面向“智慧政务”的文本挖掘

摘要

伴随网络问政普及率越来越高以及科学技术的快速发展，用户针对社会热点问题、政府行为的表达方式更加广泛、多样、便捷、高效，在各类社情民意数据量不断攀升的情况下，对数据进行智能化处理在智慧政务建设中扮演着越来越重要角色。利用人工智能等相关技术，建立基于自然语言处理技术的智慧政务系统是提升社会治理创新发展和高效履职的重要手段。

针对问题一，首先在数据预处理过程中提取群众留言内容后对其进行分词和去停用词处理，运用 word2vec 和百度百科中文预训练词向量将每条留言内容转换成向量，转成过程中利用特征加权方法改进向量的生成并建立了一个基于支持向量机的留言文本分类模型，模型选取 RBF 作为核函数并运用网格法获得最优惩罚参数和核参数，模型的 F-Score 为 0.88，为观察预训练词向量对模型的影响，利用维基百科预训练词向量重复实验获得 F-Score 为 0.89，因此预训练词向量的选择对分类有一定影响。

针对问题二，通过 TF-IDF 提取留言详情的特征词，构造词向量空间，同时运用 PCA 进行降维以消除数据的稀疏性，运用 K-means 算法进行进行聚类分析，根据轮廓系数评价方法可得聚类数目为 50 时，聚类效果最佳。为获得热点问题对留言进行聚类分析，获得聚成每一类的留言时间区间，留言主题通过 BILSTM+CRF 对实体进行识别、留言详情通过 TEXTRANK 获得文本摘要并取得相对应得到时间范围，地点或人群以及问题描述；本文通过提取每一类前 10 个热词的贝叶斯得分加上点赞数作为定义热度评价指标同时计算留言的热点值并按照热度值进行排序，排序结果详见附件。

针对问题三，建立了确定答复意见质量优劣的准则，分别从相关性、完整性、可解释性等三方面对答复意见文本进行量化描述，利用 word2vec 算法对问答数据进行训练，得到该词向量和对应词语的相似度，利用词语相似度构造文本间相似度并建立答复意见评价方案。

关键词：支持向量机、维基百科词、K-means、相似度

一、挖掘目标

近几年，伴随微信、微博、阳光热线等网络问政平台大规模使用，数据的增加呈指数，对海量文本数据的处理是人们所要面临的一个急切且重要的问题，根据题目中所给出的已知信息，分析相关数据并结合实际情况，运用数学思想，建立相关的数学模型来研究“智慧政务”中的三个问题：

1）群众留言分类：利用 jieba 分词和支持向量机的方法对非结构化文本数据进行一级标签分类，并用 F-Score 对分类结构进行评价。

2）热点问题挖掘：对附件 3 留言详情进行 K-means 聚类的文本挖掘，提取每一类时间的时间范围、地点或人群以及问题描述，并定义合理的热度评价指标根据附件 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-热点问题表的格式给出排名前 5 的热点问题，按表 2-热点问题留言明细表的格式给出相应热点问题对应的留言信息。

3）答复意见的评价：针对附件 4 相关部门对留言的答复意见，利用 word2vec 算法计算文本间的相似度，并从答复的相关性、完整性、可解释性角度对答复意见的质量给出一套评价方案。

二、问题分析

2.1 问题一的分析

问题一要求建立关于留言内容一级标签分类模型，为相关部门提供便利。基于以往人工分类的文本处理方式过于烦琐，综合分析影响中文文本分类准确率的相关因素，对比传统的机器学习和深度学习的方法。利用有限的文本数据，在模型的学习能力与复杂程度中寻求需求最佳折中方法，因此，本文使用支持向量机 SVM 进行中文文本分类[1, 2]。

本文对附件 2 中的留言详情进行中文分词，得到对应的词语集构建词向量空间，但由于词语数据过多会导致词向量的维度过高，从而影响分类器的性能和准确率，因此需要对分类性能影响较大的词语进行特征选择，把原本的词抽象成一个向量化的样本集合并与训练好的词模型进行相似度计算，迭代计算直至找到最相近的类别。

2.2 问题二的分析

首先，对附件 3 数据中的留言详情进行预处理、特征选择和 tf-idf 权重计算之后，将文档表示成具有高纬、稀疏的文档-词矩阵，接着进行 PCA 降维，进而使用 K-means 进行聚类，对于聚餐多少类 K 值的评估标准，采取使用 Calinski-Harabasz(CH)指数来评价 K-means 聚类模型，Calinski-Harabasz(CH)指数越大则聚类效果越好。

其次，将聚成的每一类数据，通过 pandas 读取留言时间获得最大最小时间，接着通过 BILSTM+CRF 算法[4]提取对应的地点和人群，对每类留言详情进行句子划分通过 TextRank 算法得到问题描述。

最后，通过贝叶斯平均法得到每类留言详情每个热词的得分，然后统计每个的前 10 个热词累计得分再加上点赞得分，最终确定热度评价指数。按表 1-热点问题表的格式给出排名前 5 的热点问题，并保存为文件“热点问题表.xls”。按表 2-热点问题留言明细表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”。

2.3 问题三的分析

针对问题三，根据附件 4 已有的答复数据，文中建立了判断答复意见的质量优劣的评价方案，跳出传统的“相关性”概念，并从多个角度考虑，答复意见质量依赖于答案本身所包含的信息、答案的完整性以及答案的新颖性等因素，并从 3 个不同的角度对答复文本进行量化描述，通过 word2vec 算法对答复文本进行训练，得到词的相识似度，利用词语的相似度获取文本间相似度。于此，本文建立答复意见评价方案模型。对数据答复文本进行量化分析，并能很好地从答复的相关性、完整性、可解释性评估答复质量的优劣。

三、模型假设

1. 假设提供的文本数据都是真实的。
2. 假设忽略其它因素对群众留言时的情绪影响。
3. 假设答复意见在规定的时间内进行回复。
4. 假设不存在无效问题。

四、符号说明

符号	说明	单位
F	答案质量评价值	
β_{sim}	相似度阀值	
HOT	热度指数	
WR	热词累计	
UP	点赞数	
Q	文档	

五、模型建立与求解

5.1 问题一:建立留言内容的一级标签分类模型

（1）文本预处理

（一）数据清洗

如果在一开始未进行数据清洗，那么最终会影响模型分类的准确性，清洗数据一般包含去除缺失值和重复数据、过滤标签文本，还有影响分类效果的早上处理。

（二）中文分词

中文分词是指将句子分成一个个独立的词，是特征提取中最常用的方法，不同于英文有天然的空格间隔作为词与词的间隔标志，中文文本中词的提取需要基于序列预测等方法的分词技术来实现。

本文分词使用当前广泛使用的基于Python的中文分词工具jieba,jieba小巧高效，支持多种模式进行分词，能有效解决未登录词和歧义词。

（三）停用词过滤

停用词，指语言表达中一类没有意义的词，比如介词、代词等不包含或包含极少语义的词。这些词在文本中大量的存在，但不能反映文本信息。通过过滤停用词，自动忽略某些字或者词语，利于模型更好地去拟合实际的语义特征和增强泛化能力。考虑原始数据中存在口语化、根据政务文本情景采用《哈工大停用词》。

5.1.2 文本的向量空间模型

gensim 工具包的 word2vec 可以把文本表示为一个词向量，词向量就是词向量空间里面的一个向量，即向量的每个特征表示为文本中出现的词。

向量空间模型进行文本表示时，要完成两个步骤：

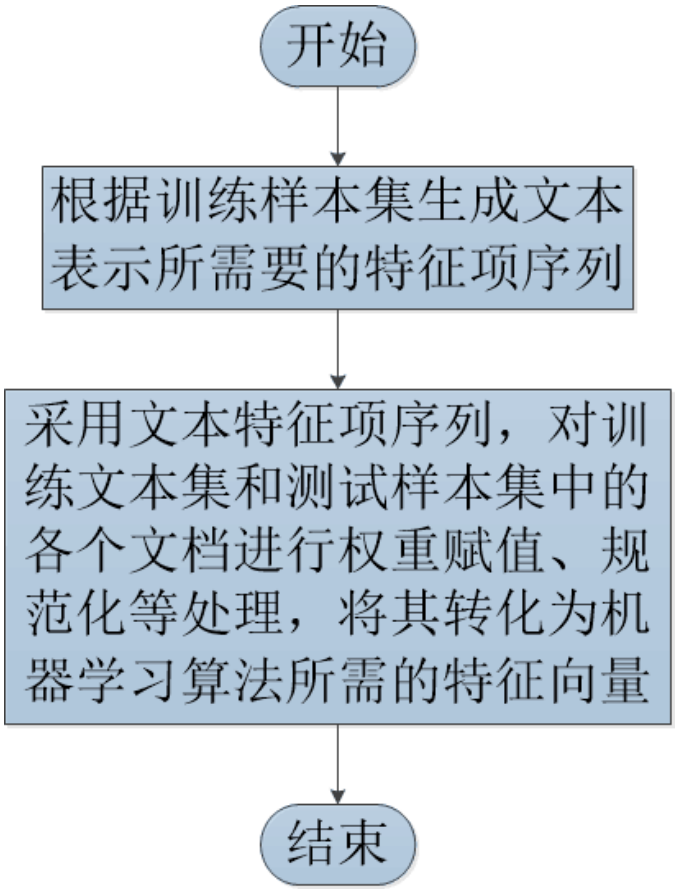


图 1 向量空间模型进行文本表示步骤

向量空间模型假设一个文档 $\mathrm{Q}=Q\left(t_1,W_1;t_2,w_2;...;t_n,W_n\right)$，D 符合以下两条要求：

1）每个特征项 $\mathrm{t}_k\left(1\le k\le n\right)$ 没有重复；

2）每个特征项 t_k 无先后顺序关系；

在满足以上两个要求下，可以把特征项 $\mathrm{t}_1,t_2,...,t_n$ 看成一个 n 维坐标系，而权重为 $\mathrm{W}_1,W_2,...,W_n$ 相应的坐标值，n 维坐标系中文本表示为维空间中的一个向量，称 $\mathrm{Q}=\mathrm{Q}\left(t_1,W_1;t_2,w_2;...;t_n,W_n\right)$ 为文本 Q 的向量或向量空间模型，如下图所示

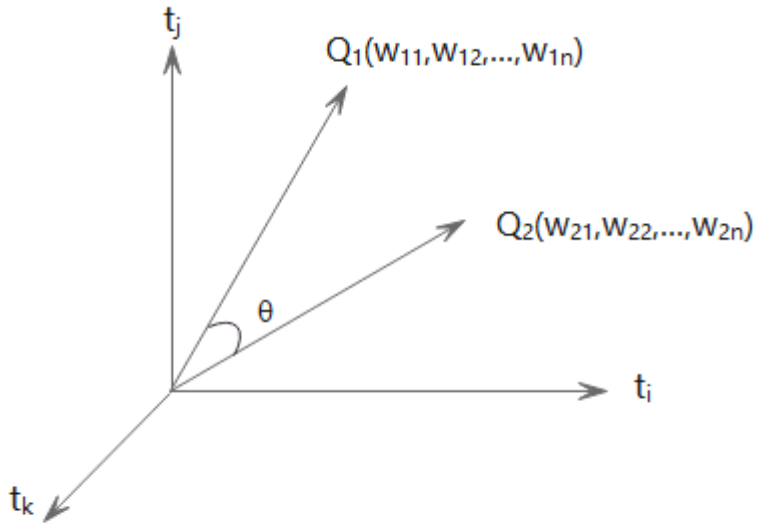


图 2 向量空间模型

向量空间模型是把非结构化的文本数据处理成可以进行向量运算的向量，相似度表达语义，简介明了，对于每一段文本，获取其每一个分词在 word2vec 模型的相关性向量，然后把文档的所有分词在 word2vec 模型中的相关性向量求和取平均数，当文本被表示为文本空间的向量，通过计算向量之间的相似性来度量文本间的相似性。计算两个向量相似性度量方式常用的方法是余弦距离。

5.1.2 文本特征处理

将文本转化为一定格式的特征编码后计算机可以直接识别自然语言，转化后可以携带更多的文本特征，提升分类算法预测对应的类别的准确率。

若文本中出现的词是词空间很少的一部分，那么文本特征就会表示非常稀疏，增加了分类算法的时间长度与空间复杂度，所以筛选重要的文本特征会提高分类性能。特征选择的步骤如图所示：

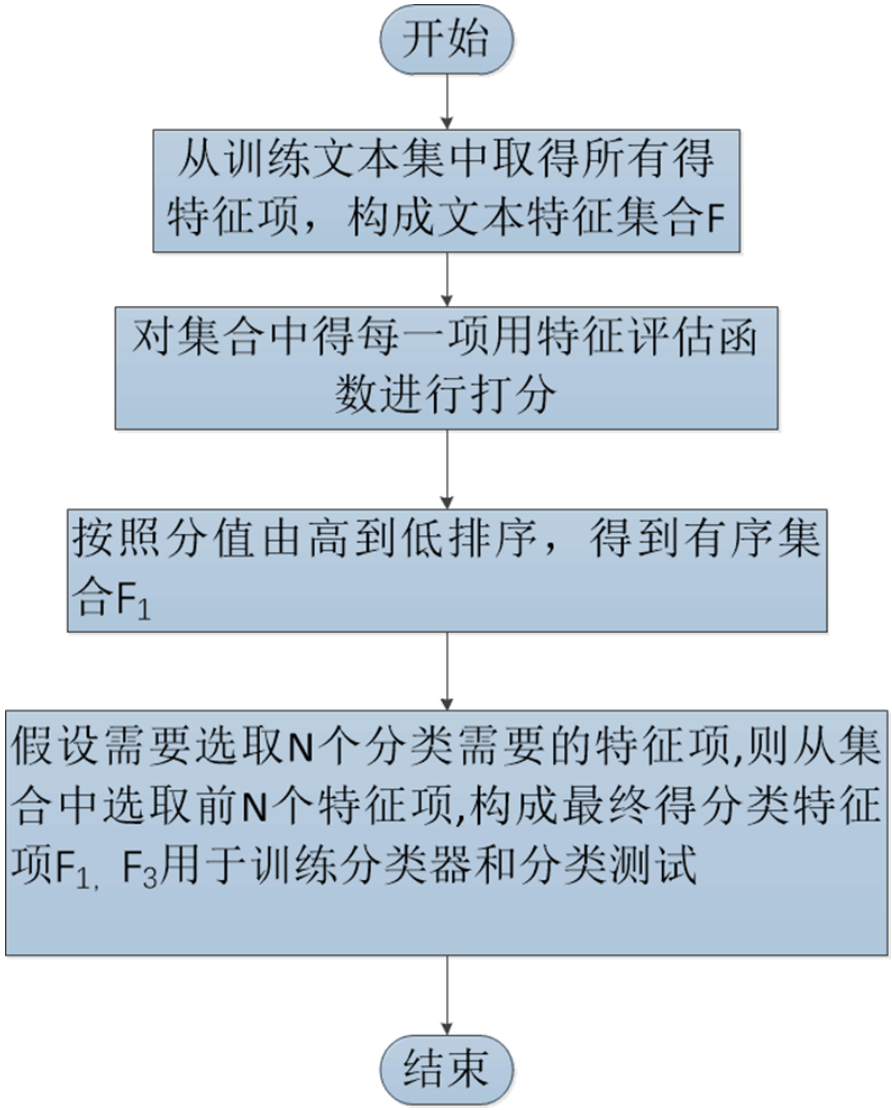


表 3 特征选择步骤

特征选择指对现有的特征空间筛选一些有代表性的特征重新组成新的特征集。通过映射可以把高纬度空间投影到低纬空间，从而完成降维，特征抽取和特征选择都是降维直接有效的方法，可以挑选出有助文本分类的特征。

特征选择的方法：信息增益法

信息增益为某一特征在文本中出现前后的信息熵之差。假定 c 为文本类变量,C 为文本类的集合,d 为文本,f 为特征（以下各节同此）。对于特征 f 其信息增益记为$IG(f)$,公式如下:

$$IG(f)=H(C)-H(C|f)=-\sum_{c\in C}p(c)\log(p(c))$$

$$+p(f)\sum_{e\in C}p(c|f)\log(p(c|f))+p(\overline{f})\sum_{c\in C}p(c|\overline{f})\log(p(c|\overline{f}))$$

$$=\left(\sum_{c\in C}\left(P(c,f)\right)\log\left(\frac{P(c,f)}{P(c)P(f)}\right)+P\left(c,\overline{f}\right)\log\left(\frac{P\left(c,\overline{f}\right)}{P(c)P(f)}\right)\right) \qquad (1)$$

只考虑单个类的时候则有：

$$\mathrm{IG}\left(c,f\right)=P(c,f)\log\left(\frac{P(c,f)}{P(c)P(f)}\right)+P\left(c,\overline{f}\right)\log\left(\frac{P\left(c,\overline{f}\right)}{P(c)P(f)}\right) \qquad (2)$$

当特征的取值较多时，信息增益能够帮助预测类别属性的信息量，特征划分可以得到对应子集，但与此同时熵更低，但因为熵划分前是保持不变的，所以信息增益变大了，同时会偏向取值较多的特征。

5.1.2 问题一模型的求解

文本分类的算法很多，常见的贝叶斯、逻辑回归，KNN，本文通过实验对比多种分类算法，最终选着 SVM。SVM 提取了分离两个类的最佳超平面或线，通过求解最优分隔超平面来得到高分类准确率的分类器，同时 SVM 有着良好的稳定性，已经成为应用效果良好的分类技术。本文模型是在文本上训练得到的 SVM 模型，并使用该模型预测分类效果。支持向量机的主要思想可概括为两点:

1)当数据集线性可分时，寻找线性分类对应的最优分类面，对于数据集线性不可分时，首先使用非线性映射算法将原始数据映射为高维特征空间，然后对高维特征空间进行线性分类;

2)基于最小化风险的思想,在数据集的特征空间中寻找最优的分割超平面，使不同分类数据之间的距离最大，并进行最优分割超平面的二次规划,使得算法模型能够得到全局范围内的最优解。

设给定的样本数据集为$\{(\mathrm{x}_1,y_1),(x_2,y_2)...(x_n,y_n)\}$，其中$x,y\in R$为输入向量，$\mathrm{y}\in\{1,-1\}$为类别标号，m 为输入向量的维数，n 为样本数目。对于线性可分的情况，样本数据集被一个超平面分割为两部分， 记这个超平面为 H，如公式(3)所示。

$$\mathrm{H}=\{x\,|\,w\times x+b=0\} \qquad (3)$$

(1)式中，W 为 n 维向量$[W_1,W_2,...W_n]$;各个维度值为输入向量的权值，b 为线性判别函数的偏移值。若样本能够按照各自正确的类别分开，在样本空间里，点到超

平面 H 的距离为:

$$\frac{|\mathrm{w}\times \mathrm{x}_i+\mathrm{b}|}{\sqrt{w_1^{\ 2}+w_2^{\ 2}+\ldots+w_n^{\ 2}}}\geq\frac{1}{\|w\|} \qquad (4)$$

当点到超平面之间的距离最大时，说明不同分类数据集之间的间隔最远，即模型的分类效果最好。对于线性不可分的问题，可以通过选取适当的核函数把线性样本 X 映射到某个高维空间 P,然后再在高维空间里使用线性分类器。因此，对于非线性数据集,有:

$$\min mizef\left(w\right)=\frac{1}{2}\|W\|^2 \qquad (5)$$

$$\mathrm{s}\times t\times y_i\left(wx+b\right)01,i=1,2...,n \qquad (6)$$

SVM 的关键在于核函数,通过核函数可以将数据集映射到高维空间,解决原始数据在低维空间中无法线性划分的问题,本文选择径向基函数作为留言文本分类支持向量机的核函数。

文本分类性能评估指标是正确率、召回率和 F-测度值。分类结构建立的混淆矩阵如下表所示：

文本与类别的关系 分类器对二者关系的判断	正确	不正确
YES	s	t
NO	u	v

表 1 分类结构混合矩阵

表 1 假设，正确分类为正样本，错误分类为负样本，s 表示分类器输入正样本被正确的预测为正样本，t 表示分类器输入负样本被正确的预测为正样本，u 表示分类器输入正样本被正确的预测为负样本，v 表示分类器输入负样本被正确的预测为正样本.通过召回率、正确率和 F-score 分别采用以下公式：

$$\text{准确率：}\mathrm{r}=\frac{\mathrm{s}}{s+t}\times100\% \qquad (7)$$

$$召回率：p=\frac{s}{s+u}\times 100\% \qquad （8）$$

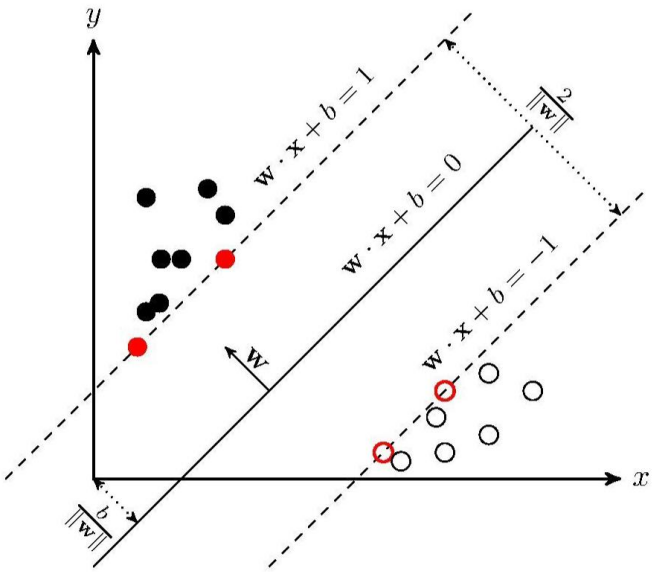
$$F-score：\mathrm{F}_{\beta}=\frac{\left(\beta^{2}+1\right)\times p\times r}{\beta^{2}\times p\times r} \qquad （9）$$

设 β =1，这时的评价指标变为：

$$F-score=\frac{2\times p\times r}{p+r} \qquad （10）$$

基于支持向量机的留言文本分类模型

支持向量机（SVM）学习算法的核心思想是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。如下图所示，$wx+b=0$ 为分离超平面，对于线性可分的数据集来说，这样的超平面有无穷多个（即感知机），但是几何间隔最大的分离超平面却是唯一的。对于线性不可分的数据，利用核函数将数据映射到高维空间后，在高维空间寻找最优的分离超平面，SVM 优点是利用少量的训练样本就能获得很好的训练效果，算法被广泛应用于各类分类问题和回归问题中，本文在实验中，将 SVM 作为算法的分类器，留言详情文本作为特征向量构建群众留言分类模型。模型实验结果如图 4 所示。



```
准确度：
 0.8767643865363735
Classification report for classifier:
              precision    recall  f1-score   support

        交通运输       0.81      0.75      0.78       122
      劳动和社会保障       0.91      0.93      0.92       393
        卫生计生       0.92      0.84      0.88       180
        商贸旅游       0.79      0.80      0.80       240
        城乡建设       0.82      0.86      0.84       393
        教育文体       0.93      0.92      0.92       332
        环境保护       0.97      0.92      0.94       182

   micro avg       0.88      0.88      0.88      1842
   macro avg       0.88      0.86      0.87      1842
weighted avg       0.88      0.88      0.88      1842
```

图 4 模型运行结果

准确率和召回率是相互制约的，单纯的看准确率和召回率是不够的，为了平衡准确率和召回率的影响，较全面的评价一个分类器，引入 $F_1-Score$ 作为综合指标，从图 4 可知，模型的 F1-Score 为 0.88。图 4 表明，留言内容的七个类别的 F1-Score 均在 0.80 以上，其中教育文化、环境保护类别的 F1-Score 在 0.90 以上。实验结果表明，模型具有很强的实用性。

5.2 问题二：热点问题挖掘模型的建立与求解

5.2.1. 问题二模型建立与求解

1、利用文本聚类法将相似文本进行归类

K-means 文本聚类[1]是将相似的文本聚集到同一个簇，使同一类的数据尽可能聚的在一起，不同的数据尽量分离开来。通过将留言详情的问题进行聚类，从而可以得到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。

K-means 文本聚类算法的步骤如下图:

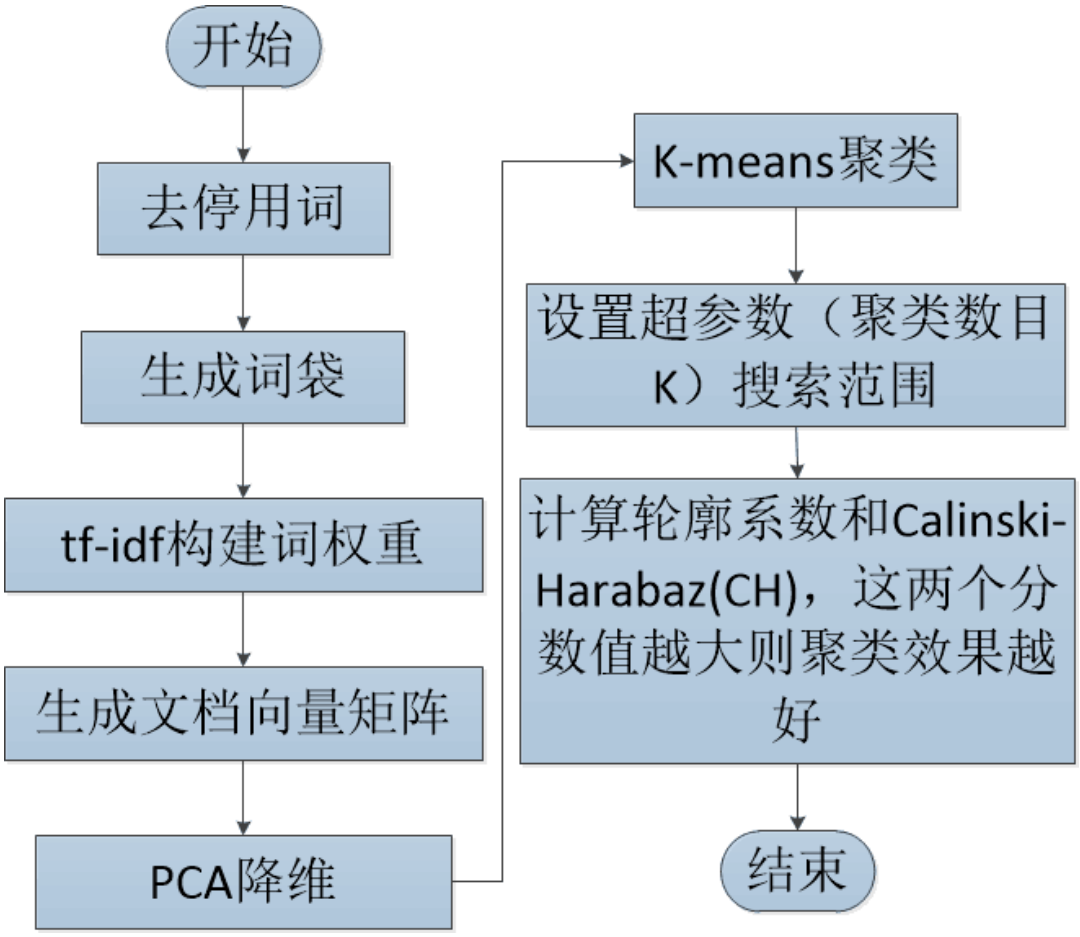


图 5　K-means 中文文本聚类流程图

对于聚类效果的评估方式，本文根据 Calinski-Harabaz(CH)的大小判断，用簇间的协方差矩阵与簇内的协方差矩阵相除,CH 系数越大表示类自身越紧密，类与类之间越紧密，聚类效果越好，反之越差.

$$s(k)=\frac{tr(B_k)}{tr(W_k)}\frac{m-k}{k-1} \qquad (11)$$

其中 m 为训练集样本数， k 为类别数。 B_k 为类别之间的协方差矩阵， W_k 为类别内部数据的协方差矩阵。 tr 为矩阵的迹。

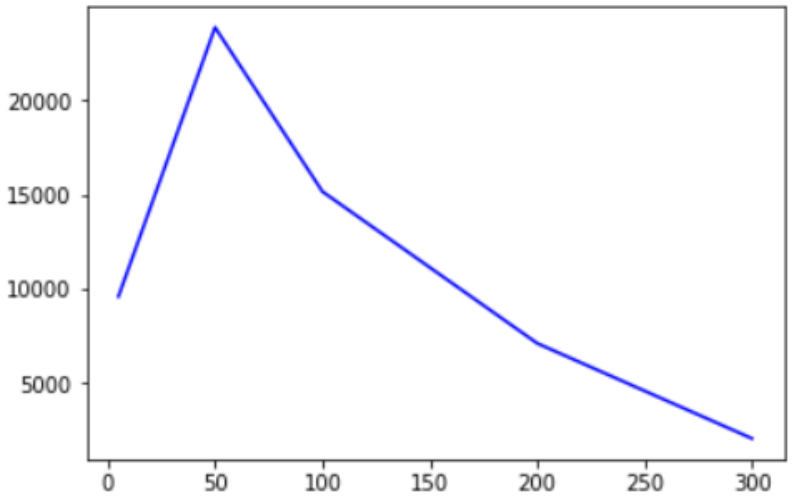


图 6 不同 k 对应的聚类效果

由上图 6 可知，当 k=50 时，Calinski-Harabaz(CH)最大，聚类效果越好，当 50<k<100 是，下降梯度过快，当 k>100 时聚类效果越来越低。

（2）BILSTM+*CRF* 获取每类地点或人群

命名实体识别是以名称为标识的实体，可以自动地从文本数据中识别出特定类型的命名实体，将原始文本数据数据 nre 工具里，NER 工具会给文本序列中的每一个字(或词)打上一个标签，用来表示这个字（或词）是否为命名实体的一部分。

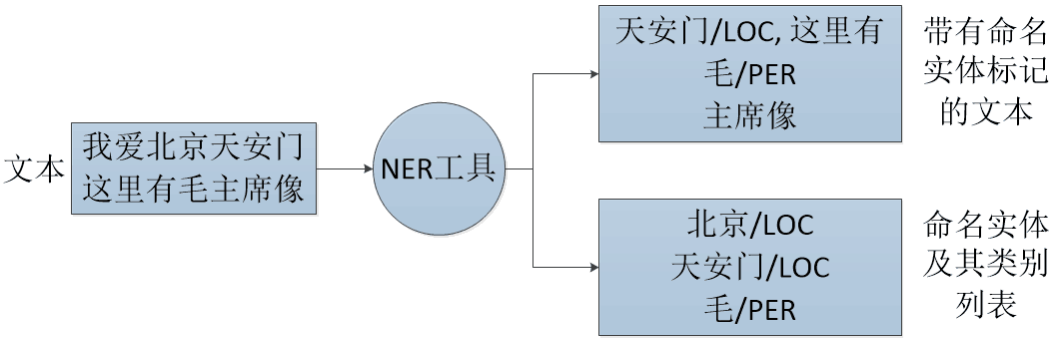


图 7 命名实体流程图

本文基于 K-means 聚类后每一类留言主题用 BiLSTM+CRF 做命名实体识别，从中提取出地点或人群，对应的是得到标记为 LOC 的文本，其中 BILSTM+CRF 有两层，一层是 BiLSTM，一层是 CRF。

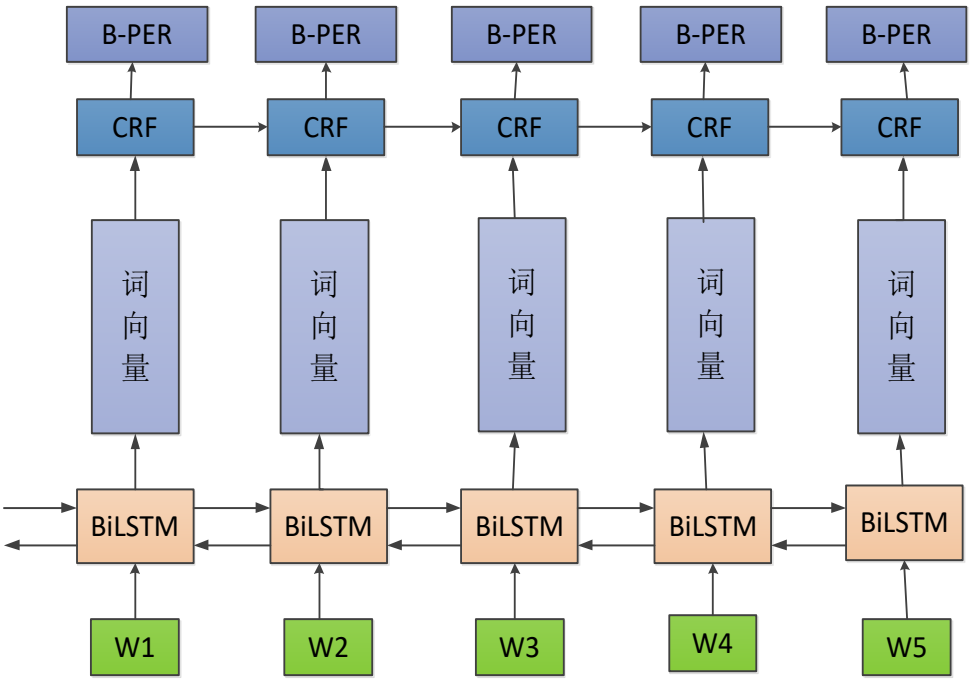


图 4 LSTM-CRF 结构

词向量输入到 BiLSTM 层，每个标签的预测分数就是输出值，CEF 层的输入便是这些预测分数。CRF 层的损失函数 Loss 可由下式计算：

$$Loss = \frac{p_{RealPath}}{P_1 + P_2 + ... + P_N} = \frac{e^{S_{RealPath}}}{e^{S_1} + e^{S_2} + ... + e^{S_N}} \qquad (12)$$

其中真实路径 Si 使用维比特算法计算，分母的归一化项，即所有可能路径得分之和使用动态规划计算，具体到每条路径的得分 Si 可由下式计算 $S_i = Emission\ score + Translation\ score$，在 BILSTM+CRF 模型中只需要将 LSTM 层的输出通过一个线性全连接层，就可以直接得到每种标签的得分，因此，在 BILSTM+CRF 模型中只需要考虑状态转移分数，即 *Translation score* 。

```
precision:  0.879801967461613
recall:  0.8438102288111291
fscore:  0.8614303160277603
```

图 8 BLSTM+CRF 评价方法

经模型训练，模型的 f-score 为 0.86，比传统的 HMM 和 CRF 训练模型效果更优。

（3）中文文本摘要

文本摘要是对某个或多个文档进行概况，既可以保证文档的主要内容，能在众多文本中找到有效的信息，保证简洁而有意义。

本文使用 TextRank 生成式文本摘要得到每一类留言详情的问题描述，TextRank 算法的是将文本划分成若干词或句子，全程矩阵运算，速度快，通过句子之间的相似度，而不是单纯计算某些句子含有的关键字数量，考虑到了整个句子的信息，采用矩阵迭代收敛的方式对节点进行排序，得到关键词或摘要句[1]。文本摘要的流程图如下：

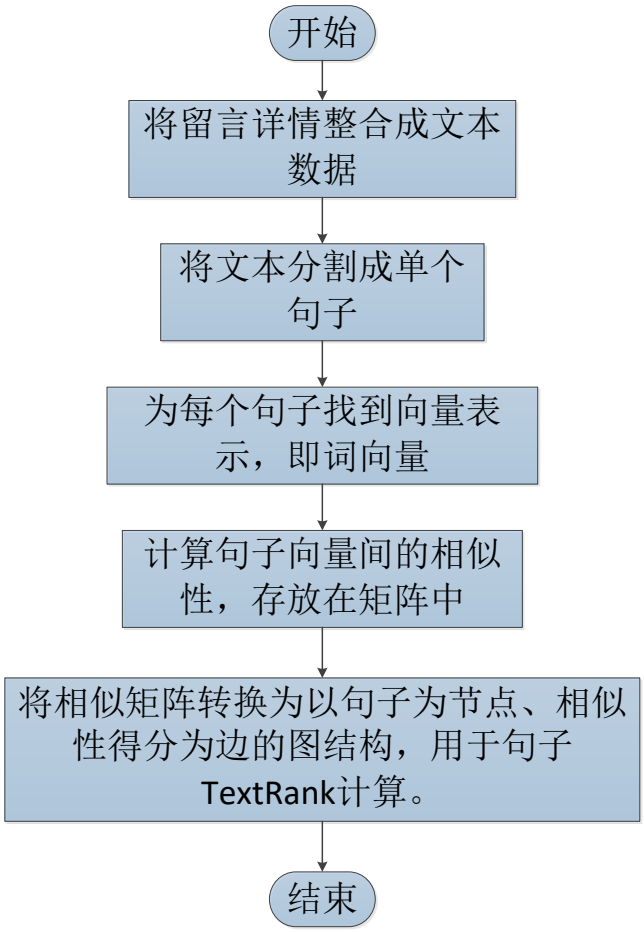


图 9 文本摘要流程图

（4）构建热度问题评价指标

热点问题影响范围大、长期存在并且被许多人关注。热点问题的关注度也有高低。根据热词形成热词的词云表，如下图：

图 10 热词五角星词云

本文通过贝叶斯平均法得到热词得分，贝叶斯平均法可由下式计算：

$$WR=\frac{v}{v+m}R+\frac{m}{v+m}C \qquad (13)$$

其中 WR 是每个词的加权得分，WR 越大表示热度越大，R 是该词汇的平均得分，v 是总词频，m 是排名前 n 的词汇的最低词频，其中 n 是自定义的阈值。

根据附件 3 的留言详情，使用贝叶斯平均法得到每类热词的得分，由于群众对某一问题的反映并不总是对网络问政平台留言，有部分群众会点赞他认同的留言，这对热点问题的排序影响很大，因此，本次热点问题的排序采用每类前 10 个热词累计得分加上点赞得分，点赞得分即为点赞数的多少。

$$HOT=WR+UP \qquad (14)$$

其中 HOT 为热度指数，UP 为点赞数。

最后，将群众反映的问题归类并且进行描述后，将留言主题相同的进行时间范围的提取，使用 pandas 读取聚类的每条数据，找到 max()和 min()，每类的时间的最大值与最小值即群众提出该问题的时间范围。详细热点排名见附件。

热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述
1	9	7.869	2019/9/1至2019/1/8	请政府救救广铁集团	请政府救救广铁集团的职工吧
2	15	7.844	2020/1/8至2019/1/11	金色溪泉湾	A7县小塘路还修吗
3	37	7.81	2020/1/7至2019/1/10	A4区大塘村陈家冲组	A市A3区兰亭湾畔小区违规开餐饮
4	32	7.646	2020/1/8至2019/1/11	A7县第三中学	请问A市西湖街道茶场村五组这边有什么规划
5	38	7.627	2020/1/2至2018/5/17	A市地铁扫码	A市A2区龙湾国际社区违建严重

图 11　热点问题表

热度排名前 5 如图 11 所示，图 5 展示了每个热度留言所对应的时间范围、地点/人群、以及问题描述。

5. 3 问题三：答复意见评价方案

5.3.1 答复意见质量影响因素

如何识别意见答复质量是“智慧政务”问政平台应用亟需解决的重要问题依赖于答案本身所包含的信息、答案的完整性等因素，本文从答复的相关性、完整性、可解释性等角度对答复意见的质量进行评价。根据答复信息数据，对答复质量优劣进行归纳，等到以下优劣评判的特征。

（1）良好答复的参考特征：

1）答复意见和专业领域词语的语意相似度高；

2）答复内容充实，有相关依据 。

（2）较差回复的参考特征：

1）答复内容过于简短，例如“以获悉”，“请来电咨询”，

2)答复内筒套用固定模板例如 ，“具体 情况具体分析” “详情咨询”等，

3)对不同问题使用同一回复，或回复内容的相似度很高.

5.3.2 答复意见的描述方法

（1）相关性

两个文本之间的相似度可以用表示两个文本的词语间的语义相似度[5]替代，减少同义词、多义词的歧义。如果两个文本相似度高，则表明其存在相关性，即同一层词语所在层分支的密度和词语之间重合度。

Word2vec 是一种结合了哈夫曼编码词的向量模型。通过对模型的训练可以降低词向维度，获得定长连续的词向量。通过计算两个向量距离余弦值来计算两个向量所表示的词的相似度。

文本由关键向量构成，同时这些关键词也反映该文本信息。所以：

$$A_k = \{w_{k1}, w_{k2}, ...w_{ki}\} \qquad (15)$$

其中 A_k 由 k 个词向量组成的文本向量集，继而推出文本向量 A_n 和 A_m 的相似矩

阵S_{nm}=textsim$\left(w_{ni}, w_{mj}\right)$，其中，i=1, 2, 3, …n, j=1, 2, 3, …m.

设全体回复构成集合 R, 则全体 重复答案对为：

$$A=\{<A_i, A_j>|A_i, A_j \in R, i<j\} \quad (16)$$

给定相似度阈值 β_{sim}, 定义复制集合:

$$A=\{<A_i, A_j>|A_i, A_j \in A, textsim\left(A_i, A_j\right) \geq \beta_{sim}\} \quad (17)$$

所以$|CP|/|A|$可以表示为答复意见的文本相似度，若接近$|CP|/|A|$接近 0 则无关性小，反之大，由此定义评价项 F_1 为：

$$F_1=\frac{|CP|}{|A|} \quad (17)$$

（二）完整性

答复意见的详细程度与答复文本长度相挂钩，较短答复的回答信息量一般不够，并且缺乏足够的细节，回复不具完整性，尤其是只回某个词的答复，得分应该较低，相反，较长的答复过于冗长，得分不应该过高，因此，通过使用对数函数来量化答复的文本长度与得分的关系, 建立“回复内容是否完整”的评价项 F_2 :

$$F_2=\frac{1}{N_0}\sum_{i=1}^{N_0}\log_m L_i \quad (19)$$

其中, L_i 为针对第 i 个问题回复的文本长度, m 为常数.

（三）可解释性

文本答复的内容需要根据留言详情的问题作出逻辑和准确的表达，即答复信息是由有理有据、可解释问题。假设文本数据共 N_0 条规范的答复信息，然后使用“智慧政务”相关类别中的关键词和答复文本进行语义匹配，若在答复的文本中匹配到相应的关键字为其一级分类专业词, 则认为该条答复对此问题进行了相关问题的回答. 统计答复信息引用类别的回答数目 N_{law}, 计算出现频率, 即 N_{law} 与

N_0的比值.该频率值可以作为“是否可解释”的评价项,记为 F_3:

$$F_3 = \frac{N_{law}}{2_0} \quad (20)$$

5.3．2 答复意见评价模型

答复意见评价模型的关键在于如何建立对答复质量的量化评分模型,根据上述量化方法,给出答复意见评价模型,接着对三项量化指标进行整合,以计算答复信息的得分情况,建立答复信息的质量评价函数 F ,即构造如下，答复质量的评价值F:

$$\begin{cases} F = M_k \bullet \lambda^T \\ \lambda = (\lambda_1, \lambda_2 \lambda_3), M_k = (F_1, F_2 F_3) \end{cases} \quad (21)$$

其中， λ^T 为λ向量的转置向量,向量λ和M_k为反映回复信息不同侧面的权重向量和得分向量，权重向量λ是按答复质量的重要性确定，分别对应的权重为0,35,0.3，0.35，满足权重相加为1，由上述公式可知F越大，则答复意见质量越高，反之则地。

六、模型评价与优化

6.1 中文文本分类评估

支持向量机算法是常用文本分类算法中比较优秀的方法之一,因其具有完善的数学理论，较为成熟，可解释性强，可以找出任务中影响重要的关键文本，相比准确率更高，新颖程度的其它算法，更适合作为实际的应用中

6.2 训练好的word2vec中文文本分类优化

原始数据量有限，文本特征表达能力不足，这里使用 gensim 训练好的word2vec，Word2vec 的模型是想通过机器学习的方法来达到提高上述任务准确率的一种方法,由此提出基于 word2vec 维基百科词模型通过 word2vec 训练维基百科语料库并得到对应词模型，建立文本特征表达方法， 得到留言详情一级

分类报告如图 12：

	precision	recall	f1-score	support
0	0.85	0.74	0.79	111
1	0.88	0.95	0.92	396
2	0.93	0.84	0.88	194
3	0.83	0.86	0.85	256
4	0.85	0.85	0.85	393
5	0.95	0.92	0.93	313
6	0.94	0.94	0.94	179
micro avg	0.89	0.89	0.89	1842
macro avg	0.89	0.87	0.88	1842
weighted avg	0.89	0.89	0.89	1842

图 12 文本分类评价方法

由图 12 可知，预测的结构的 f1-score 为 0.89，相比自己训练词向量模型，准确率提高了百分之一，同样可以用于实际应用。

6. 3 答复意见评价模型的优化

由原始文本训练的词向量文本集数据有限，下一步扩大训练词向量的文本集扩从相关一级分类的专业词汇，以提高文本相似度，增加答复质量评分项，以提高答复质量判断的准确度。

参考文献

[1] 马存.基于Word2Vec的中文短文本聚类算法研究与应用[D].中国科学院大学(中国科学院沈阳计算技术研究所),2018.

[2] 徐蔚.基于深度学习的中文新闻文本分类的研究[D].中南民族大学,2018.

[3] 李娜娜基于TextRank的文本自动摘要研究[D].山东师范大学,2019.

[4] 顾溢.基于BiLSTM-CRF的复杂中文命名实体识别研究[D].南京大学,2019.

[5] 马付玉.中文短文本语义相似度计算方法研究[D].西安科技大学,2019.

附录

1、代码运行环境 Python3、Jupyter NoteBook

2、热点问题挖掘相关代码（具体代码请见附件

```
import pandas as pd
import datetime #引入时间库
import jieba
from snownlp import SnowNLP
import datetime
from pre_processing import *
df_data = pd.read_csv("redian.csv",encoding='utf-8')

datatime = []
description =[]
str_list = []
WR = []
WR_count = []


#定义热词得分函数
def rechi(aaa):
    stopwords = [line.strip() for line in open("stopwords.txt").readlines()]

    words  = jieba.lcut(aaa)
    counts = {}
    for word in words:
        #不在 data 里的停用词表中
        if word not in stopwords:
            #不统计字数唯一
            if len(word) == 1:
                continue
            else:
                counts[word] = counts.get(word,0) + 1
    items = list(counts.items())
    items.sort(key=lambda x:x[1], reverse=True)
    for i in range(10):
        word, count = items[i]


    #根据公式计算热词
    counts = 0
```

```
    for i in range(len(items)):
        word, count = items[i]
    #     print(items)
        counts += count
#    print(counts)#词频的统计
    C = len(items)/counts
    count_sum = []
    for i in range(10):
        word, count = items[i]
    #     print ("{:<10}{:>7}".format(word, count))
    #     count_sum.append()
        WRS = (counts/(counts+count)*C) + ((count/(counts+count)))
    #     print(WR)
        WRS = round(WRS,3)
        WR.append(WRS)
    return WR

    ners = []
for i in range(50):

    #问题描述
    data = df[df['总的类数'] == i] # 判断等式是否成立
    maxdata = data['留言时间'].max()
    mindata = data['留言时间'].min()
    time = maxdata +'至'+ mindata
    datatime.append(time)

    #问题描述
    text = data['留言主题']
    content= ("。".join(i for i in text))
    s = SnowNLP(content)
#     print(s.summary(1))
    description.append(s.summary(1))

    #地点或人群
    texts = data['留言主题']
    sentence= ("\n".join(i for i in texts))
    entity_name, entity_tags = predict(sentence)
    ners.append(entity_name[0])

    #热词度
    a2 = data['留言详情.1']
    rdci = str(a2)
    WR = []
```

```
    WR = rechi(rdci)
    #计算每一类的全部值相加
    total = 0
    for j in range(0, len(WR)):
        total = total + WR[j]
    totals = round(total,3)
    WR_count.append(totals)

#获得地点人群列表
for i in range(len(ners)):
    str = ners[i].split("\n")
    num = str[len(str)-1]
    str_list.append(num)

string=[] #问题 ID
for i in range(1,51):
    string.append(i)

test1=pd.DataFrame(columns=['问题 ID'],data=string)
test2=pd.DataFrame(columns=['热度指数'],data=WR_count)
test3=pd.DataFrame(columns=['时间范围'],data=datatime)
test4=pd.DataFrame(columns=['地点/人群'],data=str_list)
test5=pd.DataFrame(columns=['问题描述'],data=description)
# test = test1 + test2
test2

result = pd.concat([test1,test2,test3,test4,test5], axis=1)
result.sort_values(by="热度指数" , ascending=False,inplace=True)
result

test6=pd.DataFrame(columns=['热度排名'],data=string)
results = pd.concat([test6,result], axis=1)
results.to_csv( '热点问题表.csv',index=False,header=True,encoding="utf_8_sig")
#保存表 1-热点问题表
title_name=['问题 ID','热度指数','问题范围','地点/人群','问题描述 ']
test=pd.DataFrame(columns=title_name,data=[['string'],['WR_count'],['datatime'],['str_list'],['description']])
```