“智慧政务”中的文本挖掘与应用

摘要

近年来，随着网络的快速发展，微信，微博，市长信箱，阳光热线逐渐成为政府了解民意，汇集民智的重要渠道。因此运用文本分析和数据挖掘技术对提升政府管理水平和施政效率具有极大的推动作用。

对于问题1,利用python读取附件2,获取留言详情作为新的列表。运用jieba分词对留言详情分词，并调用停用词去掉部分分词。用txt文件写入各个一级标签的相关词汇，利用python读取留言详情并将文本向量化，计算每一条留言中包含的各个一级标签相关词汇的频数,将包含词汇频数最高的词汇所属的一级标签定义为该留言的一级标签，运用循环对每条留言都做如此处理，保存为新的列表。计算各个一级标签的查重率和查全率，利用F-score评价该模型。

对于问题2，利用python读取附件3，运用jieba分词对留言主题分词，并调用停用词去掉部分分词。调用gensim分析文本相似度，构建语料库，并通过TF-IDF算法得到前五个热点，去除热点的留言，得到排名前五的热点的位置。使用位置数组构建成新的数组Grid以及Form分别是热点问题表以及热点问题留言细明表，再将他们保存为xlsx文件格式。

对于问题3，利用python读取附件4，将时间统一格式为年/月/日，时：分：秒，并存入一个数组中。将元组中的前4个元素重组为字符串，元组元素分别为年，月，日，时，分，秒，使用正则表达式提取。计算回复与留言的时间差，自定义时效的划分等级以及对应的分数。从时间上进行评判。提取回复内容以及留言内容，建立语料库，利用TF-IDF进行相似度比较，得到相似度的数值，基于两种标准建立模型。

关键词：jieba分词，TF-IDF算法，热点问题；

Text mining and application in "intelligent government affairs

Abstract: In recent years, with the rapid development of the Internet, WeChat, weibo, mayor's mailbox and sunshine hotline have gradually become important channels for the government to understand public opinion and gather people's wisdom.Therefore, the use of text analysis and data mining technology to improve the level of government management and governance efficiency has a great role in promoting.

For question 1, use python to read attachment 2 and get the message details as a new list.Use jieba word segmentation for message details, and call stop words to remove part of the word segmentation.With TXT file is written to the level of vocabulary related to labels, use python to read the message details and the text to quantify, calculate each message contained in each level 1 label related words frequency, will contain the highest frequency vocabulary's primary tag defined as the level of the message, using the loop to do so with every message, save for the new list.The recall and recall rates of each level tag were calculated, and the model was evaluated with f-score.

For question 2, use python to read attachment 3, use jieba word segmentation on the subject of the message, and call the stop word to remove part of the word segmentation.Gensim was called to analyze the text similarity, build the corpus, and obtain the top five hot spots through the tf-idf algorithm, remove the comments of hot spots, and obtain the position of the top five hot spots.Use the position number fabric to build a new array Grid and Form is the hot issues table and hot issues message detail table, and then save them as XLSX file format.

For question 3, use python to read attachment 4, and format the time as year/month/day, hour: minute: second, and store it in an array.The first four elements in the tuple are reorganized into strings. The tuple elements are year, month, day, hour, minute and second respectively.Calculate the time difference between the reply and the message, customize the classification of time and the corresponding score.Judge in terms of time.The reply content and message content were extracted, the corpus was established, the similarity value was obtained by using tf-idf for similarity comparison, and the model was established based on the two standards.

Key words:jieba word segmentation,tf-idf algorithm,hot issues

目录

1、挖掘目标

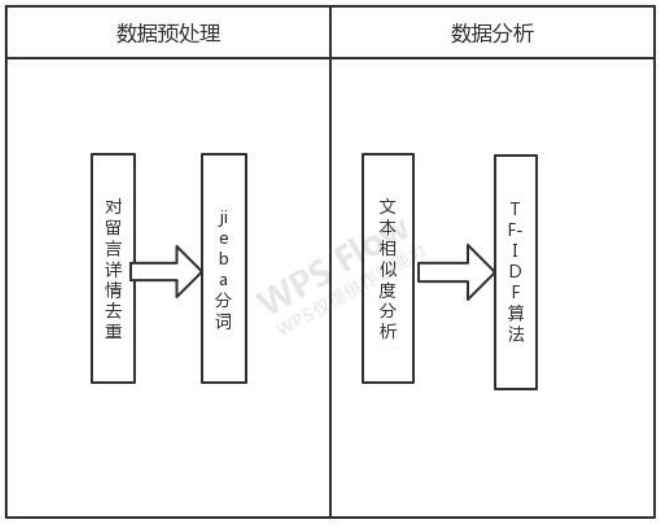
本次建模目标是利用微信微博，市长信箱，阳光热线等网络信息平台的数据，利用 jieba 中文分词工具对留言进行分词，相似度比较，命名体识别，达到以下三个目标：

（1）利用文本分词和文本向量化进行文本挖掘，自动将留言按所属一级标签分类

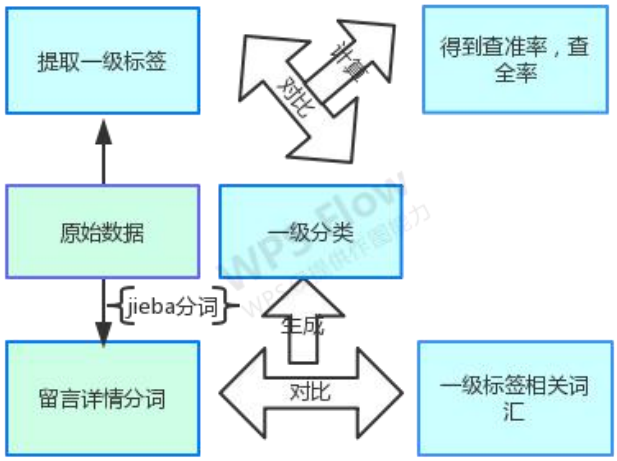
（2）根据附件 3 的数据，分析某一时段的热点问题，了解热点问题的分布。

（3）根据附件 4 的数据，对相关部门对留言的答复情况，从答复的相关性，完整性，可解释性等角度评价回复情况。

2、分析方法与过程



2.1 问题一分析方法与过程



2.1.2 数据处理

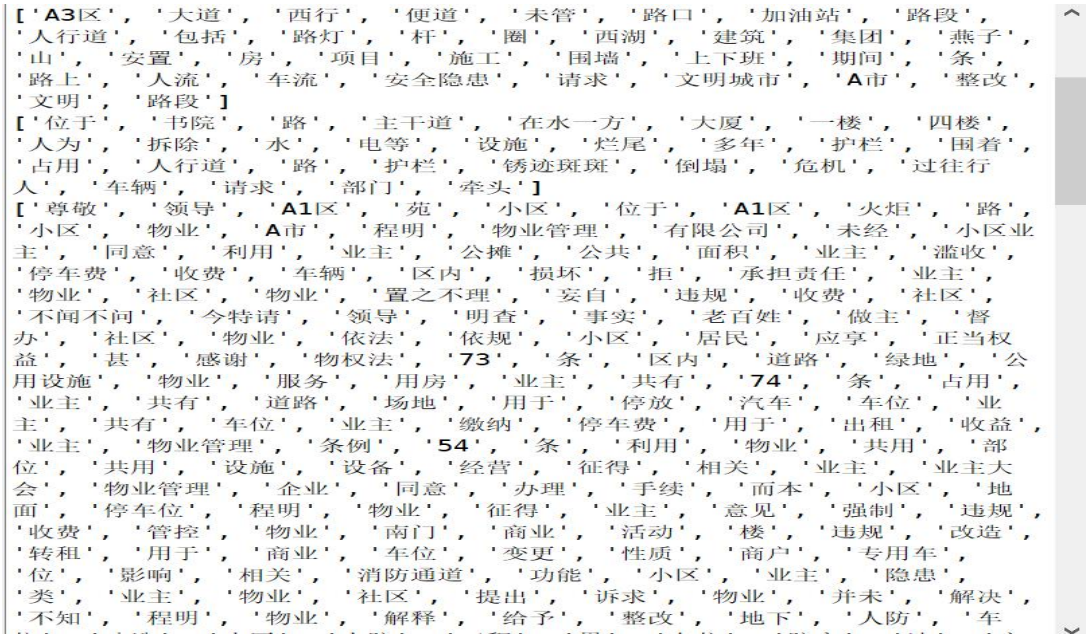
2.1.2.1 对留言详情进行分词

在对留言信息进行数据挖掘的时候，先要把非结构化的文本信息转换为计算机能够识别的结构化信息。在附件 2 中，以中文文本的方式给出了数据，为了方便转化，将数据进行分词。采用 python 的中文分词 jieba 进行分词。jieba 支持三种分词模式：精确模式、全模式和搜索引擎模式。精确模式：试图将语句最精确的切分，不存在冗余数据。全模式：将语句中所有可能是词的词语都切分出来，速度很快，但是存在冗余数据。搜索引擎模式：在精确模式的基础上，对长词再次进行切分[1]

下图为部分分词结果：

A3区	合法	纳税	艺术摄影	咨询	A6区	道路	命名	规划	初步
一米阳光	成果	公示	城乡	门牌	A7县	春华镇	金鼎村	水泥路	婚纱
黄兴路	步行街	古道	巷	住户	卫生间	粪便	外排	A市	A3区
中海	国际	社区	三期	四期	空地	夜间	施工	噪音	扰民
A3区	麓泉社区	改变	麓	谷	明珠	小区	栋	架空层	性质
A2区	富绿	新村	房产	性质	A市	地铁	违规	用工	质疑
A市	路	公交车	随意	变道	通行	A3区	保利	麓	谷林语
桐梓	坡路	与麓	松路	交汇处	地铁	凌晨	点	施工	扰民
A7县	特立	路	东四	路口	晚高峰	太堵	建议	调整	信号灯

2.1.2.2 正则表达式

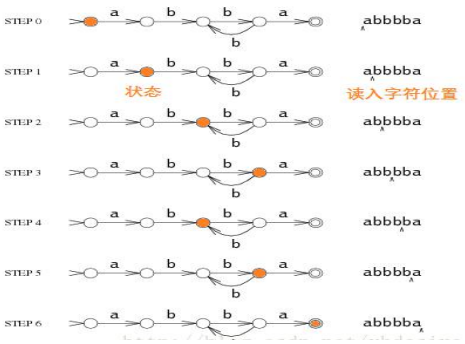


在计算一级分类相关词汇在留言详情中出现的频率，需要使用正则表达式进行计算。正则表达式的算法具体如下：

（1）给定正则表达式，例如：a(bb)+a,其中+表示前面的子表达式一次或多次。所以 abba 或者 abbbba 都能被该表达式匹配。

（2）读入字符串，此时一个状态转换为另一个状态。FSM 有开始和匹配两种特殊状态，分别位于头部和尾部。[2]

（3）进行匹配，下图为进行匹配的示意图：



特别的，当状态机结束于最后一个状态则匹配成功，否则匹配失败。

2.1.3 文本的空间向量模型

由于计算机不能够直接处理文本信息，我们需要对文本进行处理，将文本表示成为计算机能够直接处理的形式，即文本数字化。文本表示也称为文本特征表达，它不仅要求能够真实准确的反映文档的内容，而且要对不同的文档具有区分能力。目前常用的文本表示模型有向量空间模型、布尔模型和概率模型等。[3]这里使用向量空间模型。[4]向量空间模型最早是由 Salton 和 McGill 于 20 世纪 60 年代末提出的，是目前在文本挖掘技术中最常用的表示模型。其主要思想:将每一个文本表示为向量空间的一个向量，并以每一个不同的特征项对应为向量空间中的一个维度,而每一个维的值就是对应的特征项在文本中的权重，这里的权重可以由 TF-IDF 等算法得到。向量空间模型就是将文本表示成为一个特征向量:

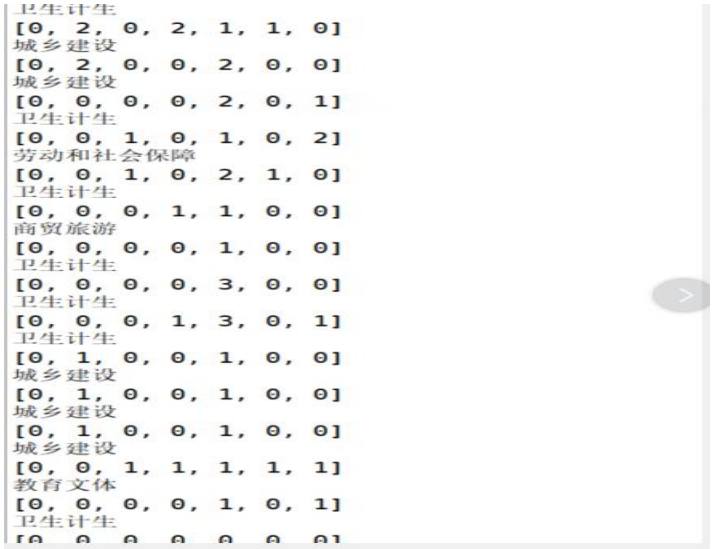
$$V(d)=(t_1,w_1(d),t_2,w_2(d),...,t_n,w_n(d))$$

其中，$t_i(i=1,2,3.....n)$ 为文档 d 的一个特征项，$w_i(i=1,2.....n)$ 为特征值的权值。

2.1.3.1 文本的向量化表示

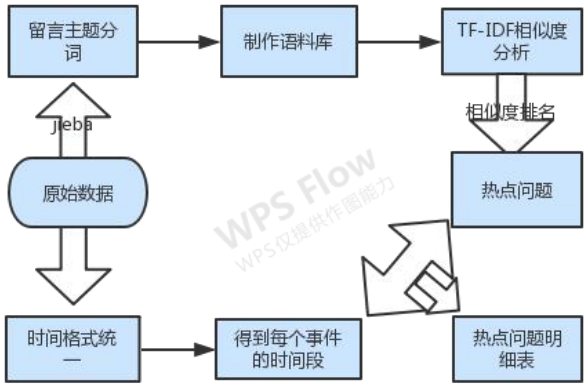
根据留言详情的特征项对应词袋的位置，组成统一维数的向量 $C=(t_1,t_2,.....,t_n)$，其中 C 为词袋集合，t_n 为每个词在向量对应的位置

下图为部分向量化结果：



2.2 问题 2 分析方法与过程

2.2.1 问题 2 流程图



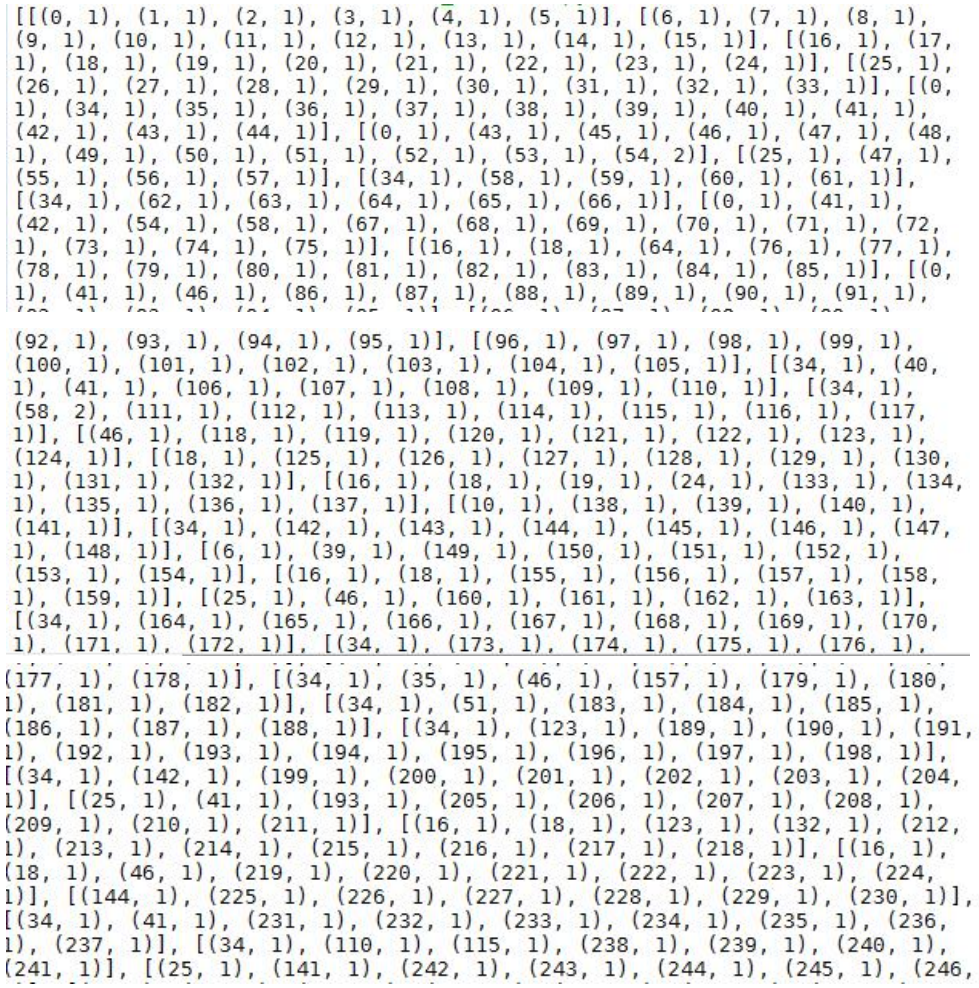
2.2.2 对留言主题分词

在对留言内容进行挖掘分析之前，先要把非结构化的文本信息转换为计算机能够识别的结构化信息。在附件 3 给出的数据是由中文文本的形式给出，为了便于转换，先要对这些留言详情进行中文分词。这里采用 python 的中文分词包 jieba 进行分词。jieba 采用了基于前缀词典实现的高效词图扫描，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),同时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，对于未登录词，采用了基于汉字成词能力的 HMM 模型，使得能更好的实现中文分词效果。

2.2.3 制作语料库

用 dictionary 方法获取词袋（bag-of-words），在词袋中用数字对所有词进行了编号，一个编号对应一个词。使用 doc2bow 制作语料库，语料库是一组向量，向量中的元素是一个二元组（编号、频次数），对应分词后的文档中的每一个词。用同样的方法，把测试文档也转换为二元组的向量。

下图为部分语料库：



2.2.4 文本相似度分析

这里利用 TF-IDF 进行文本相似度分析。 获取文档中，每个词的 TF-IDF 值，对每个项目，分析测试文档的相似度，根据相似度排序。

2.2.5TF-IDF 算法分析

TF-IDF 是一种用于信息检索的常用加权技术。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 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)， 以防止它偏向长的文件。公式：$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}$ 即：

$TF_{\omega}=\frac{\text{在某一类中词条}\omega\text{出现的次数}}{\text{该类中所有词条的数目}}$其中 ni,j 是该词在文件 dj 中出现的次数，分母则是文件 dj 中所有词汇出现的次数总和。[5]

IDF 表示逆向文件频率，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。

公式：$idf_i=\log\frac{|D|}{|\{j:t_i\in d_j\}|}$，其中，|D| 是语料库中的文件总数。 |{j:ti∈dj}| 表示包含词语 ti 的文件数目（即 ni,j≠0 的文件数目）。如果该词语不在语料库中，就会导致分母为零，因此一般情况下使用 1+|{j:ti∈dj}

所以，$IDF=\log(\frac{\text{语料库中的文档总数}}{\text{包含词条}\omega\text{的文档数+1}})$，分母要加 1，是防止分母为 0。

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。

公式：TF-IDF=TF*IDF

2.2.6 建立热点模型

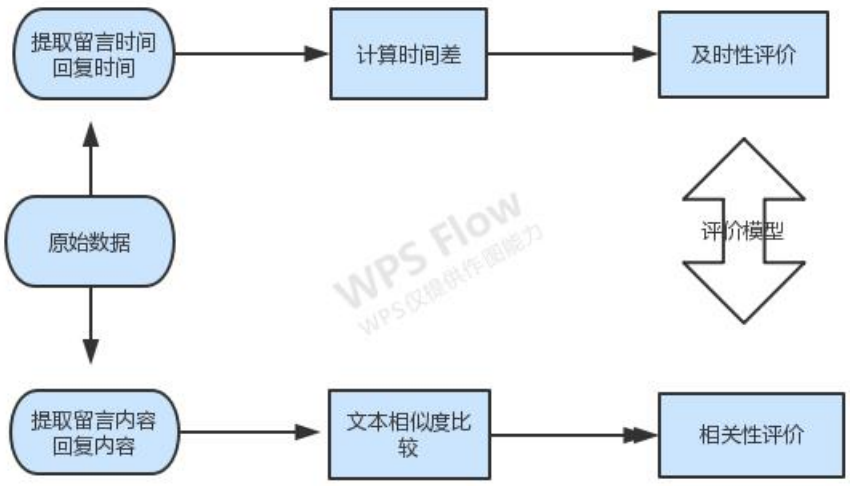
当一个词语频繁出现，那这个词语很有可能就与热点问题有关，基于这个考虑，计算词频，绘制词云如下图所示：

通过词云图显示，出现最多的为 A 市，A7 县等多为地名，因此利用词频方式效果不好，需进一步建立热点模型

由 TF-IDF 模型得到相似度排序，计算每条相似度大于 0.3 的数目将其与该留言放入一个数组里，遍历循环附件 3，得到每条留言以及与他相似的数目和该留言的反对数和点赞数，计算权重，将相似的数目的百分之八十五加上点赞与反对数的总和的百分之十五，作为最后的热点度，按照热点度的排行从大到小生成 xlsx 表格。

2.3 问题 3 分析过程与方法

2.3.1 问题 3 流程图



2.3.2 提取时间

回复的是否及时有时也是评判回复的标准，利用 python 读取留言时间以及回复时间，统一时间的格式为年/月/日 时：分：秒，存入数组，将元组前 4 个元素形成字符串。使用正则查询。

2.3.3 及时性模型

计算回复与留言的时间差，自定义以 10 天为一个范围，例如当时间差为 10

天之内时，表示及时性为 0.95，当时间差为 20 天之内时，表示及时性为 0.85，当时间差为 30 天之内时，表示及时性为 0.75，当时间差为 40 天之内，表示及时性为 0.65，当时间差为 50 天之内，表示及时性为 0.6，其余表示及时性为 0.3. 例如下表为部分留言的及时性评分：

```
[0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6,
0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.65, 0.6, 0.6, 0.75,
0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.3, 0.65,
0.6, 0.85, 0.75, 0.65, 0.6, 0.95, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75,
0.65, 0.6, 0.3, 0.95, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6,
0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6,
0.85, 0.75, 0.65, 0.6, 0.75, 0.65, 0.6, 0.65, 0.6, 0.85, 0.75, 0.65,
0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.75, 0.65, 0.6,
0.85, 0.75, 0.65, 0.6, 0.3, 0.75, 0.65, 0.6, 0.3, 0.85, 0.75, 0.65,
0.6, 0.75, 0.65, 0.6, 0.75, 0.65, 0.6, 0.75, 0.65, 0.6, 0.85, 0.75,
0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.65, 0.6, 0.65, 0.6, 0.85, 0.75,
0.65, 0.6, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.75, 0.65, 0.6, 0.75,
0.65, 0.6, 0.95, 0.85, 0.75, 0.65, 0.6, 0.75, 0.65, 0.6, 0.3, 0.3,

0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.75, 0.65,
0.6, 0.85, 0.75, 0.65, 0.6, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6,
0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85,
0.75, 0.65, 0.6, 0.95, 0.85, 0.75, 0.65, 0.6, 0.65, 0.6, 0.85, 0.75,
0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75,
0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75,
0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.95, 0.85,
0.75, 0.65, 0.6, 0.95, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6,
0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75,
0.65, 0.6, 0.95, 0.85, 0.75, 0.65, 0.6, 0.95, 0.85, 0.75, 0.65, 0.6,
0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6,
0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.65, 0.6, 0.75, 0.65,
0.6, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85, 0.75, 0.65, 0.6, 0.85,
```

2.3.4 相关性模型

利用 python 读取附件 4，提取留言详情以及回复详情，利用 jieba 中文分词对留言内容以及回复内容进行分词，如下图所示：

```
'打人', '了', '"', '。', '随行', '的', '执勤', '保安', '发现', '该名', '男
性', '身上', '酒味', '较', '重', ',', '对', '其', '乱喊', '之', '事', '深
感', '不妥', ',', '便', '对', '其', '也', '拍照', '取证', '。', '对方', '随
即', '拨打', '110', '报警', ',', '基于', '双方', '未', '发生', '肢体冲突',
',', '民警', '到达', '现场', '予以', '登记', '后', '便', '劝导', '各自', '离
开', '。', '整个', '过程', '中', ',', 'A2区', '城管', '执法', '大队', '坡
子', '街', '中队', '执法', '队员', '一直', '没有', '采取', '过激', '行为',
',', '也', '未', '对', '该', '名', '男性', '进行', '任何', '阻扰', '和',
'肢体冲突', '等', '行为', ',', '并', '成功', '劝离', '了', '流动', '摊贩',
'。', '感谢您', '对', '我们', '工作', '的', '支持', '、', '理解', '与', '监
督', '！', '2018', '年', '12', '月', '21', '日']]
```

```
'诸多', '问题', '的', '投诉'], ['投诉', 'K9', '县', '二中', '强制', '补考',
'并', '收取', '补课费'], ['投诉', 'K9', '县', '二中', '重点班', '强制', '补
课'], ['举报', '江', '花县', '教育局', '营养餐', '采购', '项目', '违法', '制
作', '招标', '文件'], ['投诉', 'K6', '县绍基', '学校', '周末', '有偿', '补
课'], ['投诉', 'K3', '县', '二中', '强制', '高三', '学生', '补课'], ['投诉',
'K9', '县', '二中', '变相', '要求', '学生', '补课'], ['投诉', 'K', '市', '明
德', '楚南', '学校', '乱收费'], ['对', 'K6', '县', '小升初', '政策', '的',
'质疑'], ['对', 'K7', '县', '公费', '定向生', '录取', '结果', '表示', '质
疑'], ['呼吁', '政府部门', '及', '教育局', '为', 'K1', '区', '孩子', '们',
'多建', '学校'], ['请求', '放宽', '入学', '界限', '！']]
```

利用留言详情与回复详情建立语料库，如下图所示:

```
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1)], []]
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8,
1), (9, 1), (10, 1)], []]
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1)], []]
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1)], []]
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1)], []]
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1)], []]
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8,
1)], []]
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1)], []]
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8,
1)], []]
```

利用 IF-IDF 进行相似度比较，得到对应的数值，如下图所示：

```
[0.9622505, 0.9486833, 0.7537783, 0.6904309, 0.7745967, 0.95618284,
0.82823646, 0.650814, 0.62994075, 0.5604485, 0.70710677, 0.8140999,
0.7071067, 0.0, 0.70710677, 0.7493075, 0.37796447, 0.7302968,
0.43519413, 0.35355338, 0.66943866, 0.53452253, 0.35355338, 0.4447496,
0.65465367, 0.7106691, 0.6634888, 0.57735026, 0.75592893, 0.71330416,
0.40824828, 0.6, 0.0, 0.606977, 0.31622776, 0.57735026, 0.66943866,
0.6681531, 0.42640144, 0.7893522, 0.4082483, 0.6666667, 0.59999996,
0.8838923, 0.5477226, 0.6324555, 0.8164966, 0.71713716, 0.79471934,
0.86154974, 0.8087458, 0.6459422, 0.30151135, 0.45834926, 0.60385966,
0.62994075, 0.65997505, 0.6446584, 0.808122, 0.5111657, 0.57735026,
0.7161149, 0.43495885, 0.5500914, 0.65465367, 0.0, 0.0, 0.80178374,
0.59999996, 0.35355338, 0.5, 0.0, 0.72547626, 0.69985425, 0.8333334,
0.6761234, 0.31622776, 0.7745967, 0.75384617, 0.88741195, 0.5659165,
```

```
0.66226614, 0.630126, 0.33333334, 0.57735026, 0.4472136, 0.25,
0.54494923, 0.57735026, 0.63997465, 0.4082483, 0.30151135, 0.35355338,
0.35355338, 0.7337994, 0.4082483, 0.5163978, 0.70710677, 0.37796447,
0.75793684, 0.7947194, 0.73333335, 0.57735026, 0.57735026, 0.5656854,
0.6324555, 0.48038444, 0.6172134, 0.64820373, 0.6324555, 0.65465367,
0.7167132, 0.30151135, 0.8944272, 0.5, 0.86193424, 0.88388336,
0.35355338, 0.5345225, 0.6324555, 0.70710677, 0.8944272, 0.49999997,
0.700649, 0.6459752, 0.51425946, 0.7119914, 0.6324555, 0.6847368,
0.5163978, 0.0, 0.0, 0.7715168, 0.86970985, 0.5378679, 0.6172134,
0.6363636, 0.5163978, 0.6823231, 0.70710677, 0.4767313, 0.848528,
0.8340577, 0.40509576, 0.49999997, 1.0, 0.5556624, 0.5500914,
0.5163978, 0.6981871, 0.35355338, 0.7745967, 0.49999997, 0.49999997,
0.6324555, 0.81649655, 0.5345225, 0.35355338, 0.4685213, 0.70710677,
```

3. 结果分析

3.1 问题 1 结果分析

3.1.1 一级标签分析

利用正则表达式对 jieba 分词后的留言详情与录入的一级分类的相关词汇进行匹配，将所含数目最多的关键词定义为该留言详情的一级标签。得到如“一级标签 txt”的一级标签，再将一级标签 txt 与读取并保存的一级分类 txt 进行对比，计算每一类的查准率以及查全率。如下表所示：

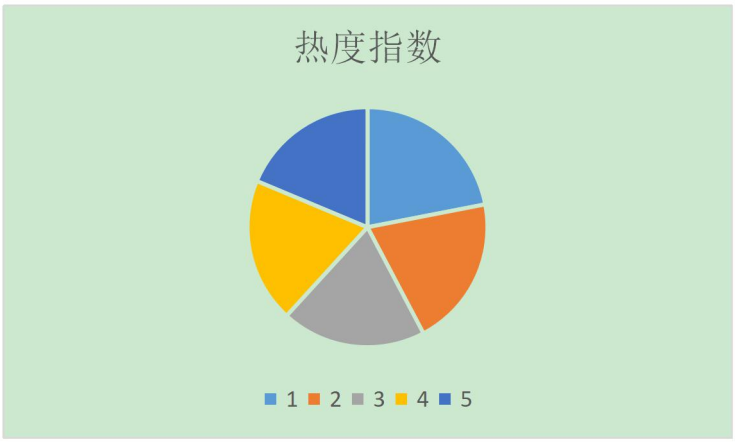
一级标签	查全率	查准率
城乡建设	100%	71.34%
环境保护	100%	84.77%
交通运输	100%	85.99%
教育文体	100%	86.54%
劳动与社会保障	100%	58.27%
商贸旅游	100%	42.76%
卫生计生	100%	76.99%

通过 F-Score 公式 $\frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$ 计算得 0.8297，其中 P_i 为第 i 类的查准率，R_i 为第 i 类的查全率。

3.2 问题 2 结果分析

通过 TF-IDF 对数据的相似度分析以及排序，得到排名前 5 的热点问题如下图所示：

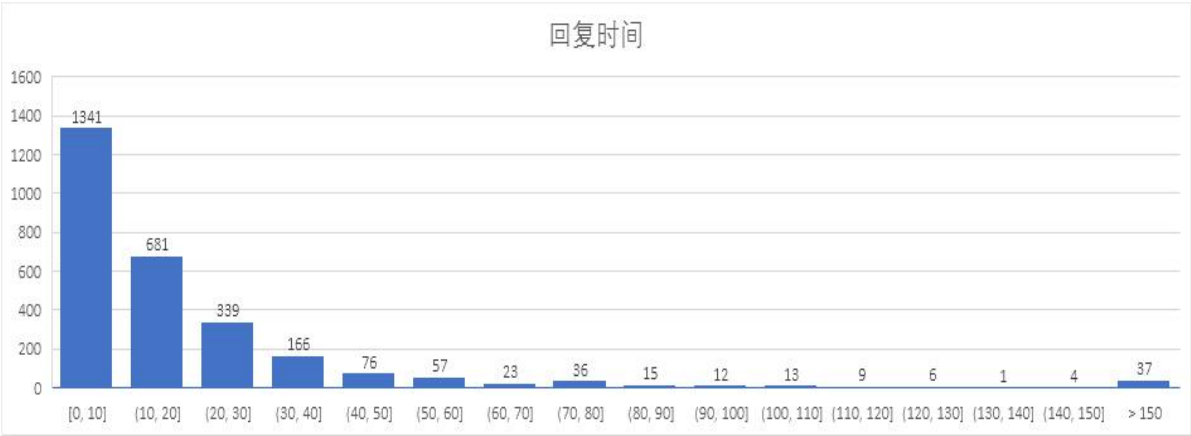
热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述
1	1	54	2019/07/03至2020/01/26	新城小区新城	丽发新城小区附近搅拌站的一些问题
2	2	50	2019/07/07至2019/09/01	伊滨河伊滨河个人	伊景园滨河苑车位捆绑销售！广铁集团做个人吧！
3	3	48	2019/01/02至2020/01/06	A市	请A市加快地铁的建设
4	4	48	2018/11/15至2019/12/24	A市人才	咨询A市人才购房补贴政策
5	5	46	2019/02/27至2019/12/25	A市	询问A市住房公积金贷款的相关问题



由表格可以得到群众反映的热点问题为丽发新城小区附近搅拌站扰民占比22%，伊景园滨河苑车位捆绑销售占比 20%，A 市地铁问题占比 20%，A 市购房补贴问题占比 20%，A 市住房公积金占比 19%，且每个热点问题所占的比例近似相等，由此可以得到百姓所反映的问题围绕生活，出行，住房。

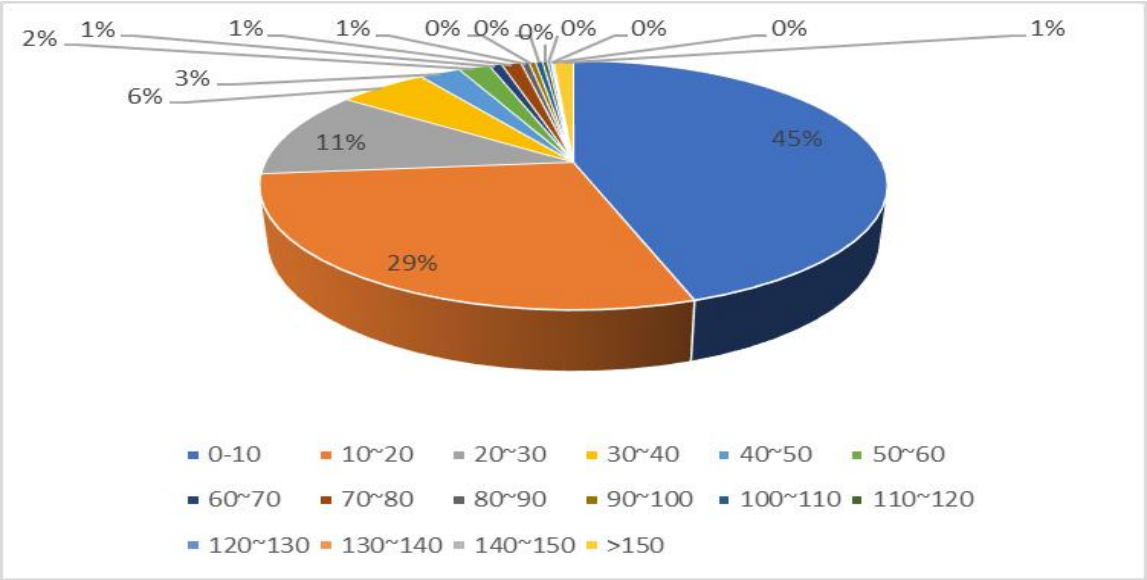
3.3 问题 3 结果分析

通过对每条留言的及时性分析，我们绘制了频数直方图：



整理成表格得到
绘制饼图得到

0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100	100-110	110-120	120-130	130-140	140-150	>150
1341	861	339	166	76	57	23	36	15	12	13	9	6	1	4	37



由以上数据可以得到，回复时间在 10 天之内的占比最多达到百分之四十五，随着时间的增加，数目也在逐渐减少，而当回复时间超过 140 天时，数目开始增多

通过综合考虑相关性以及及时性得到

```
该模型的得分为(100分制度)：  64.02172372189604
该模型评价等级为：  合格
```

也就是说市长的回复较为及时，与留言内容具有一定相关性

4、结论

对于留言信息进行分析研究，对提升政府管理水平和施政效率具有极大的推动作用。本文采用 jieba 中文分词以及 TF-IDF 对数据进行研究。统计目前市民反映最多的问题，同时定义及时性以及相关性评分标准，对市长的回复进行评价。

由分析结果可得，目前市民最关注的问题为丽发新城小区附近搅拌站扰民，伊景园滨河苑车位捆绑销售，A 市地铁问题，A 市购房补贴问题，A 市住房公积金，围绕着市民的起居生活，交通出行。

市长回复消息的时间在 10 天之内居多，占比百分之四十五，但是也有超过 150 的天，占比百分之一，因此可以说市长回复消息较为及时。根据自定义的分数，综合考察及时性以及相关性，最后得分 64 分。因此，市长的回复与留言内容具有一定相关性。也就是说，市民反映的问题基本得到回复解决。

参考文献：

[1] 陶伟. 警务应用中基于双向最大匹配法的中文分词算法实现[J]. 电子技术与软件工程，2016(4).

[2] [2]详解正则表达式匹配算法原理_网络_ybdesire 的专栏-CSDN 博客 https://blog.csdn.net/ybdesire/article/details/78255427

[3]周昭涛.文本聚类分析效果评价及文本表示研究[D].中国科学院研究生院(计算技术研究所)，2005.

[4] 胡学钢，董学春，谢飞.基于词向量空间模型的中文文本分类方法[J].合肥工业大学学报:自然科学版，2007， 30(10) :1261-1264.

[5] TF-IDF 算法介绍及实现_人工智能_Asia-Lee 的博客-CSDN 博客 https://blog.csdn.net/asialee_bird/article/details/81486700