关于“智慧政务”中的文本挖掘

摘要

近年来，微信、微博、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。人民群众在相关平台留言记录越来越多，而依靠人工根据经验处理数据，存在工作量大、效率低，且差错率高等问题。这时，为了更好的减少人力物力的资源消耗，通过大数据技术对数据进行处理和文本挖掘显得尤其重要。

首先我们针对题目进行了分析，我们需要对自然语言中的中文文本进行处理，也是数据挖掘中的一类，那么就按照数据挖掘的流程来进行文本挖掘。文本数据挖掘主要分为三步：文本预处理、文本特征提取、分类模型构建。而在第一题的建立留言内容的一级标签也就是多标签分类模型中，文本特征提取可以采用 CNN 中的 LDA，而对于分类模型我们采用传统的文本分类模型，利用朴素贝叶斯模型来对我们处理好的文本进行分类。在这个传统的分类模型中存在着它一定的不足之处。

在第二题当中，对热点问题的挖掘，对数据进行文本预处理后，文本特征提取采用 TF-IDF，进行层次聚类，迭代，归类出相似留言，利用 tensorflow+keras 模型进行命名实体识别，识别出事件的地点，提取时间段，根据群众在某一段时间的点赞数和同一问题留言的条数作为评定指标评定出热点的排名。

对于对留言答复评价，我们需要从相关性、完整性、可解释性方面综合考虑，我们首先对留言的问题以及反馈回复意见进行处理，通过余弦相似度可得出相关部门留言答复和群众留言的一个关系密度。

关键词：数据预处理，LDA，朴素贝叶斯，分类，词向量转化，聚类，cnn 模型，余弦相似度

Abstract

In recent years,WeChat microblogging network such as sunshine hotline ask ZhengPing stage gradually become the government understand the importance of public opinion, gathering intelligence ,condensed bull message record more and more channels to the people in the platform,and rely on artificial according to experience in processing data,there exists large work efficiency is low,and the error rate is high at this point,in order to reduce the manpower and resource consumption,through the big data technology for data processing and text mining is especially important.

First,we analyzed the topic.We need to process the Chinese text in the natural language,which is also a kind of data mining,so we should carry out text mining according to the process of data mining.Text data mining is mainly divided into three steps:text preprocessing,text feature extraction and classification model construction.In the first question,in the establishment of the first level tag of message content,that is,the multi-label classification model,the text feature extraction can adopt the LDA in CNN,while for the classification model,we use the traditional text classification model and the naive bayesian model to classify the text we processed.There are some shortcomings in this traditional classification model.

In the middle of the second question,the hot issues of digging,the data is text preprocessing,text feature extraction using TF-IDF,hierarchical clustering,iterations,identified a similar message,use tensorflow + keras model for named entity recognition,identify the location of the event,extraction time,according to the

number of thumb up in a certain period of time and the same problem message article number as evaluation index to assess the focus in the rankings.

For the comment reply evaluation,we need to from the relevance,integrity,interpretability aspects of comprehensive consideration,we first to the question of the message and feedback comments to deal with,through cosine similarity canbe obtained from the relevant departments message reply and public message of a relationship density.

Key words: Data preprocessing,LDA,naive bayes,classification,word vector transformation,clustering,CNN model,cosine similarity

目录

一、简介

1.1 问题背景及意义

现如今，在互联网影响下，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

面对现代互联网及人工智能时代，面对海量的数据，文本挖掘技术必不可少，大量数据通过大数据处理技术进行相关预处理，然后通过可视化对文本进行深度的挖掘与分析，可得出相关暴露性问题以及能给政府或企业带来具有针对性的解决方案，这对相关部门的发展具有重要意义。

1.2 问题的分析

问题一，利用自然语言处理技术，建立留言内容的一级标签分类模型，首先我们知道三级标签它是从属和包含关系，我们需要对文本内容进行数据预处理和建立 cnn 分类模型，以及多分类转化为多个二分类的问题，还需要对长文本转化为短文本。

问题二，对热点问题的挖掘，根据群众在某一段时间的点赞数和同一问题留言的条数作为评定指标筛选出相似留言，把特定地点或人群的数据通过birch层次聚类。

问题三，针对对留言答复评价，我们需要从相关性、完整性、可解释性方面综合考虑，通过余弦相似度可得出相关部门留言答复和群众留言的一个关系。

二、模型假设

根据平时生活的所见，所闻，本文做出以下假设：

（1）假设本文所有数据真实可靠。

（2）假设模型能根据数据每天的变化能跟得上运行。

（3）假设相关部门能实时根据群众反应随时进行答复更新。

三、问题一的分析与求解

问题一要求我们根据网络问政平台的群众留言，按一定划分体系对留言进行分类，构建关于一级分类的cnn模型。即我们需要对留言详情和留言主题进行预处理，包括分词去停用词，统计词频和做词云，提取关键词，然后根据把附件一“一级分类”为lable，附件二数据为训练样本训练分类模型，模型为cnn多分类。

3.1 数据预处理

3.1.1 剔除属性

我们主要是对留言主题和留言详情进行分析，所以将附件二的“留言ID”、“留言用户”、“留言时间”删除掉，只获取“留言主题”、“留言详情”两列属性。

3.1.2 数据清洗

用正则表达式中findall()方法去掉留言主题和留言详情两列属性特殊符号以及字符串，无意义符号，即一些数字或字母。

3.1.3 分词与去停用词

在数据清洗之后，利用 jieba 这个库来对文本进行处理，其中包括对留言详情内容进行分词，利用停用词库去掉停用词，“停用词”即为一些没有代表性和意义的词，比如“的”，“呢”，“一些”等。这一步是我们做文本处理中较为关键已经重要的一步，直接影响到我们后面建模的效果。

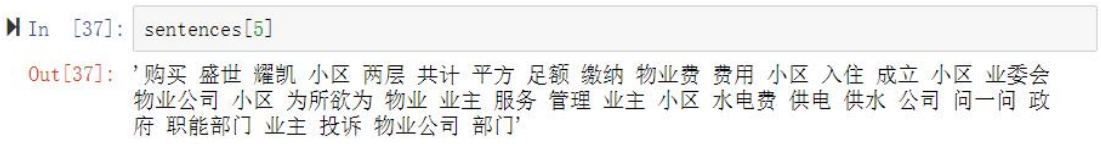


图 3.1.3

3.2 做词云和文本特征选择

用 count（）方法做好词的统计后，然后做成词云。最后利用特征权重主要使用 TF-IDF 方法，提取重要的词，这些词能够代表这段话。在做提取关键词时，可能存在一些词出现频率很高，但不一定是最要的词，相反，一些出现频率虽然不高，但它词的权重比较关键。

3.3 构建模型

提取出关键字后，生成训练集和测试集，抽取特征，利用朴素贝叶斯建立分类模型，利用训练集训练模型，减少真正数据下来时模型所带来的误差。

3.4 模型训练与评价

3.4.1 模型调优

需要继续算法调优，优化特征工程，优化分词、去停用词。可以采用交叉验证登方法来优化我们的数据，以便于的到更优的数据，便于我们训练出更好的模型。

3.4.2 评价分类模型

首先根据我们前面引用的利用传统方法得到的模型以及训练结果来看，准确率达到了 86.01%。

```
In [39]: #抽取特征
from sklearn.feature_extraction.text import CountVectorizer
vec = CountVectorizer(
analyzer = 'word',max_features =4000
)
vec.fit(x_train)
def get_features(x):
    vec.transform(x)
#把分类器improt 进来并且训练
from sklearn.naive_bayes import MultinomialNB
classifilter = MultinomialNB()
classifilter.fit(vec.transform(x_train),y_train)
#查看准确率
classifilter.score(vec.transform(x_test),y_test)
#建立模型

Out[39]: 0.8601823708206687
```

图 3.4.2

从这个准确率我们就可以看出我们利用这个传统方法做出来的分类模型的准确率不是很高，存在一定的弊端。

四、问题二的分析与求解

根据附件 3 需要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，定义合理的评价指标，给出相应的评价结果。主要是需要挖掘留言详情列的相似问题，对相似问题进行聚类。即把相似的留言归为同一问题，可得出排名前五的热点问题，可得出表 1。利用点赞数和留言数量作为热度评价指标，通过用 count（）方法可统计出相对应留言的点赞数和反对数，得出对应的热点问题留言明细表表 2，并且可根据它们的点赞数进行相应评价。

4.1 预处理

4.1.1 数据清洗

将 numpy 类型的数据转化为 list 列表形式的，利用正则表达式 sub()方法去掉留言详情列的一些字符串和一些特殊字符，如‘/s，[]等’。

4.1.2 分词与去停用词

在数据清洗之后，利用 jieba 对留言详情内容进行分词，然后利用停用词库去掉停用词，“停用词”即为一些没有代表性和意义的词，比如“的”，“呢”，“一些”等。

4.1.3 简单的抽取式文本摘要

为了使计算更简单化，以及提高数据的质量，利用 TextRank 进行简单的文本摘要，jieba 分词且去停用词，nltk 统计词频，获取词频最高的前 N 个词，根据最高的 n 个关键词，给句子打分，然后利用均值和标准差过滤非重要句子，最后输出文本的摘要。

文件保存在 FF 文件夹里：如图下

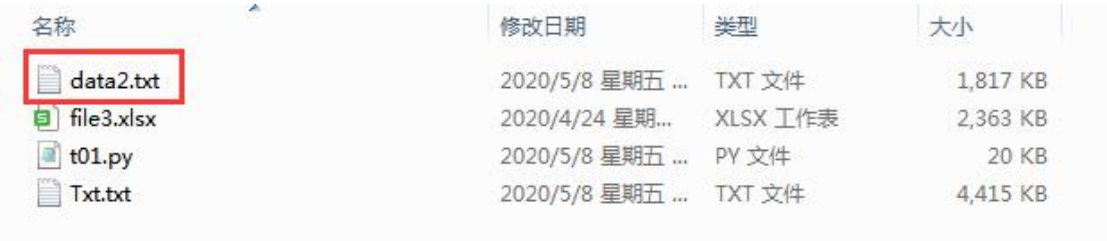


图 4.1.3

4.2 特征提取

4.2.1 TF-IDF 计算词语权重

一个词的权重由 TF * IDF 表示，其中 TF 表示词频，即一个词在这篇文本中出现的频率；IDF 表示逆文档频率，即一个词在所有文

本中出现的频率倒数。因此，一个词在某文本中出现的越多，在其他文本中出现的越少，则这个词能很好地反映这篇文本的内容，权重就越大。

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{文章的总词数}}$$

$$\text{逆文档频率(IDF)} = \log\left(\frac{\text{语料库的文档总数}}{\text{包含该词的文档数} + 1}\right)$$

在对留言详情列分词和去停用词的基础上，使用 tfidf 计算词语权重，将文本中的词语转换成词频矩阵，矩阵元素 a[i][j] 表示 j 词在 i 类文本下的词频，统计每个词语 tfidf 权值，第一个 fit_transform 是计算 tf-idf ，第二个 fit_transform 是将文本转为词频矩阵，获取词袋模型中的所有词语，将 tf-idf 矩阵抽取出来，元素 w[i][j]表示 j 词在 i 类文本中的 tf-idf 权重，对生成的 tfidf 矩阵做 PCA 降维，并返回降维后的数据。

4.4 地点和人群识别

命名实体识别（Named Entity Recognition，NER）是 NLP 里的一项很基础的任务，就是指从文本中识别出命名性指称项，为关系抽取等任务做铺垫。狭义上，是识别出人名、地名和组织机构名这三类命名实体（时间、货币名称等构成规律明显的实体类型可以用正则等方式识别）。当然，在特定领域中，会相应地定义领域内的各种实体

类型。

我们基于keras与keras-contrib构造biLSTM+CRF的命名实体标注模型，识别出事件的地点。使用Bakeoff-3评测中所采用的的BIO标注集，即B-PER、I-PER代表人名首字、人名非首字，B-LOC、I-LOC代表地名首字、地名非首字，B-ORG、I-ORG代表组织机构名首字、组织机构名非首字，O代表该字不属于命名实体的一部分。

以句子为单位，将一个含有 n 个字的句子（字的序列）记作

$$x = (x_1, x_2, \ldots, x_n)$$

其中 x_{i} 表示句子的第 i 个字在字典中的id，进而可以得到每个字的one-hot向量，维数是字典大小。

模型的第一层是 look-up 层，利用预训练或随机初始化的embedding矩阵将句子中的每个字 x_{i} 由one-hot向量映射为低维稠密的字向量（character embedding）$\boldsymbol x_{i}\in\mathbb R^{d}$ ，d 是embedding的维度。在输入下一层之前，设置dropout以缓解过拟合。

模型的第二层是双向LSTM层，自动提取句子特征。将一个句子的各个字的char embedding序列 $(\boldsymbol x_{1},\boldsymbol x_{2},...,\boldsymbol x_{n})$ 作为双向LSTM各个时间步的输入，再将正向LSTM输出的隐状态序列 $(\overset{\longrightarrow}{\boldsymbolh_1},\overset{\longrightarrow}{\boldsymbol h_2},...,\overset{\longrightarrow}{\boldsymbol h_n})$ 与反向LSTM的 $(\overset{\longleftarrow}{\boldsymbol

h_1},\overset{\longleftarrow}{\boldsymbolh_2},...,\overset{\longleftarrow}{\boldsymbol h_n})$ 在各个位置输出的隐状态进行按位置拼接 $\boldsymbol h_{t}=[\overset{\longrightarrow}{\boldsymbol h_t};\overset{\longleftarrow}{\boldsymbol h_t}]\in\mathbb R^{m}$ ，得到完整的隐状态序列

$$(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n) \in \mathbb{R}^{n \times m}$$

在设置 dropout 后，接入一个线性层，将隐状态向量从 m 维映射到 k 维，k 是标注集的标签数，从而得到自动提取的句子特征，记作矩阵 $P=({\boldsymbol p_1},{\boldsymbol p_2},...,{\boldsymbol p_n})\in\mathbb R^{n\times k}$ 。可以把 $\boldsymbol p_i\in\mathbb R^{k}$ 的每一维 p_{ij} 都视作将字 x_{i} 分类到第 j 个标签的打分值，如果再对 P 进行 Softmax 的话，就相当于对各个位置独立进行 k 类分类。但是这样对各个位置进行标注时无法利用已经标注过的信息，所以接下来将接入一个 CRF 层来进行标注。

模型的第三层是 CRF 层，进行句子级的序列标注。CRF 层的参数是一个 $(k+2)\times (k+2)$ 的矩阵 A ，A_{ij} 表示的是从第 i 个标签到第 j 个标签的转移得分，进而在为一个位置进行标注的时候可以利用此前已经标注过的标签，之所以要加 2 是因为要为句子首部添加一个起始状态以及为句子尾部添加一个终止状态。如果记一个长度等于句子长度的标签序列 $y=(y_1,y_2,...,y_n)$ ，那么模型对于句子 x 的标签等于 y 的打分为

$$score(x,y)=\sum_{i=1}^{n}P_{i,y_i}+\sum_{i=1}^{n+1}A_{y_{i-1}y_i}$$

可以看出整个序列的打分等于各个位置的打分之和，而每个位置的打分由两部分得到，一部分是由 LSTM 输出的 $\boldsymbol{p}_i$ 决定，另一部分则由 CRF 的转移矩阵 A 决定。进而可以利用 Softmax 得到归一化后的概率：

$$P(y|x)=\frac{\exp(score(x,y))}{\sum_{y'}\exp(score(x,y'))}$$

模型训练时通过最大化对数似然函数，下式给出了对一个训练样本 (x,y^{x}) 的对数似然：

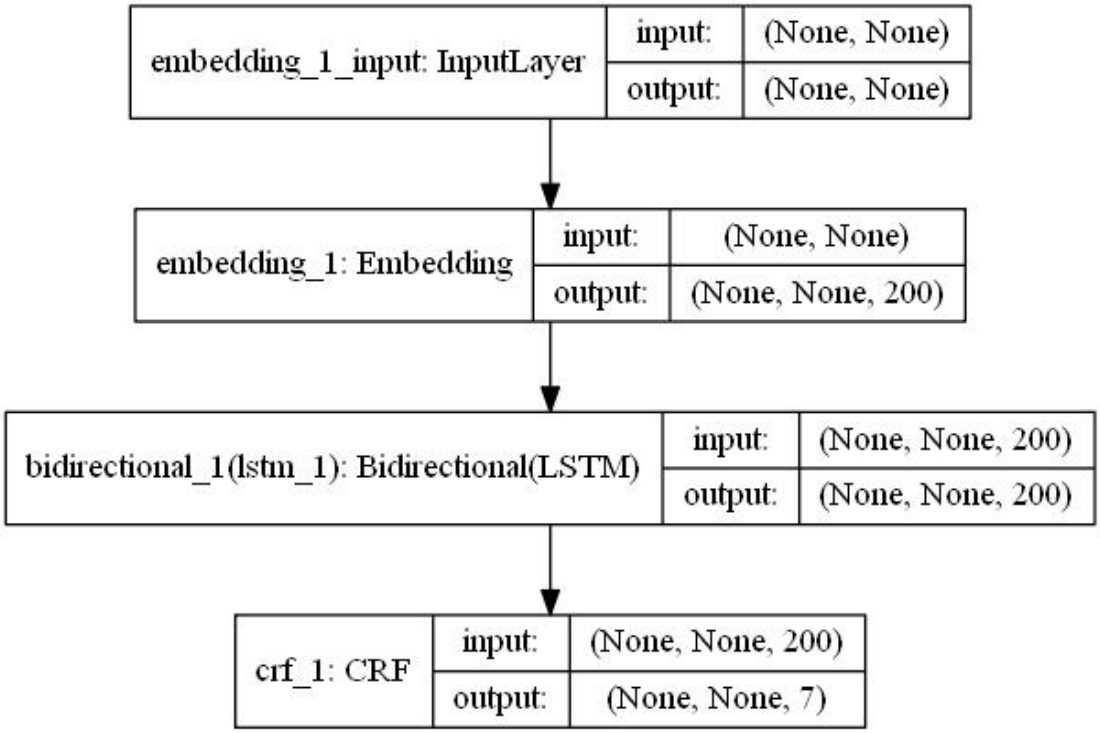
$$\log P(y^x|x)=score(x,y^x)-\log(\sum_{y'}\exp(score(x,y')))$$

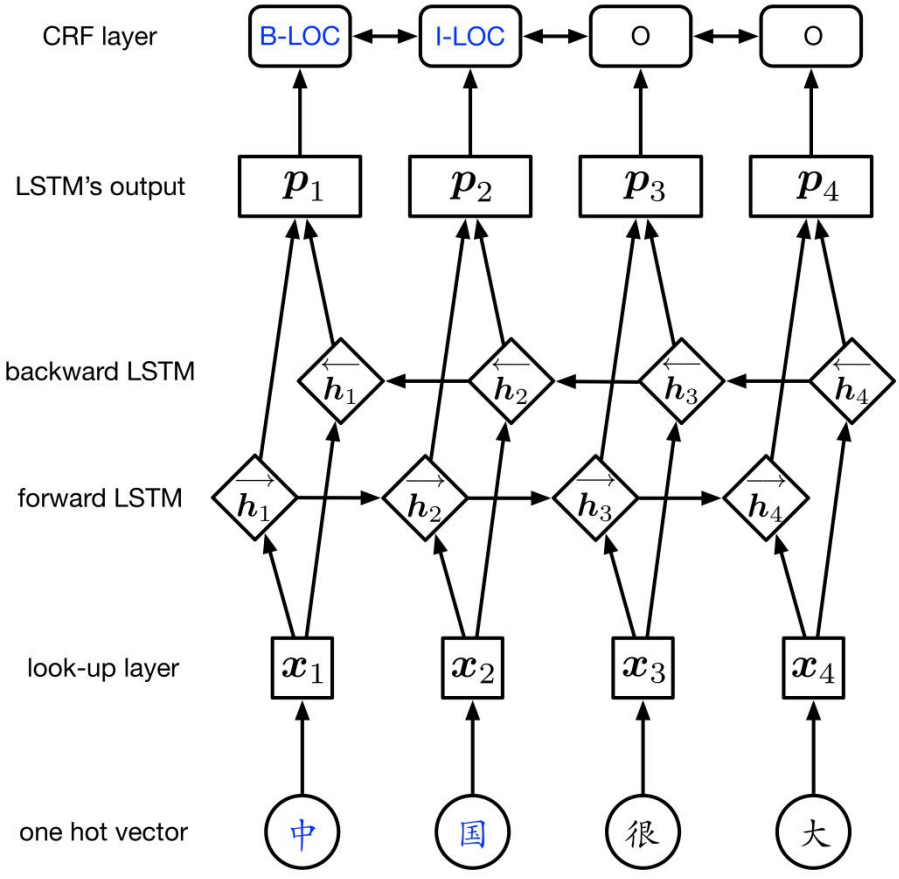
如果这个算法要自己实现的话，需要注意的是指数的和的对数要转换成 $\log\sum_i\exp(x_i)=a+\log\sum_i\exp(x_i-a)$（TensorFlow 有现成的 CRF），在 CRF 中上式的第二项使用前向后向算法来高效计算。

模型在预测过程（解码）时使用动态规划的 Viterbi 算法来求解最优路径：

$$y^*=\arg\max_{y'}score(x,y')$$

整个模型的结构如下图所示：





实验结果:

4.5 构建 birch 层次聚类模型

4.5.1 birch 思想

Birch 算法适合数据量大，类别也比较多。它运行速度很快，只需要单遍扫描就能聚类。Birch 算法是利用一个树结构来快速聚类，这个数结构类似于平衡 B+树，一般称为聚类特征树，这颗树的每一个节点是由若干个聚类特征组成。

利用 birch 将数据簇的个数，并且返回聚类的结果。

4.5.2 对生成的 tfidf 矩阵做 PCA 降维

①数据在低维下更容易处理、更容易使用;

②相关特征，特别是重要特征更能在数据中明确的显示出来；如果只有两维或者三维的话，更便于可视化展示;

③去除数据噪声

④降低算法开销

4.5.3 计算轮廓系数

轮廓系数（Silhouette Coefficient），是聚类效果好坏的一种评价方式。轮廓系数的值是介于 [-1,1] ，越趋近于 1 代表内聚度和分离度都相对较优。

计算簇内不相似度 a(i)：i 向量到同簇内其他点不相似程度的平

均值，体现凝聚度

计算簇间不相似度 b(i)：i 向量到其他簇的平均不相似程度的最小值，体现分离度

那么第 i 个对象的轮廓系数就为：

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

将所有点的轮廓系数求平均，就是该聚类结果总的轮廓系数

4.5.4 画图与保存聚类结果

实验结果

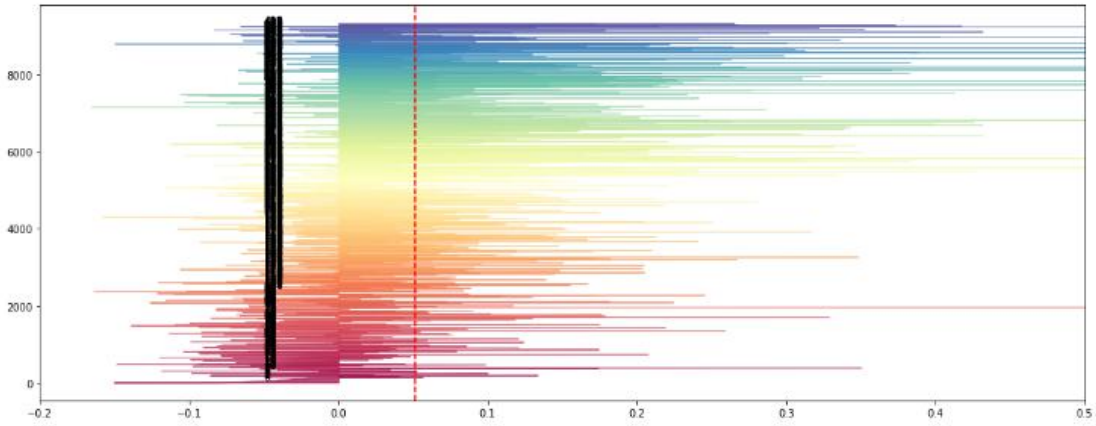


图 4.5.4

4.6 模型调参与评价

通过计算轮廓系数，为了能更好的展现，所以画出图。用 for 循环遍历刚才的聚类。发现将原始矩阵降维，降维后效果反而没有不降维的好。

4.7 定义热度评价指标和评价结果

热度评价根据聚类后同一问题的留言数，点赞数衡量。即用户留言总条数多，点赞多，排名就靠前，反之排名较低。即可排名确定得出排名前五的出热点问题，如下图表 1 部分截图：

通过层次聚类，将相似的留言问题归为同一问题，并返回问题 ID，即得出“热点问题留言明细表”，即表 2，如下部分截图

五、问题三的分析与求解

首先根据基于问题一对数据预处理的方法，即分词、去停用词、统计词频、提取关键字的基础上，将关键字转化为向量文本，然后用余弦对它进行相关性的分析，如果余弦值越接近 1，则答复意见和留言详情越相似，他们之间越有关联。所以基本思路就是：1、分词；2、列出所有词；3、分词编码；4、词频向量化；5、套用余弦函数计量两个句子的相似度。

5.1 预处理

5.1.1 分词与去停用词

利用问题一的方法对附件 4 留言详情和留言答复两列属性进行分词和去停用词，结果如下图所示：

```
In  [29]: s1_cutwords

Out[29]: ['2019',
          '以来',
          '位于',
          'A2',
          '桂花',
          '街道',
          'A2',
          '公安分局',
          '宿舍区',
          '景蓉华苑',
          '出现',
          '一番',
          '乱象',
          '小区',
          '物业公司',
          '美顺',
          '物业',
          '扬言',
          '退出'
```

图 5.1.1

5.1.2 列出所有词

在去停用词后，列出留言详情和留言答复所用词，并将所有词保存在 dict()字典中。

5.1.3 分词编码和词向量化

对留言详情和留言答复统计出来的词进行编码，即计算出每个分词出现的次数，即可得出留言详情和答复意见两列属性的词向量。

5.2 计算余弦相似度。

将所得到的词向量利用向量余弦函数进行计算，即可得出两列的相似度列表中，如下图所示：

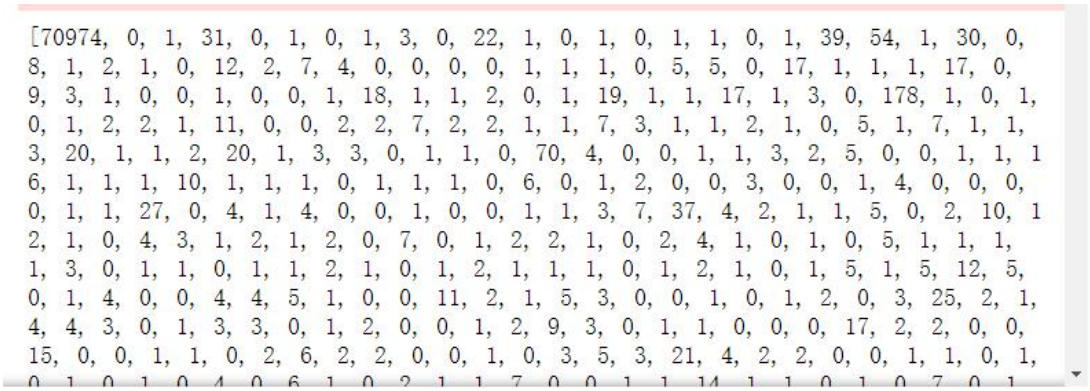


图 5.2

5.3 评价结果

以上余弦相似度结果准确率在 99%，所以相关部门是比较有针对性、完整的回答了群众的留言的。可见，相关部门时刻关注群众所反映的问题，并给予解决，体现了他们承担起了他们的责任。

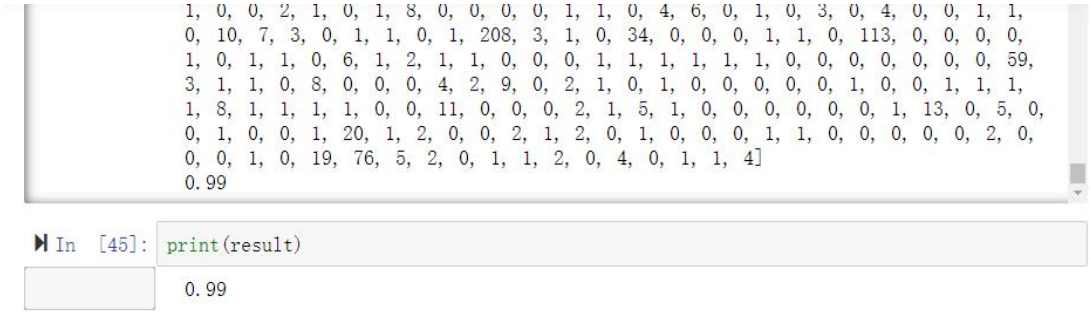


图 5.3

六、模型优化

参考文献

https://blog.csdn.net/qq_34519470/article/details/104854284?utm_source=app
https://blog.csdn.net/qq_42795281/article/details/103752181?utm_source=app
https://blog.csdn.net/qq_25560849/article/details/81288964?utm_source=app
https://www.bilibili.com/video/av17741197?from=search&seid=15493144255300210155