

# 基于自然语言处理技术的智慧政务系统

## 摘要:

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。

文本挖掘的主要用途是从原本未经处理的文本中提取出未知的知识，其主要处理那些模糊而且非结构化的文本数据。文本挖掘的主要支撑技术：自然语言处理技术和机器学习。文本挖掘一般要经过数据清洗、分词和词性处理、去停用词、文本特征选择、文本表示、计算文本相似度，构建模型和模型验证调优等几个阶段，这几个阶段都会用到各种数学模型。

本文中主要通过 jieba 中文分词库来对文本进行预处理，去除数据中无用的特殊字符和对样本进行分词操作以及去停用词等操作；得到清洗后数据再使用 TF-IDF 方法提取文本特征，利用 GridSearchCV 寻找最优参数，构建模型后使用 F-Score 对分类方法进行评价；为了找出文本之间的联系，采用 gensim 库进行文本相似度分析，HanLp 的自动摘要和命名实体识别；应用基于 Stack Overflow 修改的的排名算法来实现热度指数。

**关键字：** jieba 中文分词，TF-IDF 方法，GridSearchCV，gensim 库，HanLp

# 目录

1、绪论.....	1
1.1 背景.....	1
1.2 挖掘目标.....	1
2、问题分析.....	2
2.1 问题一的分析.....	2
2.2 问题二的分析.....	2
2.3 问题三的分析.....	3
3、群众留言分类.....	3
3.1 进行数据预处理.....	3
3.2 对留言进行分词处理.....	3
3.3 文本特征提取 TF-IDF.....	4
3.4 选择分类器算法.....	5
3.5 利用 GridSearchCV 寻找最优参数.....	6
3.5 得到 GridSearchCV 训练出的最优模型并预测.....	6
3.6 使用 F1-Score 对分类模型进行评价.....	6
3.7 使用 joblib 持久化模型.....	7
4、热点问题挖掘.....	7
4.1 进行数据预处理.....	7
4.2 采用 gensim 库进行文本相似度分析.....	8
4.3 相似问题的归类.....	10
4.4 标记分组.....	11
4.5 时间范围：转化时间格式.....	11
4.6 问题描述：HanLp 的自动摘要.....	11
4.7 地点/人群：HanLp 命名实体识别.....	11
4.8 热度指数.....	11
(1) Qsum (同类问题的总数) .....	11
(2) Qscore (问题得分) .....	12
4.9 保存到本地.....	12
5、留言质量评价.....	12
5.1 评价指标.....	12
6、参考文献.....	13

# 1、绪论

## 1.1 背景

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,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,智慧政务随之产生。

智慧政务是相关部门和地方各级政府利用信息和网络通信技术,加强政府管理,实现政务公开化透明化以此来提高处理效率、进行科学决策、改进和完善服务职能的重要手段,是一项系统工程。

自然语言处理,即实现人机间自然语言通信,或实现自然语言理解和自然语言生成是十分困难的。造成困难的根本原因是自然语言文本和对话的各个层次上广泛存在的各种各样的歧义性或多义性。

文本挖掘大致可由三部分组成:底层是文本数据挖掘的基础领域,包括机器学习、数理统计、自然语言处理;在此基础上是文本数据挖掘的基本技术,有五大类,包括文本信息抽取、文本分类、文本聚类、文本数据压缩、文本数据处理;在基本技术之上是两个主要应用领域,包括信息访问和知识发现,信息访问包括信息检索、信息浏览、信息过滤、信息报告,知识发现包括数据分析、数据预测。

## 1.2 挖掘目标

### 问题一:群众留言分类

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(参考附件 1 提供的内容分类三级标签体系)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。

通常使用 F-Score 对分类方法进行评价:

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

其中  $P_i$  为第  $i$  类的查准率,  $R_i$  为第  $i$  类的查全率。

图 1-1 F-Score 评价指标

## 问题二：热点问题挖掘

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，并保存为文件“热点问题表.xls”。按表 2 的格式给出相应热点问题对应的留言信息，并保存为“热点问题留言明细表.xls”。

## 问题三：答复意见的评价

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。

# 2、问题分析

## 2.1 问题一的分析

要解决的主要问题有文本语义带来的词语交叉、多分类问题的处理难度、数据量的增多带来的影响以及长文本的无意义表达太多等，求解步骤如下：

- (1) 导入附件 1 和附件 2 中的数据。
- (2) 去除无用标签，只保留需要分析的留言详情和一级标签。
- (3) 使用 jieba 对留言详情进行分词。
- (4) 分割训练集和测试集。
- (5) 进行文本特征提取（TF-IDF）。
- (6) 创建文本分类模型。
- (7) 利用得到的模型进行预测。
- (8) 使用 F1-score 对预测进行评分。
- (9) 保存模型。

## 2.2 问题二的分析

要解决的主要问题是，如何从众多留言中识别出相似的留言，如何把特定地点或人群的数据归并以及设计热度评价指标和计算方法，求解步骤如下：

- (1) 导入附件 3 中的数据。
- (2) 对文本进行数据预处理，去除留言详情无用的特殊字符。
- (3) 使用 jieba 进行中文分词。
- (4) 制作语料库。
- (5) 使用 TF-IDF 模型对语料库建模，稀疏矩阵相似度，从而建立索引
- (6) 计算所有留言数据两两之间的相似度，进行数据分类。
- (7) 转化时间格式。
- (8) HanLp 的自动摘要和 HanLp 命名实体识别。
- (9) 计算热度指数。
- (10) 保存到本地。

### 2.3 问题三的分析

要解决的主要问题是针对相关部门对留言的答复意见，从答复的相关性完整性和可解释性等角度对答复意见给出一套可行的评价方案。

## 3、群众留言分类

### 3.1 进行数据预处理

去除无用标签，只保留[留言详情]和[一级标签]，如图 3-1 所示。

```
In [138]: data.drop(labels=['留言编号','留言用户','留言主题','留言时间'],axis=1,inplace=True)
```

图 3-1 去除无用标签

### 3.2 对留言进行分词处理

jieba 中文分词的特点：

- (1) 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）。
- (2) 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。
- (3) 对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法。

具体步骤：

- (1) 添加用户自定义字典。

- (2) 加载用词列表。
- (3) 去停用词。
- (4) 对留言详情进行分词。

分词详情：

Out[142]:

	留言详情	一级标签
0	A3 区大道 西行 便道 未管 路口 加油站 路段 人行道 包括 路灯 杆 圈 西湖 建...	城乡建设
1	位于书院 路 主干道 在水一方 大厦 一楼 四楼 人为 拆除 水 电等 设施 烂尾 多年 ...	城乡建设
2	尊敬 领导 A1 区苑 小区 位于 A1 区 火炬 路 小区 物业 A市 程明 物业管理 有...	城乡建设
3	A1 区 A2 区华庭 小区 高层 二次 供水 楼顶 水箱 长年 不洗 现在 自来水 龙头 ...	城乡建设
4	A1 区 A2 区华庭 小区 高层 二次 供水 楼顶 水箱 长年 不洗 现在 自来水 龙头 ...	城乡建设
...	...	...
9205	夫妻 农村户口 女 9 岁 2 岁 半 15 斤 治疗 两年 一级 脑瘫 纯 女户 招郎 男...	卫生计生
9206	2015 年 2 月 16 号 B 市中心 医院 做 无痛 人流 手术 手术 怀孕 症状 2...	卫生计生
9207	再婚 想 一个 小孩 不知 我省 二胎 新 政策 出 先 怀孕 会 做 处理	卫生计生
9208	K8 县惊现 奇葩 证明 西地省 K8 县人 想 生 二胎 告知 要开 证明 没生...	卫生计生
9209	领导 你好 属于 未 婚生子 2013 年 已经 接受 处罚 小孩 上户 小孩 外地 上学 ...	卫生计生

9210 rows × 2 columns

图 3-2 jieba 分词结果

### 3.3 文本特征提取 TF-IDF

定义：TF-IDF 即词频-逆文本频率，由 TF（词频）和 IDF（逆文本频率）两部分组成，是一种加权技术，其采用统计方法根据字词在文本中出现的次数和在整语料中出现的文档频率来计算一个词在整个语料中的重要程度。

TF-IDF 计算公式：  $IDF = \log\left(\frac{\text{语料库的文档总数}}{\text{包含词条 } x \text{ 的文档数} + 1}\right)$

使用 Sklearn 的 TfidfVectorizer 实现数据的 TF-IDF 特征提取：

- (1) 创建 TfidfVectorizer 模型，步长设置为 4，以保证原语句的顺序语义不被破坏。
- (2) 使用全部数据来训练得到一个特征模型。

```
TfidfVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.float64'>, encoding='utf-8',
                input='content', lowercase=True, max_df=0.9, max_features=5500,
                min_df=5, ngram_range=(1, 4), norm='l2', preprocessor=None,
                smooth_idf=True, stop_words=None, strip_accents=None,
                sublinear_tf=False, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, use_idf=True, vocabulary=None)
```

图 3-3 TfidfVectorizer 模型各项参数

(3) 划分训练集和测试集，并混淆顺序。

(4) 分别对训练集文本和测试集文本进行特征工程，得到它们的稀疏矩阵。

### 特征工程

```
X_train = TF_feature.transform(X_train)
X_test = TF_feature.transform(X_test)
```

图 3-4 特征工程

### TF-IDF 算法评价

优点：简单快捷，结果比较符合实际情况。

缺点：（1）没有考虑特征词的位置因素对文本的区分度，词条出现在文档的不同位置时，对区分度的贡献大小是不一样的。

（2）按照传统 TF-IDF，往往一些生僻词的 IDF(反文档频率)会比较高、因此这些生僻词常会被误认为是文档关键词。

（3）传统 TF-IDF 中的 IDF 部分只考虑了特征词与它出现的文本数之间的关系，而忽略了特征项在一个类别中不同的类别间的分布情况。

（4）对于文档中出现次数较少的重要人名、地名信息提取效果不佳。

### 3.4 选择分类器算法

针对本小问，对比多种分类算法的效率和准确率最终选择使用线性逻辑回归算法。

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

图 3-5 线性逻辑回归算法

### 3.5 利用 GridSearchCV 寻找最优参数

构建参数列表，选择逻辑回归模型参数 C, dual, solver 进行网格搜索。

```
param_grid = {'C':range(1,10),
              'dual':[True,False],
              'solver':['liblinear']}
```

图 3-6

GridSearchCV 的优点：自动调参，只要把参数输进去，就能给出最优化的结果和参数，用于系统地遍历多种参数组合，通过交叉验证确定最佳效果参数。

GridSearchCV 的缺点：只适用于小数据集，一旦数据的量级大了，将难以得出结果。

```
Out[245]: GridSearchCV(cv=5, error_score=nan,
                      estimator=LogisticRegression(C=1.0, class_weight=None, dual=False,
                                                    fit_intercept=True,
                                                    intercept_scaling=1, l1_ratio=None,
                                                    max_iter=100, multi_class='auto',
                                                    n_jobs=None, penalty='l2',
                                                    random_state=None, solver='lbfgs',
                                                    tol=0.0001, verbose=0,
                                                    warm_start=False),
                      iid='deprecated', n_jobs=-1,
                      param_grid={'C': range(1, 10), 'dual': [True, False]},
                      pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
                      scoring=None, verbose=0)
```

图 3-7

### 3.5 得到 GridSearchCV 训练出的最优模型并预测

```
model = grid.best_estimator_
```

### 3.6 使用 F1-Score 对分类模型进行评价

Precision 体现了模型对负样本的区分能力，Precision 越高，模型对负样本的区分能力越强；Recall 体现了模型对正样本的识别能力，Recall 越高，模型对正样本的识别能力越强。F1 score 是两者的综合，F1 score 分数越高，说明模型越稳健。

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

图 3-8 F1-score 评价指标

使用 sklearn 实现评估函数：



```
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score

def print_evaluation_scores(Y_test, predicted):
    accuracy = accuracy_score(Y_test, predicted)
    macro = f1_score(Y_test, predicted, average='macro')
    micro = f1_score(Y_test, predicted, average='micro')
    weighted = f1_score(Y_test, predicted, average='weighted')
    print("准确率 (accuracy) :", accuracy)
    print("macro:", macro)
    print("micro:", micro)
    print("weighted:", weighted)|
```

图 3-9 使用 F1-score 评价指标的评估函数

### 3.7 使用 joblib 持久化模型

```
joblib.dump(grid.best_estimator_, './一级标签.model')
```

图 3-10 使用 joblib 保存模型

## 4、热点问题挖掘

热点问题主要思路如图 4-1 所示：

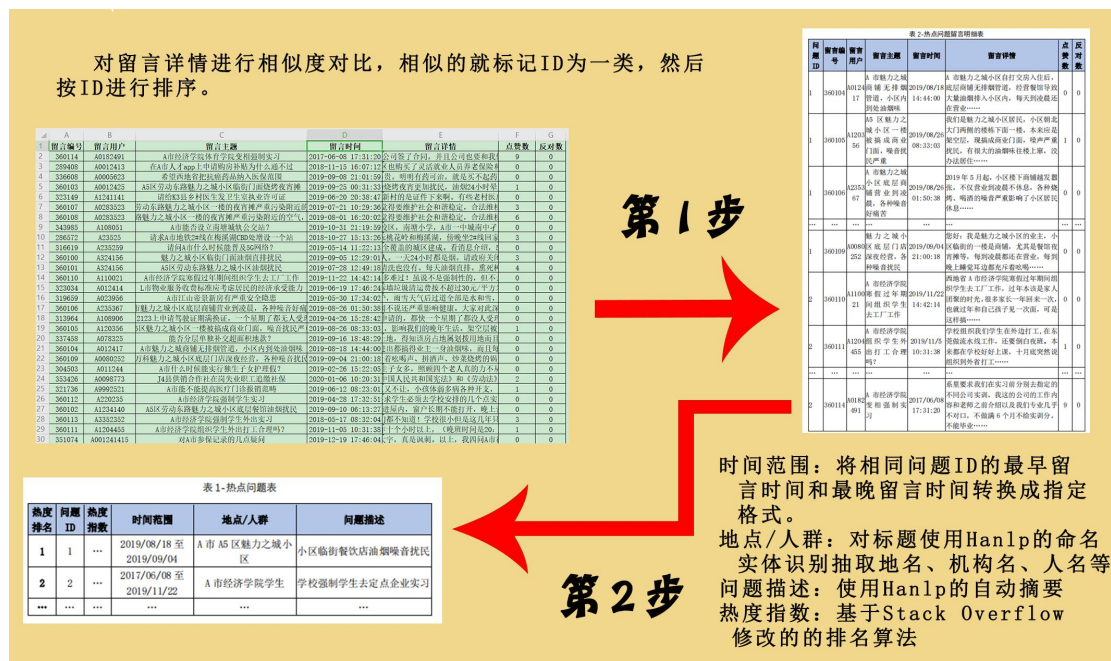


图 4-1 热点问题挖掘思路图

### 4.1 进行数据预处理

首先需要观察数据，由下图可知，读取进来的数据，由于编码缘故，留言详情两端有一些特殊字符，这些特殊字符会影响我们的分词效果，因此需要去除掉。

```
In [3]: excel_name = 'D:/编程资料以及作业/数据挖掘/第八届泰迪杯c题/C题全部数据/附件3.xlsx'
df = pd.read_excel(excel_name, encoding='utf8')
df.head(2)
```

Out[3]:

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	
0	188006	A000102948	A3区一米阳光婚纱摄影是否合法纳税了?	2019/2/28 11:25:05	该店座落在A市A3区联丰路米兰春天G2栋320, ...	0	0
1	188007	A00074795	咨询A6区道路命名规划初步成果公示和城乡门牌问题	2019/2/14 20:00:00	A市A6区道路命名规划已经初步成果公示文件, ...	0	1

图 4-2 观察留言详情两端的特殊字符

除此之外，在进行相似度计算中，为了能够提高索引的速度和节省存储空间，以及提升相似度计算的准确性，需要过滤掉一些自身无明确意义的词，包括语气助词、副词、介词、连接词等，如常见的“的”、“在”之类。

我们抽取“留言主题”和“留言详情”作为相似度计算的样本，接下来加载停用词和自定义字典，使用了 jieba 库对样本进行分词以及去停用词操作。

#### 4.2 采用 gensim 库进行文本相似度分析

由于计算机不能直接识别中文字符，因此需要考虑词如何在计算机中表示，即需要将文本转化成词向量，主要使用的算法是 TF-IDF。

数据两两之间的相似度

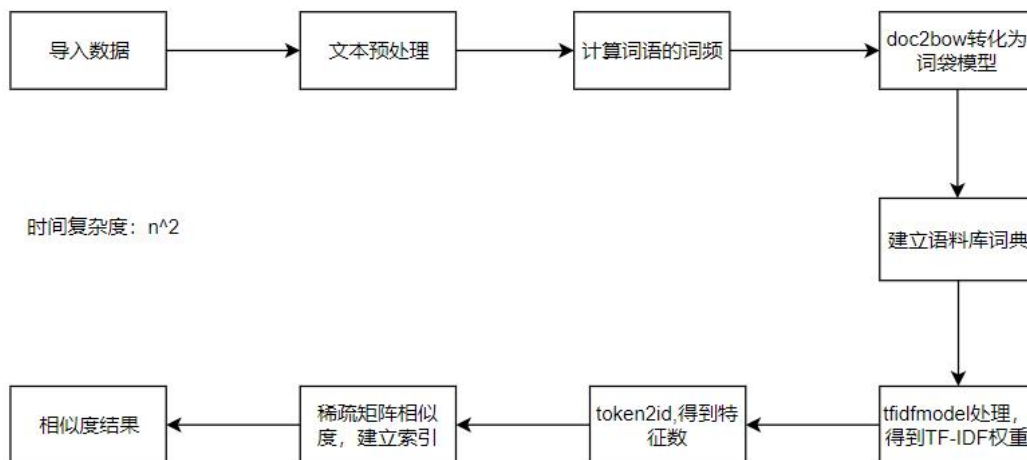
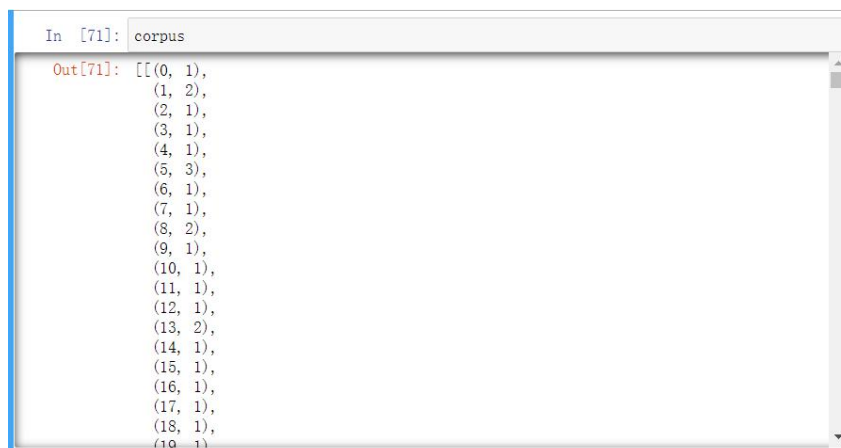


图 4-3 留言文本相似度计算的流程图

留言文本相似度计算的步骤如下，具体流程图如图 4-2 所示：

- (1)：因此需要把分词后的留言文本列表，使用 gensim 的 `corpora.dictionary` 方法获取词袋（bag of words）。
- (2)：使用词袋的 `doc2bow` 方法依次遍历留言文本列表，建立语料库。



```
In [71]: corpus

Out[71]: [(0, 1),
          (1, 2),
          (2, 1),
          (3, 1),
          (4, 1),
          (5, 3),
          (6, 1),
          (7, 1),
          (8, 2),
          (9, 1),
          (10, 1),
          (11, 1),
          (12, 1),
          (13, 2),
          (14, 1),
          (15, 1),
          (16, 1),
          (17, 1),
          (18, 1),
          (19, 1)]
```

图 4-4 生成的语料库

- (3)：将语料库通过 `tfidfmodel` 进行处理，得到 TF-IDF 权重。

词频（term frequency, TF）

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

逆向文件频率（inverse document frequency, IDF）

$$idf_i = \lg \frac{|D|}{|\{j : t_i \in d_j\}|}$$

主要思想是：如果某个词或短语在一篇文章中出现的频率高，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。

统计第一个留言样本的词数  $N$ ，计算第一个留言样本第一个词在该样本中出现的次数  $n$ ，再找出该词在所有样本中出现的次数  $m$ 。

则该词的  $tf-idf$  为： $n/N * 1/(m/M)$ 。

- (4)：重复第(3)步，计算出一个留言样本所有词的  $tf-idf$  值。
- (5)：重复第(4)步，计算出所有留言样本每个词的  $tf-idf$  值。

(6)：稀疏矩阵相似度，从而建立索引 index，获取每个留言样本与所有留言的相似度列表。

### 4.3 相似问题的归类

- (1) 创建一个空集合 data\_set，用于存放已加入字典中的序号。
- (2) 创建一个空字典 data\_dic，用于存放 ID（标志 flag）号（key）和相似留言序号列表（values）。
- (3) 遍历第一个留言样本。
- (4) 获取该留言与全部留言的相似度列表 sim。
- (5) 求出相似度列表 sim 中满足预先设定的指标的留言序号，并存放到集合 data\_list 中。
- (6) 判断上一步保存的列表是否为空？如果为空则进入下一个循环，跳转到第（9）步。
- (7) 判断：如果集合 data\_list 与总集合有交集，即总集合中已存在该留言序号，便遍历字典 data\_dic 中每一个键值对，判断集合 data\_list 是否与字典 data\_dic 中的元素存在交集，有交集就添加进字典 data\_dic 当前元素中；无交集则添加新的元素（键值对）进字典 data\_dic。
- (8) 把集合 data\_list 的值更新到集合 data\_set 中。
- (9) 续遍历下一个留言样本，即跳转到到第（4）步，直到遍历完所有留言样本。
- (10) 得到按照相似问题分类的字典 data\_dic。

```
相似文档的序号为：
Out[55]: {1: {0},
          2: {1},
          3: {2},
          4: {3},
          5: {4, 61, 707, 1120, 1340, 1902, 2031, 2207, 4248, 4289},
          6: {5},
          7: {6, 326, 748},
          8: {7},
          9: {8},
          10: {9, 917, 960},
          11: {10, 2303, 2721, 2928, 4163},
          12: {11},
          13: {12, 434, 681, 3618, 4013},
          14: {13},
          15: {14},
          16: {15, 1136},
          17: {16, 4304},
          18: {17},
```

图 4-5 得到相似问题的分类字典

#### 4.4 标记分组

使用相似问题的分类字典更新一列“问题 ID”到附件三表

问题ID	留言编号	留言用户	留言主题	留言时间	留言详情	点赞数	反对数	
0	1	188006	A000102948	A3区一米阳光婚纱摄影是否合法纳税了?	2019/2/28 11:25:05	座落在A市A3区联丰路米兰春天G2栋320，一家名叫一米阳光婚纱摄影的影楼，据说年单这一...	0	0
1	2	188007	A00074795	咨询A6区道路命名规划初步成果公示和城乡门牌问题	2019/2/14 20:00:00	A市A6区道路命名规划已经初步成果公示文件，什么时候能转化成正式的成果，希望能加快完成的路...	1	0
2	3	188031	A00040066	反映A7县春华镇金鼎村水泥路、自来水到户的问题	2019/7/19 18:19:54	本人系春华镇金鼎村七里组村民，不知是否有相关水泥路到户政策和自来水到户政策，如政府主导投资村...	1	0
3	4	188039	A00081379	A2区黄兴路步行街大古道巷住户卫生间粪便外排	2019/8/19 11:48:23	靠近黄兴路步行街，城南路街道、大古道巷、一步两梯桥小区（停车场东面围墙外），第一单元一住户卫...	1	0
4	5	188059	A00028571	A市A3区中海国际社区三期与四期中间空地夜间施工噪音扰民	2019/11/22 16:54:42	A市A3区中海国际社区三期四期中间，即蓝天瑞和洲幼儿园旁边那块空地一直处于三不管状态，物业不...	0	0

表 4-6 完成问题分类后的附件三表

#### 4.5 时间范围：转化时间格式

对附件三表按“问题 ID”和“留言时间”进行升序操作，接着按“问题 ID”进行分组，每组按取“留言时间”的第一个和最后一个，转化成一致的时间格式。

#### 4.6 问题描述：HanLp 的自动摘要

对附件三按“问题 ID”进行分组，每组对组内的“留言主题”使用 HanLp 的自动摘要来提取摘要

```
def bbd(text):  
  
    document = ""  
    document = text['留言主题'].str.cat(sep='。')  
    return ''.join(HanLP.extractSummary(document, 1))
```

表 4-7 通过 HanLp 自动摘要实现的提取摘要函数

#### 4.7 地点/人群：HanLp 命名实体识别

对于地名/人群这一列，则要求提取该问题在哪个地点发生的问题或哪个群体的问题，这就需要用命名实体识别来提取相应词性的词，这里我们使用 Hanlp 的命名实体识别，内置分词器采用感知机算法，并开启地名识别功能。然后筛选出满足预先设定好的词性组合的词的列表。

#### 4.8 热度指数

基于 Stack Overflow 修改的排名算法

```
def HeatIndex(Qsum, Qscore):  
    return math.log10(Qsum*4) + float(Qscore) / 5
```

图 4-8 排名算法核心代码

(1) Qsum（同类问题的总数）

某类问题的数量越多，就代表越受关注，得分也就越高。这里使用了以 10 为底的对数，用意是当此类问题的数量越来越大，它对得分的影响将不断变小。

(2) Qscore (问题得分)

首先，Qscore (问题得分) = 赞成票-反对票。如果某个问题越受到好评，排名自然应该越靠前。

**总结:基于 Stack Overflow 算法修改的的排名,与参与度 Qsum 和质量 Qscore 成正比。**

#### 4.9 保存到本地

- (1)将处理好的数据插入表中,热度指数 Heat\_rusults,时间范围 liuyan\_date, 问题描述 problem。
- (2) 根据热度指数对每一个问题进行排名, 并将排名写入“热度排名”中; 生成排名列表 Heat\_ranking, 列表的(下标+1)是排名, 此位置的元素为: 该排名的问题 ID。
- (3) 按照热度排名进行排序。
- (4) 保存热点问题表。
- (5) 生成热点问题留言明细表并保存。

## 5、留言质量评价

### 5.1 评价指标

- (1) 留言和回复的相关性: 可用文本相似度来进行比较, 当两者相似度满足某个阈值则可判定留言与回复相关, 但若相似度过高则判定为照搬问题, 恶意灌水。
- (2) 留言的完整性: 则看回复是否满足一定的规范性, 可以使用正则表达式匹配。

规范性: 可以用正则表达式对答复的格式进行筛选, 比方说, 标准答复格式: 尊敬的网民、市民: 您好, 收到您反映的问题。现回复如下: …… (有些分点作答) 感谢您对我们工作的信任与支持! 祝您生活愉快! 2019 年 5 月 20 日。



## 6、参考文献

- [1] 结巴中文分词 <https://github.com/fxsjy/jieba>
- [2] HanLP 中文分词 <http://hanlp.com>
- [3] 模型选择 [https://blog.csdn.net/qq\\_41664845/article/details/80305101](https://blog.csdn.net/qq_41664845/article/details/80305101)
- [4] gensim 库《中文自然语言处理入门》 宿永杰
- [5] TfidfVectorizer <https://blog.csdn.net/laobai1015/article/details/80451371>
- [6] Stack Overflow <https://yq.aliyun.com/articles/54351>
- [7] GridSearchCV <https://www.biaodianfu.com/gridsearchcv.html>