“智慧政务”中的文本挖掘应用

摘要

随着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的不断攀升，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。通过建立基于自然语言处理技术的智慧政务系统，对提升政府的管理水平和施政效率具有极大的推动作用。

争对问题一，要建立关于留言内容的一级标签分类模型并且使用 F-Score 对分类方法进行评价。我们首先用 Excel 对一级标签进行处理，然后运用 Anaconda3 中的 Jupyter Notebook 软件对留言主题和留言详情进行处理，之后对处理出来的数据其进行建模，并且评估。

争对问题二，要给出排名前 5 的热点问题和给出相应热点问题对应的留言信息，热点的判定有多种，我们首先运用 Excel 对日期进行处理，之后运用 Excel 的求和对每一条留言的反对数和点赞数进行求和，并且降序，取出排名前 5 条信息。接着，我们运用 Anaconda3 中的 Jupyter Notebook 软件塞选出关键词，接着运用 Excel 表格对关键词进行筛选，提取出排名前 5 条信息。

对于问题三，我们首先对附件四进行数据预处理，运用 Excel 对其进行处理

筛选出时间正确的数据，然后再对其进行判断是否与留言详情有相关性、完整性、可解释性。

关键词： Excel　　数据处理　　Jupyter Notebook　　筛选

一、问题重述

随着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。请利用自然语言处理和文本挖掘的方法解决下面的问题。

问题一，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，同时对该分类方法进行评价。

问题二，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，按表 2 的格式给出相应热点问题对应的留言信息。

问题三，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。

二、符号说明

符号	含义	备注
precision	精确率	
recall	召回率	
f1-score	F1 值	
id	留言编号	
labels	一级分类标签	
content	留言主题	

三、模型假设

四、模型的建立与求解

4.1 问题一模型的建立与求解

4.1.1 数据的预处理

首先，我们运用了 Excel 软件对附件一的一级标签进行标住

一级分类	labels
城乡建设	0

党务政务	1
国土资源	2
环境保护	3
纪检监察	4
交通运输	5
经济管理	6
科技与信息产业	7
民政	8
农村农业	9
商贸旅游	10
卫生计生	11
政法	12
教育文体	13
劳动和社会保障	14

接着，我们运用 Excel 软件对附件二的一级标签参考上表数值进行替换，同时将其分为两个表。

4.1.2 对留言主题的建模评估

我们将分类好的 data 表导入 Anaconda3 中的 Jupyter Notebook 软件中如下表：

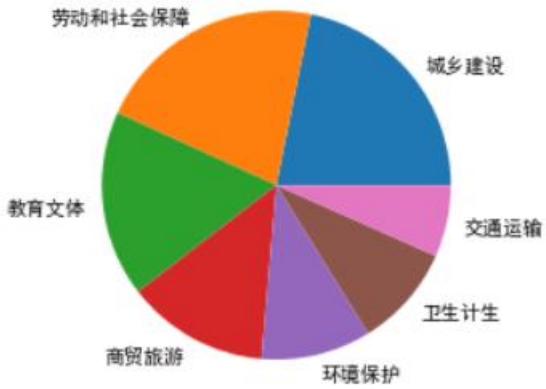
Out[2]:

	id	labels	content
0	24	0	A市西湖建筑集团占道施工有安全隐患
1	37	0	A市在水一方大厦人为烂尾多年，安全隐患严重
2	83	0	投诉A市A1区苑物业违规收停车费
3	303	0	A1区蔡锷南路A2区华庭楼顶水箱长年不洗
4	319	0	A1区A2区华庭自来水好大一股霉味

接着我们运用 Jupyter Notebook 对一级标签进行统计，得到下表：

代表序号	一级标签	数量
0	城乡建设	2008
3	环境保护	938
5	交通运输	613
10	商贸旅游	1215
11	卫生计生	877
13	教育文体	1589
14	劳动和社会保障	1939

我们通过 Jupyter Notebook 绘画出饼图，使得对一级标签的占比了解的更为直观



接着我们对其文本进行处理，删除留言主题中的标点符号，得到下表

```
Out[9]: 0                    A市西湖建筑集团占道施工有安全隐患
        1                  A市在水一方大厦人为烂尾多年安全隐患严重
        2                    投诉A市A1区苑物业违规收停车费
        3                  A1区蔡锷南路A2区华庭楼顶水箱长年不洗
        4                    A1区A2区华庭自来水好大一股霉味
                                      ...
        9205                   两孩子一个是一级脑瘫能再生育吗
        9206      B市中心医院医生不负责任做无痛人流手术后结果还是活胚芽
        9207                   西地省二胎产假新政策何时出台
        9208                       K8县惊现奇葩证明
        9209                 请问J4县卫计委社会抚养费到底该交多少钱
        Name: zhuti, Length: 9210, dtype: object
```

然后，我们对得到的这个数据进行去除停用词和分词，得到下表

```
Out[57]: 0                    [A, 市, 西湖, 建筑, 集团, 占, 道, 施工, 有, 安全隐患]
         1                  [A, 市, 在水一方, 大厦, 人为, 烂尾, 多年, 安全隐患, 严重]
         2                    [投诉, A, 市, A1, 区苑, 物业, 违规, 收, 停车费]
         3                [A1, 区, 蔡锷, 南路, A2, 区华庭, 楼顶, 水箱, 长年, 不洗]
         4                    [A1, 区, A2, 区华庭, 自来水, 好大, 一股, 霉味]
                                          ...
         9205                       [两, 孩子, 一个, 一级, 脑瘫, 能, 生育]
         9206    [B, 市中心, 医院, 医生, 不负责任, 做, 无痛, 人流, 手术, 结果, 活, 胚芽]
         9207                     [西地省, 二胎, 产假, 新, 政策, 何时, 出台]
         9208                         [K8, 县, 惊现, 奇葩, 证明]
         9209                  [请问, J4, 县, 卫计委, 社会, 抚养费, 该交, 钱]
         Name: zhuti, Length: 9210, dtype: object
```

紧接着，我们对该文本进行向量化表示，得到

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

然后我们把该数据分割成三份，两份作为测试数据，一份作为检测数据，这样得出来的模型会更加精准。

4.1.2.1 朴素贝叶斯

我们运用朴素贝叶斯模型分类器对测试数据一、测试数据二、检测数据和全部数据进行运算，得到模型为

```
model_nb = MultinomialNB().fit(cv_train, y_train)
model_nb.score(cv_text, y_text)
```

```
0.827723488961274
```

测试数据一

```
model_nb = MultinomialNB().fit(cv_train1, y_train1)
model_nb.score(cv_text1, y_text1)
```

```
0.7751937984496124
```

测验数据二

```
model_nb = MultinomialNB().fit(cv_train2, y_train2)
model_nb.score(cv_text2, y_text2)
```

```
0.7695290858725762
```

检测数据

```
model_nb = MultinomialNB().fit(cv_data, data['labels'])
model_nb.score(cv_data, data['labels'])
```

```
0.9457111834961998
```

全部数据

接着我们对测试数据一，测试数据二和检测数据进行求平均值，即：
（0.827723488961274+0.7751937984496124+0.7695290858725762）/3
得到其平均值为 0.790815457，这个值同全部数据相对比会较为准确，所以我们后面都不考虑全部数据。

4.1.2.2 KNN

同理我们运用 KNN 分类器对测试数据一、测试数据二、检测数据进行运算，得到模型为

```
model_knn = KNeighborsClassifier().fit(cv_train, y_train)
model_knn.score(cv_text, y_text)
```

```
0.46326456749909517
```

测试数据一

```
model_knn = KNeighborsClassifier().fit(cv_train1, y_train1)
model_knn.score(cv_text1, y_text1)
```

```
0.42997416020671836
```

测试数据二

```
model_knn = KNeighborsClassifier().fit(cv_train2, y_train2)
model_knn.score(cv_text2, y_text2)
```

```
0.4199445983379501
```

检测数据

接着我们对测试数据一，测试数据二和检测数据进行求平均值，得到平均值为：0.437727777

4.1.2.2 SVC

同理我们运用 SVC 分类器对测试数据一、测试数据二、检测数据进行运算，得到模型为

```
model_svc = LinearSVC().fit(cv_train, y_train)
model_svc.score(cv_text, y_text)
```

```
0.8418385812522621
```

测试数据一

```
model_svc = LinearSVC().fit(cv_train1, y_train1)
model_svc.score(cv_text1, y_text1)
```

```
0.8191214470284238
```

测试数据二

```
model_svc = LinearSVC().fit(cv_train2, y_train2)
model_svc.score(cv_text2, y_text2)
```

```
0.8127423822714681
```

测试数据三

接着我们对测试数据一，测试数据二和检测数据进行求平均值，得到平均值为：0.82456747

最后，我们对三个模型的平均数进行对比，得出 SVC 分类器模型的效果更好。但是该分类器运行速度较慢，数据量大的话更建议使用朴素贝叶斯分类器模型。

4.1.2.3 模型评估

我们运用函数 classfication_report 对上面的朴素贝叶斯分类器模型和 SVC 分类器模型进行评估，评估结果如下

	precision	recall	f1-score	support
0	0.92	0.97	0.94	2009
3	0.98	0.95	0.96	938
5	0.99	0.90	0.94	613
10	0.96	0.92	0.94	1215
11	0.97	0.89	0.93	877
13	0.97	0.95	0.96	1589
14	0.91	0.98	0.94	1969
accuracy			0.95	9210
macro avg	0.96	0.94	0.95	9210
weighted avg	0.95	0.95	0.95	9210

朴素贝叶斯分类器模型

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2009
3	1.00	1.00	1.00	938
5	1.00	1.00	1.00	613
10	1.00	1.00	1.00	1215
11	1.00	1.00	1.00	877
13	1.00	1.00	1.00	1589
14	1.00	1.00	1.00	1969
accuracy			1.00	9210
macro avg	1.00	1.00	1.00	9210
weighted avg	1.00	1.00	1.00	9210

SVC 分类器模型

4.1.3 对留言详情的建模评估

运用同样的方法，我们对留言详情进行了处理，得到以下模型

4.1.3.1 朴素贝叶斯

我们运用朴素贝叶斯模型分类器对测试数据一、测试数据二、检测数据和全部数据进行运算，得到模型为

```
#测试数据一
model_nb = MultinomialNB().fit(cv_train, y_train)
model_nb.score(cv_text, y_text)
```

```
0.8574013753166848
```

```
#测试数据二
model_nb = MultinomialNB().fit(cv_train1, y_train1)
model_nb.score(cv_text1, y_text1)
```

```
0.8268733850129198
```

```
#检测数据
model_nb = MultinomialNB().fit(cv_train2, y_train2)
model_nb.score(cv_text2, y_text2)
```

```
0.8094182825484765
```

接着我们对测试数据一，测试数据二和检测数据进行求平均值，得到平均值为：0.831231013

4.1.3.2 SVC

同理我们运用 SVC 分类器对测试数据一、测试数据二、检测数据进行运算，得到模型为

```
#测试数据一
model_knn = KNeighborsClassifier().fit(cv_train,y_train)
model_knn.score(cv_text,y_text)
```

```
0.5678610206297503
```

```
#测试数据二
model_knn = KNeighborsClassifier().fit(cv_train1,y_train1)
model_knn.score(cv_text1,y_text1)
```

```
0.5653746770025839
```

```
#检测数据
model_knn = KNeighborsClassifier().fit(cv_train2,y_train2)
model_knn.score(cv_text2,y_text2)
```

```
0.5307479224376731
```

接着我们对测试数据一，测试数据二和检测数据进行求平均值，得到平均值为：0.554661207

4.1.2.2 SVC

同理我们运用 SVC 分类器对测试数据一、测试数据二、检测数据进行运算，得到模型为

```
#测试数据一
model_svc = LinearSVC().fit(cv_train,y_train)
model_svc.score(cv_text,y_text)
```

```
D:\python\ANACONDA\lib\site-packages\sklearn\svm\_base.py:947: Con
ations.
  "the number of iterations.", ConvergenceWarning)
```

```
0.8718783930510315
```

```
#测试数据二
model_svc = LinearSVC().fit(cv_train1,y_train1)
model_svc.score(cv_text1,y_text1)
```

```
0.8671834625322997
```

```
#检测数据
model_svc = LinearSVC().fit(cv_train2,y_train2)
model_svc.score(cv_text2,y_text2)
```

```
0.8437673130193906
```

接着我们对测试数据一，测试数据二和检测数据进行求平均值，得到平均值为：0.860943057

最后，我们对三个模型的平均数进行对比，得出 SVC 分类器模型的效果更好。但是该分类器运行速度较慢，数据量大的话更建议使用朴素贝叶斯分类器模型。

4.1.3.3 模型评估

我们运用函数 classfication_report 对上面的朴素贝叶斯分类器模型和 SVC 分类器模型进行评估，评估结果如下

	precision	recall	f1-score	support
0	0.92	0.95	0.94	2009
3	0.95	0.99	0.97	938
5	0.98	0.85	0.91	613
10	0.96	0.90	0.93	1215
11	0.96	0.90	0.93	877
13	0.95	0.96	0.96	1589
14	0.93	0.96	0.94	1969
accuracy			0.94	9210
macro avg	0.95	0.93	0.94	9210
weighted avg	0.94	0.94	0.94	9210

朴素贝叶斯分类器模型

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2009
3	1.00	1.00	1.00	938
5	1.00	1.00	1.00	613
10	1.00	1.00	1.00	1215
11	1.00	1.00	1.00	877
13	1.00	1.00	1.00	1589
14	1.00	1.00	1.00	1969
accuracy			1.00	9210
macro avg	1.00	1.00	1.00	9210
weighted avg	1.00	1.00	1.00	9210

SVC 分类器模型

4.2 问题二模型的建立与求解

4.2.2 从点赞数和反对数进行分析

首先，我们运用 Excel 软件对每一条的点赞数和反对数进行相加，在运用倒序取出排名前五的数据，如下表

留言用户	留言主题	留言时间	留言详情	反对数	点赞数	
A00077171	A市A5区汇金路五矿万境K9县存在一系列问题	2019/8/19	的？我们小区也曾发生过狗咬人，请问有人对养宠物的情况进行过排查吗	0	2097	2097
A00087522	反映A市金毛湾配套入学的问题	2019/4/11	市教育局暂未将金毛湾楼盘纳入配套入学，A3区教育局向市教育局汇报，	5	1762	1767
A00031682	请书记关注A市A4区58车贷案	2019/2/21	，焦急万分，急切盼望有案情消息总是失望，四处诉求也无效。此种办	0	821	821
A00056543	严惩A市58车贷特大集资诈骗案保护伞	2019/2/25	还有功；说大股东苏纳和小股东、苏纳弟弟苏吕是挂名；说担保公司法定	0	790	790
A00010616	承办A市58车贷案警官应跟进关注留言	2019/3/1	了希望。但是，A市A4区经侦并没有跟进市领导的留言，案件调查进展	0	733	733

后面仔细观看可知，后三条都是反应同一问题，所以我们对其归为一条，并且对

这三条的点赞数和反对数进行相加，接着，我们从新在提取点赞和反对数量总和排名第六和第七的数据，如下

A00041448	A4区绿地海外滩小区距长赣高铁最近只有30米不到，合理吗？	2019/9/5	境下要好。谢谢领导，敬礼！希望能回复到我如下问题：1、关于高铁，如果从这里过，是否考虑	0	669	669
A00097965	A市富绿物业丽发新城强行断业主家水	2019/6/19	收，应该提供自来水公司发票，物业只提供地摊上买的收据，对于不交水费的，物业私自关二次供	0	242	242

最后我们对其热度指数进行计算，用反对和点赞总数/全部数据，最后得到以下结果：

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	0.54	2019/2/21-2019/3/1	西地省展星投资有限公司设立58车贷	严惩A市58车贷特大集资诈骗案
2	2	0.48	2019/8/19	A市A5区汇金路五矿万境K9县	一系列安全问题
3	3	0.41	2019/4/11	梅溪湖金毛湾	配套入学的问题
4	4	0.15	2019/9/5	A4区绿地海外滩小区	高铁站距离问题
5	5	0.06	2019/6/19	A市富绿物业丽发新城	强行断业主家水

热点问题表

问题ID	留言编号	留言用户	留言主题	留言时间	留言详情	反对数	点赞数
1	220711	A00031682	请书记关注A市A4区58车贷案	2019/2/21 18:45:14	切盼望有案情消息总是失望，四处诉	0	821
1	217032	A00056543	严惩A市58车贷特大集资诈骗案保护伞	2019/2/25 9:58:37	苏纳和小股东、苏纳弟弟苏吕是挂名	0	790
1	194343	A000106161	承办A市58车贷案警官应跟进关注留言	2019/3/1 22:12:30	A市A4区经侦并没有跟进市领导的留言	0	733
2	208636	A00077171	A市A5区汇金路五矿万境K9县存在一系列问题	2019/8/19 11:34:04	发生过狗咬人，请问有人对养宠物的	0	2097
3	223297	A00087522	反映A市金毛湾配套入学的问题	2019/4/11 21:02:44	毛湾楼盘纳入配套入学，A3区教育局	5	1762
4	263672	A00041448	A4区绿地海外滩小区距长赣高铁最近只有30米不到，合理吗？	2019/9/5 13:06:55	希望能回复到我如下问题：1、关于高	0	669
5	193091	A00097965	A市富绿物业丽发新城强行断业主家水	2019/6/19 23:28:27	物业只提供地摊上买的收据，对于不	0	242

热点问题留言明细表

4.2.2 从所提取的关键词进行分析

我们运用 Anaconda3 中的 Jupyter Notebook 软件塞选出关键词，并导出如下表：

镇	57
改造	57
质量	57
违法	54
中心	54
A9	54
大	54

接着运用 Excel 表格对附件三进行关键词塞选，最后我们确定了出现频率最高的有效关键词，如下表所示：

扰民	278
噪音	147
施工	146
地铁	128
车位	91

接着我们运用这些关键词找出相应的留言信息，得到最终结果，如下表所示：

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	0.0642	2019/7/28-2019/9/10	魅力之城小区	扰民问题
2	2	0.0339	2019/11/2-2020/1/25	A市丽发小区	噪音问题
3	3	0.0337	2019/8/22-2020/1/7	A市居民	施工问题
4	4	0.0295	2018/10/27-2020/1/7	A市地铁	地铁问题
5	5	0.0210	2018/7/11-2020/1/6	A市	车位问题

热点问题表

问题ID	留言编号	留言用户	留言主题	留言时间	留言详情	反对数	点赞数
1	360105	A120356	A5区魅力之城小区一楼被搞成商业门面，噪音扰民严重	2019-08-26 08:33:03	主，影响我们的晚年生活。架空层被	1	0
1	360109	A0080252	万科魅力之城小区底层门店深夜经营，各种噪音扰民	2019-09-04 21:00:18	着吆喝声、拼酒声、炒菜烧烤的锅	0	0
1	360100	A324156	魅力之城小区临街门面油烟直排扰民	2019-09-05 12:29:01	人。一天24小时都是烟。请政府关闭	3	0
1	360102	A1234140	A5区劳动东路魅力之城小区底层餐馆油烟扰民	2019-09-10 06:13:27	进屋内，窗户长期不能打开，晚上若	0	0
1	236798	A00039089	A5区劳动东路魅力之城小区油烟扰民	2019/07/28 12:49:18	清洗也没有。每天油烟直排。熏死楼	0	4
1	242792	A909115	A5区魅力之城小区一楼被搞成商业门面，噪音扰民严重	2019/08/26 08:33:03	主，影响我们的晚年生活。架空层被	0	1
1	360101	A324156	A5区劳动东路魅力之城小区油烟扰民	2019-07-28 12:49:18	清洗也没有。每天油烟直排。熏死楼	4	0
1	245136	A909117	万科魅力之城小区底层门店深夜经营，各种噪音扰民	2019/09/04 21:00:18	着吆喝声、拼酒声、炒菜烧烤的锅	0	0
1	195095	A00039089	魅力之城小区临街门面油烟直排扰民	2019/09/05 12:29:01	人。一天24小时都是烟。请政府关闭	0	3

热点问题留言明细表

4.3 问题三模型的建立与求解

4.3.1 数据处理

对于问题三，我们首先对附件四进行数据预处理，运用 Excel 软件中的用 xlsx “=IF(D2>G2,"大","小")”公式区别出答复时间错误的，再”筛选”出答复时间正确的，如下表：

	E	F	G
2241	女结婚未迁户以及其	遇到的各类矛盾和问题，做到群众满意，把好事办好！对个别人地矛盾较为突出的村组，绝大多数群众有意愿，可民	2016/4/1 10:11
2242	卡一直不发，不知什么	，属财政全额拨款，资金确实不足。近两年有5个退休干部需要发放独生子女补贴，该支出项目金额较大且由单位自	2013/8/5 10:11
2243	一直无人接听。请	其深刻检讨、迅速整改。今后将对全局工作人员的组织纪律性提出更高要求，确保不再发生类似事件。给您带来不便，	2019/9/18 17:08
2244	月一交。现在你又说	公积金代扣部门请示协商后，2016年9-12月实现了公积金自助按月缴存。由于受到公积金数据对接、财政代扣工资系	2017/4/6 15:21
2245	里来了，苍蝇泛滥唯	是组织村干部到邻近村、乡镇考察，寻求更好的垃圾处理办法。 三、对于大塘坪村的垃圾处理(即农村清洁	2016/9/9 15:15
2246	交通安全隐患不容	根据管养权限，我局农村公路管理所将在牛首公路上（桃马线交叉出入口）增设相关提示性标志。另因我局无权对杭	2015/9/9 8:51
2247	主反对的情况下依	网友：您好!您所反映的问题，已转交。2019年10月17日	2019/10/17 12:46
2248	要改入户管道得跑	具是否符合《燃烧器适配性目录》进行检查，检查通过后方可通气；燃气具符合目录的用户需要通气，在通过电话向	2019/9/29 8:41
2249	强行在居民楼前修厕	您于2019年6月16日在问政西地省反映的《H3县许家坊乡政府不顾反对，强行在居民楼前修厕所》相关事宜，许家坊	2019/6/24 8:59
2250	化吗？增加了学生	所有财政供养人员均按个人基本工资的5%预算安排了住房公积金。2019年，经县委、县政府研究，决定对全县所有财	2019/3/18 15:41
2251	）只要是个车都能	，且大部分是私家车接入平台后从事营运活动，网约车从业人员约有200多人。但在我县从事网约车经营业务的平台	2018/4/17 18:43
2252	看我家，吵死了，	多数群众的认可和喜爱。近期，正值秸秆禁烧与新农合收缴的政策宣传期，播放频次会多一些。收到群众反映的问题	2019/11/15 14:56
2253	区的政务中心预计何	您好，你所反映的问题已转交相关单位调查处置。	2019/10/22 11:27
2254	我，在这期间，晓	您好，你所反映的问题已转交相关单位调查处置。	2019/10/23 22:32
2255	”改革政策背道而	您好，你所反映的问题已转交相关单位调查处置。	2019/10/23 22:32
2256	公司，想问问咱市有	企业（网店）6500多家，纳入西地省统计监测系统的企业107家，被西地省认定的电商企业33家，现我市没有成规模	2019/7/1 8:12
2257	险办理地点，条件，	财政补助520元，共740元）。参保地点为户籍所在地的社区或者乡镇劳保站。 支付比例：城乡居民医保基金设置住院	2019/6/24 10:36
2258	下高速以后，路上也	路灯安装和接线通电已全部通电亮灯。 二是您反映的森林公园标志牌的问题：按照G1区山大道的设计图纸和道路交	2018/10/30 9:42
2259	个本子和一支笔共	您好，您所反映的问题，已转交相关部门调查处置。	2018/9/28 11:01
2260	游形势，中湖乡全地	出的宝贵建议，现将具体情况回复如下： 2018年，中湖乡修建加油站一事已经在看手规划，由区招商局负责招商引资	2019/1/28 9:32
2261	示。对于周围居民的	您的留言已收悉。关于您反映的问题，已转I1区委、区人民政府调查处理。	2019/1/8 16:54
2262	力和服务表现而言	“UU008194”您的留言已收悉。关于您反映的问题，已转市交通运输局调查处理。	2018/7/4 16:55
2263	要学生购买相关的物	学有所长，有利于培养“合格+特长”的学生。二、经调查，暂未发现文盛小学开展学生社团活动中有违规行为。1.为	2018/10/24 9:22
2264	公路局到底多少费	税资金并不是网民所说的按5%至8%的基数固定递增，而是中央根据上年燃油税实际征收情况及当年燃油税资金增长情	2013/1/6 15:41
2265	路--宁朱公路。201	二五”发展规划实施方案。目前，该项目正在抓紧开展前期工作，根据项目基本建设程序，干线公路建设项目前期工	2012/2/28 10:19
2266			

一共筛选掉了五百多条答复时间小于留言时间的数据，剩下 2265 条数据，接下来我们对留言内容进行筛选，把答复内容敷衍的数据筛选出来

F124

	A	B	C	D	E	F	G	H
1	留言编号	留言用户	留言主题	留言时间	留言详情	答复意见	答复时间	
111	178547	A00024755	政务中心预计何时搬至	2019/10/21 11:06:25	政务中心预	您好，你所反映的问题已转交相关单位调查处置。	10/22 11:27:50	
112	179558	A00081228	天国际旅行社强制性	2019/10/16 18:35:14	在这期间	您好，你所反映的问题已转交相关单位调查处置。	10/23 22:32:29	
113	179560	A0006565	招投标市场乱象何时	2019/10/15 13:37:42	改革政策背	您好，你所反映的问题已转交相关单位调查处置。	10/23 22:32:56	
119								
120								
121								

由以上数据可以看出来，部分地区的部门对群众问题不够上心，态度敷衍

五、程序与算法的评价

5.1 模型的优点

1.为了使模型更加精确，我们对文本进行处理，同时，我们对数据进行细分，在对其进行运算，得到模型。

5.2 模型的缺点

由于在校上课时间不够充足，还有些特征没有得到补充。

六、程序的改进与推广

使用 python 可以提升模型。

参考文献

[1]参考书籍：数字图像处理——Visual C#.NET 编程与实验 孙燮华 编著

附录

问题一代码：（主题）

```
import pandas as pd
data = pd.read_csv('../shuju/data.csv',header=None)
data.columns = ['id','labels','zhuti']
data.head()
num=data['labels'].value_counts()
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']= 'SimHei'
plt.figure(figsize=(4,4))
plt.pie(num,labels=['城乡建设','劳动和社会保障','教育文体','商贸旅游','环境保护','卫生计生','交通运输'])
plt.show()

#文本预处理
import re
tmp = data['zhuti'].apply(lambda x:re.sub('，|。|？|！',',',x))
tmp
import jieba
with open('../Chinese/Stoplist.txt','r',encoding='utf-8') as f:
    stop = f.read()
    stop = stop.split()
    stop = [' ']+stop
jieba.load_userdict('../Chinese/UserList.Txt')
data_cut = tmp.apply(jieba.lcut)
data_after = data_cut.apply(lambda x: [i for i in x if i not in stop])
data_after
tmp = data_after.apply(lambda x: ' '.join(x))
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer().fit(tmp)
cv.transform(tmp)
cv.vocabulary_
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer().fit(tmp)
cv_data = cv.transform(tmp)
cv_data.toarray()
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
```

```
cv_train,cv_text, y_train,y_text = train_test_split(
        cv_data,data['labels'],
        test_size=0.3,random_state=123
)

cv_train1,cv_text1, y_train1,y_text1 = train_test_split(
        cv_train,y_train,
        test_size=0.3,random_state=123
)

cv_train2,cv_text2, y_train2,y_text2 = train_test_split(
        cv_train1,y_train1,
        test_size=0.4,random_state=123
)

model_nb = MultinomialNB().fit(cv_train,y_train)
model_nb.score(cv_text,y_text)

model_nb = MultinomialNB().fit(cv_train1,y_train1)
model_nb.score(cv_text1,y_text1)

model_nb = MultinomialNB().fit(cv_train2,y_train2)
model_nb.score(cv_text2,y_text2)

model_nb = MultinomialNB().fit(cv_data,data['labels'])
model_nb.score(cv_data,data['labels'])

model_knn = KNeighborsClassifier().fit(cv_train,y_train)
model_knn.score(cv_text,y_text)

model_knn = KNeighborsClassifier().fit(cv_train1,y_train1)
model_knn.score(cv_text1,y_text1)

model_knn = KNeighborsClassifier().fit(cv_train2,y_train2)
model_knn.score(cv_text2,y_text2)

model_svc = LinearSVC().fit(cv_train,y_train)
model_svc.score(cv_text,y_text)

model_svc = LinearSVC().fit(cv_train1,y_train1)
model_svc.score(cv_text1,y_text1)
```

```
model_svc = LinearSVC().fit(cv_train2,y_train2)
model_svc.score(cv_text2,y_text2)
from sklearn.metrics import classification_report,confusion_matrix

y_pre_nb = model_nb.predict(cv_data)
print(classification_report(y_true=data['labels'], y_pred=y_pre_nb))

y_pre_knn = model_knn.predict(cv_data)
print(classification_report(y_true=data['labels'], y_pred=y_pre_knn))

y_pre_svc = model_svc.predict(cv_data)
print(classification_report(y_true=data['labels'], y_pred=y_pre_svc))
```

（详情）：

```
import pandas as pd
data = pd.read_csv('../shuju/text.csv',header=None)
data.columns = ['id','labels','zhuti','content']
data.head()
#文本预处理
import re
tmp = data['content'].apply(lambda x:re.sub('，|。|？|！|\\n|\\t','',x))
tmp
import jieba
with open('../Chinese/Stoplist.txt','r',encoding='utf-8') as f:
    stop = f.read()
    stop = stop.split()
    stop = [' ']+ stop
jieba.load_userdict('../Chinese/UserList.Txt')
data_cut = tmp.apply(jieba.lcut)
data_after = data_cut.apply(lambda x: [i for i in x if i not in stop])
data_after
tmp = data_after.apply(lambda x: ' '.join(x))
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer().fit(tmp)
cv.transform(tmp)
cv.vocabulary_
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer().fit(tmp)
cv_data = cv.transform(tmp)
cv_data.toarray()
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
```

```
cv_train,cv_text, y_train,y_text = train_test_split(
        cv_data,data['labels'],
        test_size=0.3,random_state=123
)

cv_train1,cv_text1, y_train1,y_text1 = train_test_split(
        cv_train,y_train,
        test_size=0.3,random_state=123
)

cv_train2,cv_text2, y_train2,y_text2 = train_test_split(
        cv_train1,y_train1,
        test_size=0.4,random_state=123
)

model_nb = MultinomialNB().fit(cv_train,y_train)
model_nb.score(cv_text,y_text)

model_nb = MultinomialNB().fit(cv_train1,y_train1)
model_nb.score(cv_text1,y_text1)

model_nb = MultinomialNB().fit(cv_train2,y_train2)
model_nb.score(cv_text2,y_text2)

model_nb = MultinomialNB().fit(cv_data,data['labels'])
model_nb.score(cv_data,data['labels'])

model_knn = KNeighborsClassifier().fit(cv_train,y_train)
model_knn.score(cv_text,y_text)

model_knn = KNeighborsClassifier().fit(cv_train1,y_train1)
model_knn.score(cv_text1,y_text1)

model_knn = KNeighborsClassifier().fit(cv_train2,y_train2)
model_knn.score(cv_text2,y_text2)

model_svc = LinearSVC().fit(cv_train,y_train)
model_svc.score(cv_text,y_text)

model_svc = LinearSVC().fit(cv_train1,y_train1)
model_svc.score(cv_text1,y_text1)
```

```
model_svc = LinearSVC().fit(cv_train2,y_train2)
model_svc.score(cv_text2,y_text2)
from sklearn.metrics import classification_report,confusion_matrix

y_pre_nb = model_nb.predict(cv_data)
print(classification_report(y_true=data['labels'], y_pred=y_pre_nb))

y_pre_knn = model_knn.predict(cv_data)
print(classification_report(y_true=data['labels'], y_pred=y_pre_knn))

y_pre_svc = model_svc.predict(cv_data)
print(classification_report(y_true=data['labels'], y_pred=y_pre_svc))
```

问题二：

```
# 第一步，导入数据
import pandas as pd
data = pd.read_csv('../aaa/data2.csv',header=None)
data.columns = ['id','content']
data.head()
# 文本预处理
import re
tmp = data['content'].apply(lambda x:re.sub('，|。|？|！|\\n|\\t','',x))
tmp
# 分词和去除停用词
import jieba
with open('../aaa/Chinese/Stoplist.txt','r',encoding='utf-8') as f:
    stop = f.read()
    stop = stop.split()
    stop = [' ']+ stop
jieba.load_userdict('../aaa/Chinese/UserList.Txt')
data_cut = tmp.apply(jieba.lcut)
data_after = data_cut.apply(lambda x: [i for i in x if i not in stop])
data_after
mport matplotlib.pyplot as plt
from wordcloud import WordCloud
import itertools

num = pd.Series(list(
    itertools.chain(*list(data_after)))).value_counts()


num
#导出 csv
num.to_csv('keywords.csv',encoding='utf-8')
```