。

2. 热点问题挖掘。热点问题的及时发现，对于相关部门进行有针对性的处理有极大的促进作用，利用附件 3 的数据，定义合理的热度评价指标，并找出排名前 5 的热点问题以及相应热点问题对应的留言信息。

3. 答复意见的评价。根据附件 4 相关部门对留言的答复意见，试建立并尝试实现从答复相关性、完整性、可解释性角度评价答复意见质量的方案。

二、问题分析

2.1 问题一的分析

问题一属于文本分类类型的问题，对附件 2 所给出的数据进行文本分类，读取出有价值的留言数据，并将这些数据进行分类，并使用高斯贝叶斯模型，把训练数据和对应的标签放入模型中进行训练，并用 F-Score 对分类方法进行评价。

2.2 问题二的分析

问题二我们需要寻找出附件 3 中排名前五位的热点问题。首先我们需要把留言中不是反映问题的留言清理掉，再将留言提取出来，对其进行数据预处理操作，将预处理完之后的留言进行向量化，然后用 K-均值聚类算法对留言向量进行聚类，其中 K 值采用“手肘法”进行确定，最后从聚类结果中找出最大的那 5 个类，这 5 个类聚类中心点附近的向量对应的留言，即为排名前 5 位的热点问题。

2.3 问题三的分析

关于答复的相关性：通过对答复和问题进行量化，构建评价指标反应相关部门回复这个问题的质量如何。对附件 4 中的留言详情和答复进行数据预处理过滤掉无意义的词，取两类数据预处理后得到所有词的词合并成为关键词，将数据量化，找出留言相对应的答复中两者词的相同性和相似性，再通过两个向量的余弦相似度分析评价相关部门对相应留言的答复的质量如何。

对于答复的完整性：对于相关性我们要分析的是文本数据中问题的答复是否符合 mysql 完整性约束，mysql 完整性约束可用于保证数据的完整性和一致性。数据分析之前首先要保证文本数据集的质量，在 pandas 数据源中用 missingno 库提供了一个灵活易用的可视化工具来观察数据缺失情况。然后用 Matrix 函数将数据集矩阵显示，能更直观看到其完整性，利用标出完整性点的图计算出整个数据完整性的平均值。

三、模型假设

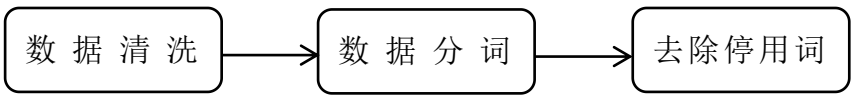
1. 假设题目所给的文件数据真实可靠
2. 假设停用词表包括了所有的停用词
3. 假设 80%的数据训练结果等于全部数据训练结果

四、定义与符号说明

符号	定义与说明
$n_{i,j}$	该词在文件中出现的次数
$\sum_k n_{k,j}$	文件 d_j 中所有词汇出现的次数总和
$\lvert D \rvert$	语料库中的文件总数
$\lvert \{ j : t_i \in d_j \} \rvert$	包含 t_i 词语的文件数目

x，y_i, y_j	样本
P_i	第 i 类的查准率
R_i	第 i 类的查全率
$P(y_i)$	先验概率
$P(x\|y_i)$	似然度
$P(y_i\|x)$	后验概率
$J(c_k)$	该类内个点到聚类中心距离的平方和
$J(c)$	各类总的距离的平方和
μ_k	聚类中心
C_i	第 i 个簇
P	C_i 中的样本点
m_i	C_i 的质心

五、数据预处理



5.数据预处理总体流程

5.1 数据清洗

对于附件 2、3、4 的数据，均需要利用 Python 进行对回车符、换行符及特殊符号的字符进行去除操作。首先要导入正则表达式 re 模块，用 re.sub 进行对特殊符号的替换操作，re.sub 是正则表达式的函数是对于输入的一个字符串，利用正则表达式，去实现相对复杂的字符串替换处理，然后返回被替换后的字符串，它能够实现比普通字符串的 replace 更加强大的替换功能。

5.2 数据分词

中文分词是文本挖掘后续处理的基础，是将文本数据中的句子分成一个个具有单独词性的词组或字，jieba 库的分词是 Python 中基于前缀词典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)，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，是 Python 非常强大的中文分词包。对于清洗过的数据我们用的是 jieba 库进行分词，导入 jieba 库用 jieba.lcut 对每条数据进行分词并返回 list 列表的形式。Jieba 库有通用词典但是在默认情况下，使用默认分词，是识别不出文本数据中的一些（“东二环”，“牛塘角”）等新兴词，所以需要自定义添加词和字典使使用用户字典提高分词准确性。

5.3 去除停用词

分词后的数据其实还有很多无用的词，这些词有可能会对后续分类而言反而是累赘，所以我们要进行停用词的去除。我们准备了通用的停用词表并将词表导入进行读取的操作（用 read_cvs 读取词表时，词表中的存在的元素不能作为分隔符号）。如还有其他无用元素未存在词表中，可以建立列表与转为列表形式的停用词表进行合并，并将这些停用词从文本数据中去除。最后对上述数据预处理的函数进行封装以便后续建模的调用。

六、问题的求解过程

6.1 TF-IDF 算法

TF-IDF 是一种用于信息检索与文本挖掘的常用加权技术。是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。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-IDF 的主要思想是：如果某个单词在一篇文章中出现

的频率 TF 高，并且在其他文章中很少出现，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。

（1）词频（TF）表示词条（关键字）在文本中出现的频率。这个数字通常会被归一化（一般是词频除以文章总词数），以防止他偏向长的文件。其公式表达为：

$$tf_{ij}=\frac{n_{i,j}}{\sum_k n_{k,j}} \quad 即 \quad TF_w=\frac{在某一类中词条w出现的次数}{该类中所有的词条数目}$$

（2）逆向文件频率 (IDF)，某一特定词语的 IDF，可以由总文件数目除以包含该词语的文件的数目，再将得到的商取对数得到。如果包含词条 t 的文档越少，IDF 越大，则说明词条具有很好的类别区分能力。其公式表达为：

$$idf_{ij}=\log\frac{|D|}{|\{j:t_i\in d_j\}|}$$

但如果该词语不在语料库中，就会导致分母为零，因此一般情况下，为避免分母为 0，其公式表达为：

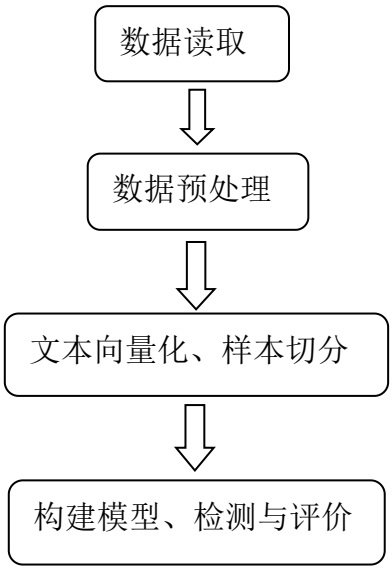
$$IDF=\log\left(\frac{语料库的文档总数}{包含词条w的文档数+1}\right)$$

（3）TF-IDF 实际上是 TF*IDF。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。即：

$$TF-IDF=TF*IDF$$

6.2 问题一的求解过程

从原有的群众留言数据中抽取部分数据利用 Python 读取，将抽取到的数据进行清洗，去除重复无用异常问题的数据，然后将得到的文本数据进行分词，并转化为词语，然后得到的词语转达成数值的向量化形式，再构建高斯朴素贝叶斯模型的分类方法，最后对得到的分类方法利用 F-Score 进行评价得到方法准确率在 69%到 71%之间。



6.2 问题一求解流程图

6.2.1 数据读取

首先在 Python 中建立一个脚本导入 pandas 库实现数据的读取操作。将事先获取的附件 2 的 Excel 文件转化为 csv 文件，使文本数据以列表形式并每一行的多元素是用逗号隔开的数据的文本读取。然后用 pandas 库的读写函数将 cvs 文件读入 DataFrame，读取之后我们可以查看附件 2 中数据的结构，以及一级标签中分类的“城乡建设”，“环境保护”，“交通运输”等七个“二级标签”的分类样品数量组成。由于我们读取的数据里一级标签中分类的二级标签的留言数分布得并不均匀，所以我们将数据进行过抽样和欠抽样，使一级标签中读取的各类样本留言的数量均衡，并将抽样后的数据进行合并。

6.2.2 数据预处理

利用 Python 首先将文本数据中的“留言详情”进行去重处理，清除重复数据后，再进行数据清洗、数据分词、去除停用词的处理，具体过程见前面第五节。

6.2.3 文本向量化及样本切分

文本数据权值向量表达需要在 Python 中新建一个脚本来进行模型的构建。

导入上一个脚本数据预处理的函数,将 sklearn.feature_extraction.text 文本特征提取模块导入 CountVectorizer 转化词频向量统计函数和 TfidfTransformer 权值转化为 tf-idf 权值向量函数的类。以及导入切分函数，对全部文本数据以及“一级标签”中的样本共分类内容进行测试集样本占 20%，训练集样本占 80%切分成数据训练集，数据测试集，标签训练集，标签测试集四份，切分后将训练集样本转换成 tf-idf 全值向量，再按照训练集样本形式将测试集样本进行转换进行模型的构建与转换。由于数据的训练集和测试集是独立的行数及列数不一样，它们所提取的 feature 维度不一样所以我们需要将训练集和测试集进行相应的转换使两个 CountVectorizer 共享 vocabulary。

6. 2. 4 模型的构建

首先介绍高斯朴素贝叶斯模型：

贝叶斯公式为：$P(y_i|x)=\frac{P(x|y_i)P(y_i)}{P(x)}$，根据全概率公式 $P(x)=\sum_{1}^{m}P(x|y_i)P(y_i)$ ，展开分母可得：$P(y_i|x)=\frac{P(x|y_i)P(y_i)}{\sum_{1}^{m}P(x|y_i)P(y_i)}$，即已知 x，想要求解 y_i，以及似然度 $P(x|y_i)$，即可求解后验概率 $P(y_i|x)$。但如果 x 是连续变量，则需假设在 y_i 的条件下，x 服从高斯分布（正态分布）。根据正态分布的概率密度函数公式 $P(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{(x-\mu)^2}{2\sigma^2}}$ 即可计算出 $P(x|y_i)$。如果 x 是多维数据，则假设 $P(x_1|y_i),\ P(x_2|y_i),...,P(x_n|y_i)$ 对应的事件是彼此独立的，这些值连乘在一起得到 $P(x|y_i)$，即高斯朴素贝叶斯的原理。

高斯贝叶斯实际上是假设各个属性在各个类别中均服从高斯分布，然后便可以使用极大似然估计得到高斯分布的参数——均值和方差，之后使用概率密度来得到某样本属于各个类别的概率，所以在本题中运用高斯朴素贝叶斯模型。在 Python 中导入的 sklearn 贝叶斯库中导入高斯朴素贝叶斯模型，并调用数据训练集和标签训练集进行训练。

6.2.5 模型的评价

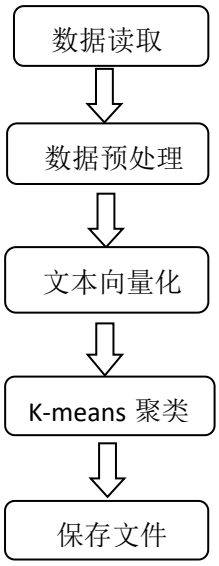
最后对构建并训练好的模型进行检测，并观察训练的模型在测试样本上的表现，对于分类模型的检测与评价我们通常使用 F-Score 对分类方法进行评价：

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

其步骤为：对分类方法进行评价，调用训练好的模型去对输入样本的基变量进行预测然后将预测值和实际值进行比较，最后通过高斯朴素贝叶斯函数得到的模型精度在 69%到 71%之间。

6.3 问题二的求解过程

利用 python 读取附件 3 中“留言主题”一列的全部内容，将读取到的文本数据进行清洗，去掉文本中无意义的留言（例如留言内容是一些标点符号等）。首先是对文本进行分词，再去停用词，最后再将词与词之间用空格隔开并拼接成字符串的形式。通过上述步骤清理过的文本数据已经较为完整且规范，进一步将其转化成向量的形式，再利用 K-means 聚类算法对留言向量进行聚类。聚类时，通过“手肘法”进行 K 值的确定。聚类后，在聚类结果中得出最大的 5 个类，并分别找出距离这 5 个聚类中心最近的留言向量。这 5 个留言向量对应的即是排名前 5 的热点问题。最后，将问题排列好保存为“热点问题表.xls”;将相应热点问题对应的留言信息保存为“热点问题留言明细表.xls”。



6.3 问题二求解流程

6.3.1 数据的读取

首先，在 Python 工程脚本中导入 pandas 库（pandas 是基于 NumPy 的一种工具，提供了大量能使我们快速便捷地处理数据的函数和方法），通过调用 pandas 中 read_excel 函数可以直接读取 excel 文件。由于后续数据处理操作只针对“留言主题”，所以将附件 3 中“留言主题”这一列单独提取出来。

6.3.2 数据预处理

利用 Python 首先对文本数据进行数据清洗、数据分词、去除停用词的处理，具体过程见前面第五节。最后将去停用词后的文本转成字符串的形式：数据去停用词之后是以列表的形式存储。我们需要将列表中的数据转成字符串，以便于后续操作。通过调用 join 函数，将列表中的数据转成以空格作为分隔符串联起来的字符串，形成比较规范的数据格式。

6.3.3 文本向量化

在数据转成字符串形式之后，通过 Python 中 sklearn 的 CountVectorizer 模块中 fit_transfrom 函数来实现一个词袋模型，将文档转换成为特征向量。转

化成的特征向量是一个稀疏的矩阵，我们需要调用 to.array 函数将矩阵转化成数组的形象，再调用 TfidfTransform 模块中 fit_transform 函数进一步转化成 TF-IDF 权值向量。

6.3.4 K-均值聚类

（1）K-均值聚类的原理：

假设有一个包含 n 个 d 维数据点的数据集 $X=\{x_1,x_2,x_3,...,x_i,...,x_n\}$，其中 $x_i \in R^d$，K-均值聚类将数据集 X 组织为 K 个划分 $C=\{c_k,i=1,2,...,k\}$。每个划分代表一个类 c_k，每个类 c_k 有一个类别中心 μ_i。选择欧几里得距离作为相似性和距离判断准则，计算该类内个点到聚类中心 μ_i 的距离平方和：

$$J(c_k)=\sum_{x_i \in c_i}\|x_i-\mu_i\|^2$$

聚类的目标是使各类总的距离平方和 $J(c)=\sum_{k=1}^{k}J(c_k)$，

$$J(c)=\sum_{k=1}^{k}J(c_k)=\sum_{k=1}^{k}\sum_{x_i \in c_i}\|x_i-\mu_i\|^2=\sum_{k=1}^{k}\sum_{i=1}^{n}d_{ki}\|x_i-\mu_i\|^2$$

其中，$d_{ki}\begin{cases}1,若x_i \in c_i\\0,若x_i \notin c_i\end{cases}$，所以根据最小二乘法和拉格朗日原理，聚类中心 μ_k 应该取为类别 c_k 类个数据点的平均值。

（2）K-均值聚类中 K 值的手肘法选取

手肘法的核心指标是 SSE（误差平方和），其公式为：

$$SSE=\sum_{i=1}^{k}\sum_{p \in C_i}|p-m_i|^2$$

SSE 是所有样本的聚类误差，代表了聚类效果的好坏。

手肘法的核心思想是：随着聚类数 k 的增大，样本划分会更加精细，每个簇的聚合程度会逐渐提高，那么误差平方和 SSE 自然会逐渐变小。并且，当 k 小于真实聚类数时，由于 k 的增大会大幅增加每个簇的聚合程度，故 SSE 的下降幅度会很大，而当 k 到达真实聚类数时，再增加 k 所得到的聚合程度回报会迅速变小，所以 SSE 的下降幅度会骤减，然后随着 k 值的继续增大而趋于平缓，也就

是说 SSE 和 k 的关系图是一个手肘的形状，而这个肘部对应的 k 值就是数据的真实聚类数。

在本题中，我们首先通过“手肘法”确定 K 值，在工程中导入 sklearn 库中的 Kmeans 模块，通过调用 kmeans 中的 fit 函数，对向量化的留言进行聚类。再调用 kmeans 中的 pridict 函数，对聚类结果进行预测。其次，导入 matplotlib.pyplot 模块，调用 figure 函数绘制图画框，再通过 scatter 函数对绘制聚类结果的散点图。最终在图中找出最大的 5 个类，调用 kmeans 中的 cluster_centers_函数得到 5 个类中心点。

6.3.5 寻找热点问题

通过自定义函数 disances，求出每个类中距离类中心最近的留言。 其中，函数中距离公式采用的是欧式距离公式。

```
#通过循环找出距离之质心最近的留言在数组中对应的索引值
d = 99
i=0
j=0
for j in range(5):
    print("--------")
    for i in range(len(excel_tifid)):
        a = np.array(excel_tifid[i])
        b = np.array(center[j])
        c = distances(a,b)
        if c < d:
            d = c
            print(i)
```

6.3.5 distance 函数代码实现

6.3.6 定义热度指数

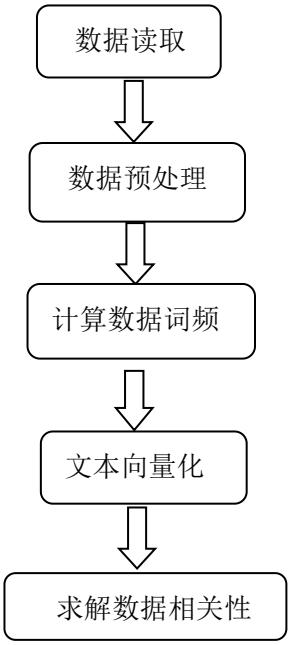
热度指数分为三个等级：*、**、***。

通过上述步骤找出排名前五的热点问题之后，如果问题在所有留言中出现的

次数在 50 次以下，热度指数为“*”；次数在 50-100 次，热度指数为“**”；次数达 100 次以上，热度指数为“***”。

6.4 问题三的求解过程

对于答复的相关性：通过对答复和问题进行量化，构建评价指标反应相关部门回复这个问题的质量如何。对附件 4 中的留言详情和答复进行数据预处理过滤掉无意义的词，取两类数据预处理后得到所有词的词合并成为关键词，将数据量化，找出留言相对应的答复中两者词的相同性和相似性，再通过两个向量的余弦相似度分析评价相关部门对相应留言的答复的质量如何，最终得到相关性约为 0.746。



6.4 问题三求解流程图

6.4.1 数据的读取

在 Python 中导入 pandas 模块进行数据读取，为方便停用词的去除将附件 4 的 excel 文件转化为 csv 类型使文本数据以列表形式并每一行多元素用逗号隔开的数据的文本读取，并读取出转化后文件内的答复建议和留言详情的数据并将形式定为 string 型。

6.4.2 数据预处理

利用 Python 对文本数据进行数据清洗、数据分词、去除停用词的处理，具体过程见前面第五节。

6.4.3 计算数据词频

给留言详情和答复建议分别创建一个字典，遍历 key()中每个词在句子中的出现次数，得出留言详情和答复建议中词语的词频统计。代码实现如下：

```
word_fre1 = {}
for i in data_stop1:
    for j in i:
        if j not in word_fre1.keys():
            word_fre1[j] = 1
        else:
            word_fre1[j] += 1
word_fre2 = {}
for i in data_stop2:
    for k in i:
        if k not in word_fre2.keys():
            word_fre2[k] = 1
        else:
            word_fre2[k] += 1
```

6.4.3 词频统计代码实现

6.4.4 文本向量化

将两类去除停用词和词语的单引号（ ‘’ ）与列表框（[ ]）后的数据进行向量化。我们通过 TF-IDF 将文本型的数据向量化。TFIDF 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，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。

6.4.5 数据的相关性求解

（1）词语相似度：

1.点间互信息（PMI）

点间互信息（PMI）主要用于计算词语间的语义相似度，基本思想是统计两个词语在文本中同时出现的概率，如果概率越大，其相关性就越紧密，关联度越高。两个词语 word1 与 word2 的 PMI 值计算公式如下式所示为：

$$SO-PMI(word1)=\sum_{Pword\in Pwords}PMI(word1,Pword)-\sum_{Nword\in Nwords}PMI(word1,Nword)$$

从中可以看到这个值代表着，x 在 y 出现情况的概率，同时也 y 在 x 出现情况下的概率。

P(word1&word2)表示两个词语 word1 与 word2 共同出现的概率，即 word1 与 word2 共同出现的文档数， P(word1)与 P(word2)分别表示两个词语单独出现的概率，即 word 出现的文档数。若两个词语在数据集的某个小范围内共现概率越大，表明其关联度越大；反之，关联度越小。P(word1&word2)与 P(word1)P(word2)的比值是 word1 与 word2 两个词语的统计独立性度量。其值可以转化为 3 种状态：

P(word1&word2) > 0；两个词语是相关的；值越大，相关性越强。

P(word1&word2) = 0；两个词语是统计独立的，不相关也不互斥。

P(word1&word2) < 0；两个词语是不相关的，互斥的。

当 X,Y 关联大时，MI(X,Y)大于 0；当 X 与 Y 关系弱时，MI(X,Y)等于 0；当 MI(X,Y)小于 0 时，X 与 Y 称为“互补关系”。

2.MI 的应用与延伸：

1）互信息在文本自动分类中的应用，体现了词和某类文本的相关性；

2）新词发现的思路如下：对训练集中的文本进行字频的统计，并且统计相邻的字之间的互信息，当互信息的值达到某一个阀值的时候，我们可以认为这两个字是一个词，三字，四字，N 字的词可以在这基础上进行扩展

3）计算检索的关键字与检索结果的相关性，而这种计算又可以转换为检索的关键字与检索结果的词的相关性计算。此时还是可以使用互信息来进行计算，但是计算的数量要增加不少 ；

4）互信息的缺点是前期预处理的计算量比较大，计算结果会形成一个 big table,当然只要适当调整阀值还是可以接受的。

（2）余弦相似度

余弦相似度是余弦夹角。几何中夹角余弦可用来衡量两个向量方向的差异，机器学习中借用这一概念来衡量样本向量之间的差异。

余弦取值范围为[-1,1]。求得两个向量的夹角，并得出夹角对应的余弦值，此余弦值就可以用来表征这两个向量的相似性。夹角越小，趋近于 0 度，余弦值越接近于 1，它们的方向更加吻合，则越相似。当两个向量的方向完全相反时，夹角余弦取最小值-1。当余弦值为 0 时，两向量正交，夹角为 90 度。因此可以看出，余弦相似度与向量的幅值无关，只与向量的方向有关。在二维空间中向量 A（X1，Y1）与向量 B（X2，Y2）的夹角余弦公式为：

$$\cos\theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^{\,2} + y_1^{\,2}}\sqrt{x_2^{\,2} + y_2^{\,2}}}$$

在本题解决答复的相关性中，利用 Python 建立一个新的字典用来加入各个留言详情与答复建议的相关性。得到 2816 个余弦相似度，将这 2816 个余弦相似度相加除以数据总数，得出附件 4 相关部门对留言的答复意见，从答复的相关性角度对答复意见的相似度，其结果约为 0.746。

七、参考文献

[1] 李光明,潘以锋,周宗萍.基于自然语言处理技术的学生管理论坛文本挖掘与分析[J].智库时代,2019(29):122+127.

[2] 林蔚.文本挖掘在串通投标行为识别中的应用[J].中国内部审计,2019(09):62.

[3] 黄源.k-means 聚类算法介绍及应用[J].科学咨询(科技·管理),2019(10):86-87.

[4] 柏宇轩.Kmeans 应用与特征选择[J].电子技术与软件工程,2018(01):186-187.

[5] 魏烨敏，蒋子元.机器学习聚类组合算法及其应用[J].电子世界,2018(13):102.