“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。网络问政平台随之兴起，政府也更加了解民心，能更好的为人民服务。因此，分析和挖掘网络数据对网络问政有很重要的意义。

对于问题 1，通过 python 对附件二中留言详情数据进行文本分词，首先用 python 中定义的函数删除字母、数字、汉字以外的所有符号，再加载停用词库，过滤掉停用词，再词频统计选出高频词汇，利用朴素贝叶斯算法确定有效词范围，结合 TF-IDF 算法赋权，定义预测数据函数，构建分类模型，判断其属于一级标签内容。执行预测数据函数后，对比判断对错，用 F-Score 算法对分类模型进行评价。

对于问题 2，通过问题一的方法对附件 3 进行文本分词、停用词处理，选出高频词汇后，利用 TF-IDF 算法计算权重，创建文档词条矩阵，再用 K 均值聚类，每一类为一个热点问题，且时间范围就是这类问题中问题时间总范围，建立起以聚类问题中留言个数与点赞数两个关键因数的热度评定指标，对附件三中热点问题按照热度指数从大到小顺序排序，排出排名前 5 的热点问题和相应热点问题对应的留言信息。

对于问题 3，根据附件一中三级标签体系分类，应用问题一的方法，反复循环三次，对群众留言问题分类，对于群众的留言，分配至相关部门后，相关部门进行答复，结合附件 4 中具体的答复，从相关性、完整性、可解释性等方面，对回复意见建立起分类评价方案。并结合附件 4 中数据进行具体的答复意见评价。

【关键词】：文本分词　朴素贝叶斯算法　TF-IDF 算法　F-Score 算法　K 均值聚类

Abstract

In recent years, with the WeChat, weibo, mayor's mailbox, sunshine hotline and other network political platform gradually become an important channel for the government to unders tand public opinion, gather people's wisdom, and condense people's spirit. With the rise of the network political platform, the government has become more aware of the people an d can better serve the people. Therefore, the analysis and mining of network data is of great signi ficance to network politics.

For question 1, through the python out message details data text participle, first defined in python function to delete all other than letters, Numbers, Chinese character symbol, reload the stop library, filters out the stop, and then high frequency vocabulary, word frequency statistics to select the naive bayes algorithm is used to determine the effective scope of word, empowerment, TF - IDF algorithm combining with definition forecast function, constructing classification model, determine its belongs to level 1 label content. After the prediction data function is executed, the right and wrong are judged by comparison, and the classification model is evaluated by f-score algorithm.

For question 2, through the problem a method of attachment 3, stop words processing, word segmentation in Chinese text selected after high frequency vocabulary, TF - IDF algorithm is used to calculate the weights, create a document entry matrix, then used k-means clustering, every kind is a hot issue, and the time range is the problem in this kind of problem always time range, set up with the number of messages in the clustering problem with the heat of the thumb up for two key factor evaluation indicators, to annex 3 hot issues according to the heat index from big to small order, get rid of the top five hot issues and corresponding hot issues the corresponding message information.

For question 3, according to the attachment 1) tag system classification, application problem a method, repeated cycle three times, leave a message to the crowds, question classification for the comments of the masses, assigned to related departments, the relevant departments to reply, according to detailed answer in the appendix 4, from the relevance, integrity and interpretability advice to reply to establish a classification evaluation scheme. The specific response comments were evaluated in accordance with the data in annex 4.

[Key words]: Text segmentation Naive bayes algorithm TF-IDF algorithm F-Score algorithm K means clustering

目录

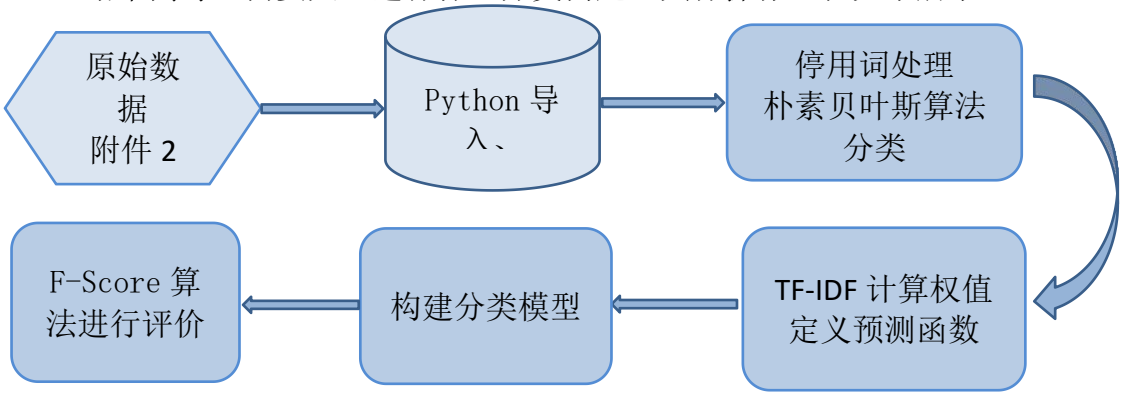
一、问题分析

1.1 问题背景分析

截至 2020 年 3 月，我国网民规模为 9.04 亿，较 2018 年底增长 7508 万，互联网普及率达 64.5%，较 2018 年底提升 4.9 个百分点。科技的发展，互联网发展进入大数据时代，信息传播更具有时效性、智能化和移动化。近年来，微信、微博、市长信箱、阳光热线等网络问政平台成为公民和政府沟通的桥梁，提高了公民政治参与率。政府也因此更加的了解民意、汇聚民智、凝聚民气；并且关于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。何祖坤是我国最早研究政府回应的学者，后续又有李伟权、蒋晓萍、俞可平等人研究，主张政府应建立积极主动地回应路径形成良好的政治互动，即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。

1.2 问题一分析

处理网络问政平台的群众留言时，首先按照一定的划分体系对留言进行三级标签体系分类，以便后续相应的职能部门处理。主要依赖于对海量数据的分类，其中文本分类尤为重要。首先统计附件 2 中一级标签的各个类别的数量，其次就是文本分词，这是文本处理中最基本的过程；在分词结果中有一些词语是没有实际的作用，如介词、感叹词等，会干扰高频词汇的选取，需要剔除，于是使用停用词处理；再用朴素贝叶斯算法分类，TF-IDF 计算权值，构造分类模型，对附件 2 留言进行标签分类，根据 F-Score 对分类结果进行评价。从而解决了效率低，且差错率高等且需要人工进行标签分类问题。具体操作过程如下所示。



1.3 问题二分析

为了及时发现热点问题，我们需要合理定义的热度评价指标，其决定因素分为两大类，第一是同问题的留言条数，另一个是点赞数；对于这两个因素需要我们定义权重。首先需要对留言进行标签分类，有利于统问题 ID；其次利用 TF-IDF 算法计算特征词权重，与附件 3 中留言进行匹配及计算相似度，得到文本词条矩阵；最后使用 K 均值聚类算法计算，得到的聚类结果与点赞数进行最后的权重计算得到最终的热度排名。经过这一过程计算，更能及时的发现热点问题，有利于问题的及时解决。

1.4 问题三分析

为建立起对相关部门对留言答复意见的评价方案，首先对附件 4 中答复意见分析，从相关性来看，由问题一可知，群众在留言按照三级标签体系分类后，便会分配至相关部门，由相关部门进行回复，再结合附件四中具体答复意见可知，相关性很高。从完整性来看，基本所有的留言都有具体详细的回复，有相关工作人员进行核实后，解决相关问题或提出建议予以回复，完整性很高。从可解释性来看，可以建立问题解决与否的评定标准。考虑到这些因素后，可构建评价方案，求出具体系数，予以评价。

二、数据准备

2.1 python 读取 excel 文本文件

利用 python 中 pandas 库读取 excel 文件参数，并用 drop 函数删除附件 2 中“留言编号”、“留言用户”、“留言主题”、“留言时间”、“一级标签”等列的内容，只保留“留言详情”列具体内容作为评判依据。

2.2 jieba 库中文分词

为了方便统计分析，利用 python 读文本文件，再利用 jieba 进行中文分词处理，将关键词汇提取出来，并用定义的函数删除字母、数字、汉字以外的所有的符号。

2.3 将频繁出现的领域干扰词构建停用词库

将文本分词后的高频词汇进行统计并整理出来，从大量的数据中我们不难发现，除了描述标签特征的词汇外，同时也会存在许多出现频率很高但是却没有实际意义的词汇，没有鲜明的特征，也没有很高的辨识度，不能使我们快速辨析出

各类标签的不同之处，体现不出分类标签的具体内容，甚至会对我们的判断产生干扰。比如在城乡建设分词中：我们、没有、问题、领导、政府、部门、一个、现在、相关、进行、可以等等词汇，均对描述城乡建设问题不能起到判断作用，对于这一类词语，我们把它规定为干扰词，并加入停用词库中，结合网上做找停用词表，构建本论文的停用词库。

2.4 词频统计

过滤掉停用词后，遍历所有词语，统计出高频词汇，作为后续朴素贝叶斯算法和预测函数的依据。

三、模型假设

1、假设附件 2 中一级分类均正确

2、假设停用词库处理无缺漏

3、假设特征词选取均适用

4、假设涉及计算部分无误差

四、任务一

4.1 朴素贝叶斯算法分类

朴素贝叶斯算法是以贝叶斯定理为基础的一种算法。贝叶斯定理即：

$$\mathrm{P}(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

即事件 A 发生的前提下事件 B 发生的概率。朴素贝叶斯算法就是基于此，对于给出的待分类项，求解该项出现的条件下各个类别出现的概率，哪个最大，就将此待分类项归为哪个类别。

4.2 TF-IDF 计算权值

在朴素贝叶斯算法分好类的基础上，利用 TF-IDF 计算权值，将其相似词汇归为一类。

4.3 定义预测函数与构建分类模型

定义预测函数 myPredict，根据附件 2 中一级标签构建分类模型，并计算预测数据正确概率为 0.7216673903603995，实际验证，带入附件 2 中留言编号 160873、185685、248924 三个用户留言详情数据，得到如图 1 所示结果：

```
0.7216673903603995
城乡建设
劳动和社会保障
教育文体
```

图 1　详情数据

而留言编号 160873、185685、248924 三个用户的留言问题实际分别为城乡建设、交通运输、教育文体，与我们预测不符，这也验证了我们预测函数精准率为 0.72 左右。

4.4 F-Score 算法对分类模型评价

F-Score 又称 F-Measure，它是 Precision(精准率)和 Recall（召回率）加权调和平均是信息检索领域常用的一种评价标准，也常用来评价一个分类模型的好坏。本文就是采用 F-Score 进行模型评价。

F-Score 的公式是：

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

其中 P 是精准率，也称查准率，即针对预测结果而言，正确预测为正的占全部预测为正的比例，R 是召回率，也称查全率，即针对原样本而言，正确预测为正的占全部实际为正的比例，β是平衡 Precision(精准率)和 Recall（召回率）的参数，通常有三种取值情况，当$\beta = 1$ 时，Precision 和 Recall 同样重要，当 $\beta > 1$ 时，Recall 比 Precision 重要，当$\beta < 1$ 时，Precision 比 Recall 重要。其中当参数$\beta = 1$ 时，就是最常见的F_1-Score

$$F_1 = \frac{2PR}{P+R}$$

其中

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

$$R = \frac{TP}{TP+FN}$$

其中 TP 为真阳，即判断为真，实际为真，FP 为假阳，即判断为真，实际为假，FN 为假阴，即判断为假，实际为假，也是判断准确。

介于是分类模型

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

其中P_i、R_i分别为第 i 类查准率和查全率。因为附件 2 中一级标签有 7 类，所以这里 $n=7$，再带入具体运算数值得$F_1=83.72\%$。由此可知，本文构建模型判断较为精准。

五、任务二

5.1 特征词权重

为了方便后面聚类分析和 TF-IDF 算法的运用，我们首先用 python 将 excel 中每一行转化为一个 txt 文件。然后将所有转化的 txt 文件命名一个为集合 $\mathrm{D}=\{d_1,d_2,d_3\ldots d_n\}$，其中$d_1,d_2,d_3\ldots d_n$分别对应转化的 txt 文件。将其分词处理后，命名为集合$\mathrm{N}_i=\{n_{i1},n_{i2},n_{i3}\ldots n_{im}\}$通过 TF-IDF 算法对文件中出现的特征词赋权，命名为集合$\mathrm{T}_i=\{t_{i1},t_{i2},t_{i3}\ldots t_{im}\}$其中 $i\in[1,n]$。

5.2 TF-IDF 算法计算权重

TF-IDF 算法全称是（term frequency-inverse document frequency）即词频-逆向文件频率，是一种信息检索和文本数据挖掘的加权算法。TF-IDF 算法的公式是：

$$\mathrm{TF-IDF}=\mathrm{TF}*\mathrm{IDF}$$

其中

$$\mathrm{TF}=\mathrm{TF_w}=\frac{\text{在某一个文件中词条 w 出现的次数}}{\text{该文本中所有的词条数目}}$$

$$\mathrm{IDF}=\log\left(\frac{\text{集合 D 中文件总数}}{\text{包含词条 w 的文件数}}\right)$$

即特征词权重$t_{im}=\mathrm{TF}_{n_{im}}*\mathrm{IDF}$, 表示为第 i 个文件中第 m 个特征词的权重。

5.3 文本词条矩阵

根据 TF-IDF 算法计算出的权重t_{im}，作为特征词的向量：

$$\mathrm{T}_1=(t_{11},t_{12},t_{13}\ldots\ldots t_{1m})$$

$$\mathrm{T}_2=(t_{21},t_{22},t_{23}\ldots\ldots t_{2m})$$

$$\mathrm{T}_3=(t_{31},t_{32},t_{33}\ldots\ldots t_{3m})$$

$$\vdots$$

$$\mathrm{T}_n=(t_{n1},t_{n2},t_{n3}\ldots\ldots t_{nm})$$

联系起 n 个文件的特征词向量，组成一个词条矩阵，据实际运算可知，该矩

阵为稀疏矩阵。

5.4 相似度计算

我们在 TF-IDF 算法基础上，结合余弦相似性，衡量不同文件间的相似性。向量余弦公式为：

$$\cos(\theta) = \cos\left(T_i, T_j\right) = \frac{\sum_{k=1}^{m} t_{ik} \times t_{jk}}{\sqrt{\sum_{k=1}^{m} {t_{ik}}^2} \times \sqrt{\sum_{k=1}^{m} {t_{jk}}^2}}$$

其中T_i为第 i 个文本向量，即$T_i = \{t_{i1}, t_{i2}, t_{i3} \ldots t_{im}\}$，$T_j$为第 j 个文本向量$T_j = \{t_{j1}, t_{j2}, t_{j3} \ldots t_{jm}\}$，$i,j \in [1,n]$。$\cos(\theta)$计算所得值，即为第 i 个文本和第 j 个文本的相似度。

5.5 K-均值聚类

K-均值聚类算法是采取距离作为相似性指标，从而发现所给定数据集的 K 个类，且每个类的中心是由所有值的均值计算得到，所有类都用聚类中心来描述。对于给定的含 n 个 m 维的坐标数据以及分类所得的 K，选用欧氏距离作为相似度指标，使得各类聚类平方和最小，即

$$\mathrm{JL} = \sum_{k=1}^{k} \sum_{i=1}^{n} \|x_i - u_k\|^2$$

其中x_i为数据坐标，u_k为所有数据点所算得均值。结合最小二乘法和拉格朗日中值定理，聚类中心为各数据点均值，为保障算法的收敛，使聚类中心尽可能不变。

算法步骤：

1)选取 K 个对象作为初始化中心。给出迭代终止条件，直到类别变动很小。

2）迭代计算。根据数据对象与聚类中心的欧氏距离，按照距离远近，不断划分到最近的聚类中心中，即组成新的相似的一类。

3）重新计算聚类中心。将新建类别中所有对象对象均值的均值，作为新的聚类中心。

4）判断是否终止。若所有类别变化很小或不变，则输出结果，否则重复 2）3）步骤。

本文对均值聚类算法，利用编程实现，最终将附件三中热点问题分为 60 类。

5.6 建立热度评定指标

根据聚类所得结果分析，影响热点问题排序的主要因素有两个：留言人数、点赞和反对数。本文结合新浪微博热度评价指标体系，将留言人数归为一级指标，点赞数和反对数归为二级指标，并将点赞数和反对数定义为支持度，支持度=点赞数-反对数。参考问题二中给出的热点问题 1 和热点问题 2 的数据，建立综合因子方程：

$$F = 0.92L + 0.08Z$$

其中 F 为热度指数，L 为留言人数，Z 为支持度，带入示例数据，热点问题 1 （A 市 A5 区魅力之城小区临街餐饮店油烟噪音扰民），$L = 4, Z = 1$，算得热度指数 $F = 0.92 \times 4 + 0.08 \times 1 = 3.76$，同理带入热点问题 2，$L = 3, Z = 10$，可得热度指数 $F = 0.92 \times 3 + 0.08 \times 10 = 3.56$，显然 $3.56 < 3.76$，所以该热度评定指标基本合理。再带入聚类所得结果，算出热度指数，从大到小排序，保留前 5，结果如图 2 下：

图 2　热度指数的排名

前五名分别为附件 3 中留言编号 220730、217032、194343、200667、209742 为主的热点问题。热点问题表见附件。

六、任务三

6.1 答复意见设计

沿用问题一、二思想，对附件 4 中答复意见进行中文分词、停用词处理、朴素贝叶斯分类、TF-IDF 赋权等。将答复意见进行分类，并提取出高频词汇，作为分类评价方案依据。

完整性：朴素贝叶斯分类对留言进行分类，得到数值型评论和文本型评论；如果包含数值型评论和文本型评论，则留言的完整性为 1，否则为 0。

相关性：通过朴素贝叶斯分类对文本型评论进行分类，分为“主题词”和“评论词”；得到“主题词”和“评论词”类型矩阵，计算余弦相似度，评论与主题相似度越高，说明评价对主题的质量就越高。

可解释性：就是追踪到数据的来源，统计分词，利用可读性指数 ARI 来计算，公式为：

$$\mathrm{ARI} = 4.71 * \left(\frac{\text{总字数}}{\text{总字数}}\right) + 0.5\left(\frac{\text{总字数}}{\text{总句数}}\right) - 21.43$$

数值越接近我们理解文字的最低程度，说明可解释性越高。

6.2 分类评价方案与实践.

通过对答复意见的文本分类处理，提取出相关性、完整性、可解释性三种指标，并对指标有效性关键词聚类分析，构造分类评价方案。

七、参考文献

[1]石凤贵.基于 TF-IDF 中文文本分类实现[J].现代计机,2020(06):51-54+75.

[2]张亚娜,高子婷,胡溢,杨成.融媒体新闻生产中的中文评论关键词提取[J].人工智能,2020(02):57-66.

[3]钟建,高海洋.一种针对中国移动客服文本的分词方法[J].现代信息科技,2020,4(01):7-8+11.

[4]Sisi Liu,Kyungmi Lee,Ickjai Lee. Document-level multi-topic sentiment classification of Email data with BiLSTM and data augmentation[J]. Knowledge-Based Systems,2020.

[5]王艺颖.朴素贝叶斯方法在中文文本分类中的应用[J].中国高新科技,2019(07):57-60.

[6]杨欣,郭建彬.基于改进 TF-IDF 的百度百科词语相似度计算[J].甘肃科学学报,2019,31(02):143-147.

[7]江俊,黄骅,任条娟,张登辉.基于峰值密度聚类的电信业投诉热点话题检测方法[J].电信科学,2019,35(05):97-103.

[8]朱晓霞,宋嘉欣,孟建芳.基于主题—情感挖掘模型的微博评论情感分类研究[J].情报理论与实践,2019,42(05):159-164.

[9]王巧玲,乔非,蒋友好.基于聚合距离参数的改进 K-means 算法[J].计算机应用,2019,39(09):2586-2590.

[10]乔艳霞,邹书蓉,张洪伟.基于 K-means 的改进差分进化聚类算法[J].四川理工学院学报(自然科学版),2014,27(05):64-67.

八、附录

附录一：问题一代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu May  7 15:10:06 2020

@author: Radien
"""

# -*- coding:utf-8 -*-
import pandas as pd
import jieba as jb
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB

df = pd.read_excel('D:/taidibei/C题全部数据/附件2.xlsx')

df.drop(['留言编号','留言用户','留言主题','留言时间','一级标签
'],axis=1) #当你要删除某一行或者某一列时，用drop函数，它不改变原有的
df中的数据
print("数据总量: %d ." % len(df))


# 添加id
df['cat_id'] = df['一级标签'].factorize()[0]
```

```
cat_id_df = df[['一级标签',
'cat_id']].drop_duplicates().sort_values('cat_id').reset_index(drop=T
rue)
cat_to_id = dict(cat_id_df.values)
id_to_cat = dict(cat_id_df[['cat_id', '一级标签']].values)

# 定义删除除字母,数字，汉字以外的所有符号的函数
def remove_punctuation(line):
    line = str(line)
    if line.strip() == '':
        return ''
    rule = re.compile(u"[^a-zA-Z0-9\u4E00-\u9FA5]")
    line = rule.sub('', line)
    return line
def stopwordslist(filepath):
    stopwords = [line.strip() for line in open(filepath, 'r',
encoding='utf-8').readlines()]
    return stopwords
# 停用词
stopwords = stopwordslist("D:/taidibei/C题全部数据
/chineseStopWords.txt")
# 删除除字母,数字，汉字以外的所有符号
df['clean_review'] = df['留言详情'].apply(remove_punctuation)
# 过滤停用词
df['cut_review'] = df['clean_review'].apply(lambda x: " ".join([w for w
in list(jb.cut(x)) if w not in stopwords]))

df.to_csv('fujian2.txt', header=None, sep=' ', index=False)
print("保存文件成功")
```

```
sourceTxt = 'fujian2.txt'
# 分好词后的文本路径
targetTxt = 'fujian21.txt'
# 对文本进行操作
with open(sourceTxt, 'r', encoding = 'utf-8') as sourceFile,
open(targetTxt, 'a+', encoding = 'utf-8') as targetFile:
    for line in sourceFile:
        word = jb.cut(line.strip(), cut_all = False)
        # 分好词之后之间用空格隔断
        output = ' '.join(word)
        targetFile.write(output)
        targetFile.write('\n')
    print('写入成功!')
    txt = open("fujian21.txt", "r", encoding='utf-8').read()
words = jb.lcut(txt)      # 使用精确模式对文本进行分词
counts = {}      # 通过键值对的形式存储词语及其出现的次数
for word in words:
    if len(word) == 1:     # 单个词语不计算在内
        continue
    else:
        counts[word] = counts.get(word, 0) + 1     # 遍历所有词语，每出
现一次其对应的值加 1
items = list(counts.items())
items.sort(key=lambda x: x[1], reverse=True)     # 根据词语出现的次数进
行从大到小排序
for i in range(10):
    word, count = items[i]
    print("{0:<5}{1:>5}".format(word, count))
```

```
print('遍历高频词完成!!')
# 朴素贝叶斯
X_train, X_test, y_train, y_test = train_test_split(df['cut_review'],
df['cat_id'], random_state=0)
count_vect = CountVectorizer()

model = MultinomialNB()

X_train_counts = count_vect.fit_transform(X_train)

# TF-IDF 计算权值
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
clf = model.fit(X_train_tfidf, y_train)

# 预测数据概率
X_test1 = count_vect.transform(X_test)
a = clf.score(X_test1,y_test)
print(a)



# 预测数据函数
def myPredict(sec):
    format_sec=" ".join([w for w in
list(jb.cut(remove_punctuation(sec))) if w not in stopwords])
    pred_cat_id = clf.predict(count_vect.transform([format_sec]))
    print(id_to_cat[pred_cat_id[0]])

# 传入留言详情即可输出分类  括号内可以自己填补
```

myPredict("""

我们用了一生的积蓄，于 2014 年 2 月 K9 县锦绣豪庭买的商品房。首付等各项费用都交了，贷款也是按时还款。到现在已经五年多了，还没得房产证，好像这房子不是自己的一样，也不能交易。请问是什么原因？如果是楼盘原因，又为什么还不整改？如果是开发商的问题，为什么不给其制裁？拖得太久了，开发商有问题，为什么不强制执行，这样卡不动产权证不能办理，让我们这些老百姓如何是好？一辈子的钱在那，不能交易，也没有证件？

""")

myPredict("""

你好,我是一名的哥,全国各地的 2012 年出租车油补都发放了,为什么 A 市还没动静,就象 2011 年的油补,我到 2012 年 12 月才领到,这是不是有些部门在挪用这笔钱.

""")

myPredict("""

B 市二中学校附近的恒升培训机构老板在家长微信群里宣称，市二中高考 600 分以上的学生都是经过其培训机构培训的，另外还有部分老师在其机构参加有偿补课。请问市二中就靠外面培训吗？学校领导就是这样管理教师的吗？

""")

附录二：问题二代码

```
from sklearn.feature_extraction.text import
CountVectorizer,TfidfTransformer

from sklearn.cluster import KMeans
import pandas as pd

import jieba
import re


df = pd.read_excel('D:/taidibei/C 题全部数据/附件 3.xlsx')
df.drop(['留言编号', '留言主题', '留言用户'], axis=1, inplace=True)


# 定义删除除字母,数字，汉字以外的所有符号的函数
def remove_punctuation(line):
    line = str(line)
    if line.strip() == '':
        return ''
    rule = re.compile(u"[^a-zA-Z0-9\u4E00-\u9FA5]")
    line = rule.sub('', line)
    return line


def stopwordslist(filepath):
    stopwords = [line.strip() for line in open(filepath, 'r',
encoding='utf-8').readlines()]
    return stopwords
```

```
# 加载停用词
stopwords = stopwordslist("D:/taidibei/C题全部数据
/chineseStopWords.txt")

# 删除除字母,数字，汉字以外的所有符号
df['clean_review'] = df['留言详情'].apply(remove_punctuation)

# 词频
vectorizer = CountVectorizer()
data = df['clean_review'].values

X = vectorizer.fit_transform([" ".join([b for b in jieba.cut(a)]) for a
in data])
# TF-IDF
tfid = TfidfTransformer()

X = tfid.fit_transform(X.toarray())


# 得到最佳参数=6   K均值最佳分类
km = KMeans(n_clusters=6)
km.fit(X.toarray())
clusters = km.labels_.tolist()
df['class'] = clusters

num = []
for i in range(6):
```

```
    a = df.groupby(['class']).size()[i]
    num.append(a)

df["热度指数"] = 0

for i, item in enumerate(df["class"]):
    df.iloc[i, -1] = 0.92 * num[item] + 0.08*(df["点赞数"][i] - df["
反对数"][i])


a = df.sort_values(by='热度指数',ascending = False)
print(a)
a.to_csv("聚类.csv")
```