“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，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。本文将基于数据挖掘技术对群众问政留言数据进行内在的信息挖掘，提取我们需要进行分析的部分进行深度挖掘和分析。

针对问题一：首先对文本进行预处理，使用 jieba 模块分词对文本进行分词，去掉停用词，然后利用 DM，DBOW 两种类型的算法分别建立两个 Doc2vec 模型，将两个模型融合，然后用融合的模型将文本转换成文本向量，通过建立 xgboost 模型 ，随机森林模型，logistic regression 模型，高斯贝叶斯模型，SVM 模型，然后计算准确率和 F-sorce，确定使用 XGBoost 模型对文本分类，并对模型进行调参，最后对模型进行交叉验证，确定模型的稳定性，可靠性。

针对问题二：首先对文本进行预处理，提取文本的留言的地名，留言内容取掉停用词提取的地名，建立 word2vec 模型，对文本进行向量化，利用 Mini Batch KMeans 算法对文本进行初聚类，然后按照评价规则提取前五的地区，然后对提取的地区，提取全部的信息，去掉停用词，建立 word2vec 模型，利用 Agglomerative 算法进行再聚类，按照评价规则提取热点前五的问题。

针对问题三：首先对文本进行预处理，提取留言的内容，回复的内容，留言的主题，取掉停用词，建立 word2vec 模型。然后分别从从答复的相关性、完整性、可解释性，时效性，规范性对答复意见给出评价方案。相关性是计算留言的内容和回复的内容的文本相似度，留言的主题和回复的内容的相似度。时效性依据留言评论的时间和回复的时间差，用来评价留言的回复的是否及时。规范性确定一个标准的留言的格式，用正则表达式匹配，用来评价回复的格式规范性。完整性通过计算全部回复长度的分位数，将所有回复区分，评价回复内容的完整性。可解释性利用正则表达式匹配是否提及了相关的条例，法律等，评价回复的可解释性。最后对所有的结果汇总。

关键词：XGBoost; 文本聚类 ;Word2vec ;Doc2vec;Mini Batch KMeans

Application of Text Mining in "Smart Government Affairs"

Abstract

In recent years, with the online inquiry platforms such as WeChat, Weibo, Mayor's Mailbox, and Sunshine Hotline gradually becoming important channels for the government to understand public opinion, gather public wisdom, and consolidate public opinion, the amount of text data related to various social conditions and public opinion has continued to rise, givingThe work of related departments that mainly rely on manual work to divide messages and organize hot spots has brought great challenges.At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of smart government affairs systems based on natural language processing technology has become a new trend in the development of social governance innovation and development.Push role.This article will use the data mining technology to carry out internal information mining on the data of the masses' messages, and extract the parts we need to analyze for deep mining and analysis.

For problem one:first preprocess the text, use the jieba module to segment the text, remove the stop words, and then use the DM and DBOW algorithms to establish two Doc2vec models, fuse the two models, and then use The fused model converts the text into a text vector. By establishing the xgboost model, random forest model, logistic regression model, Gaussian Bayes model, SVM model, and then calculating the accuracy and F-sorce, it is determined to use the XGBoost model to classify the text, and Adjust the parameters of the model, and finally cross-validate the model to determine the stability and reliability of the model.

For problem two: first preprocess the text, extract the place name of the message of the text, remove the place name of the stop word extraction of the message content, establish the word2vec model, vectorize the text, and use the Mini Batch KMeans algorithm to initially cluster the text Then, extract the top five regions according to the evaluation rules, then extract all the information from the extracted regions, remove the stop words, establish the word2vec model, use Agglomerative algorithm for re-clustering, and extract the top five hot issues according to the evaluation rules.

For problem three:first preprocess the text, extract the content of the message, the content of the reply, the subject of the message, remove the stop words, and establish the word2vec model. Then the evaluation plan is given to the response opinions from the relevance, completeness, interpretability, timeliness and standardization of the responses. Relevance is to calculate the text similarity of the content of the message

and the content of the reply, and the similarity of the subject of the message and the content of the reply. Timeliness is used to evaluate whether the reply of the message is timely according to the time difference between the comment time of the message and the time of reply. Normativeness determines the format of a standard message, and uses regular expression matching to evaluate the formativeness of the reply. Integrity By calculating the quantile of the total reply length, all responses are distinguished, and the completeness of the reply content is evaluated. Interpretability Use regular expressions to match whether relevant regulations, laws, etc. are mentioned to evaluate the interpretability of the reply. Finally, summarize all the results.

Keywords:XGBoost; Text clustering ;Word2vec ;Doc2vec;Mini Batch KMeans

目录

1.挖掘目标

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。

本次建模目标是利用收集自互联网公开来源的群众问政留言记录，及相关部门对部分群众留言的答复意见的信息数据，在对数据进行基本的预处理后，利 jieba 中文分词工具对群众留言进行分词、停用词过滤，Doc2vec,Word2vec 的模型进行分词及 XGBoost 模型，Mini Batch KMeans 算法，Agglomerative 算法，对文本进行分类，聚类，计算文本相似度等以达到以下三个挖掘目标:

1) 利用文本分词和文本多分类的方法对留言数据进行文本挖掘，建立关于留言内容的一级标签分类模型，根据分类结果，使用 F-Score 和分类的准确率对分类方法进行评价,确定使用的模型种类，最后对模型进行交叉验证确定模型的稳定性和可靠性。
2) 根据问题的留言，首先进行对留言的地点进行聚类，定义合理的地区评价指标，对所有的留言进行初筛选并给出筛选的结果。给出排名前五的地区，然后对这些地区进行第二次聚类，按照热度的评价指标得到相应热点问题。
3) 根据相关部门对留言的答复意见，计算回复的相似度，长度，时间差，和出现的内容，用来评价答复的质量，并从答复的相关性、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。

2.总体流程与步骤

本文的总体架构及思路如下:

步骤一：文本数据预处理，对附件 2 文本数据预处理，中文文本分词、停用词过滤，把语料整理成 Doc2vec 模型的输入格式，以便后续分析。

步骤二：文本向量化，利用 Doc2vec 训练学习得到词向量，将预处理后的文本转换成文本向量。

步骤三：文本分类，根据文本向量，对各种模型进行对比使用基于 XGBoost 分类器完成文本多分类处理。

步骤四：分类评价，针对文本分类问题提出性能评估的指标，使用 F-Score 指标对文本分类结果和分类器性能进行评估。

步骤五：分类模型评价，对上述的模型进行 5 折交叉验证，用来评价模型的稳定性。

步骤六：留言的地区聚类，对附件 3 文本数据预处理，提取中文，英文，地名信息，建立 Word2vec 模型,利用模型将文本向量化，然后利用 Mini Batch KMeans 算法对文本进行初聚类，按照地区评价指标得到 5 个地区的留言。

步骤七：留言的内容聚类，对上述生成的留言的内容和主题进行预处理，提取中文，去掉停用词，建立 Word2vec 模型,利用模型将文本向量化，利用 Agglomerative 算法，对留言进行再聚类，按照热点指数，得到前 5 的热点问题。

步骤八：文本相似度的计算，对附件 4 文本数据预处理，提取中文，去掉停用词，建立 Word2vec 模型,利用模型将文本向量化,计算向量的余弦作为评价文本相似度的指标。

步骤九：文本出现相关信息的判断，利用正则表达式，对文本出现的字进行判断，得到相应的指标。

3.问题一的分析与解决

3.1 流程图

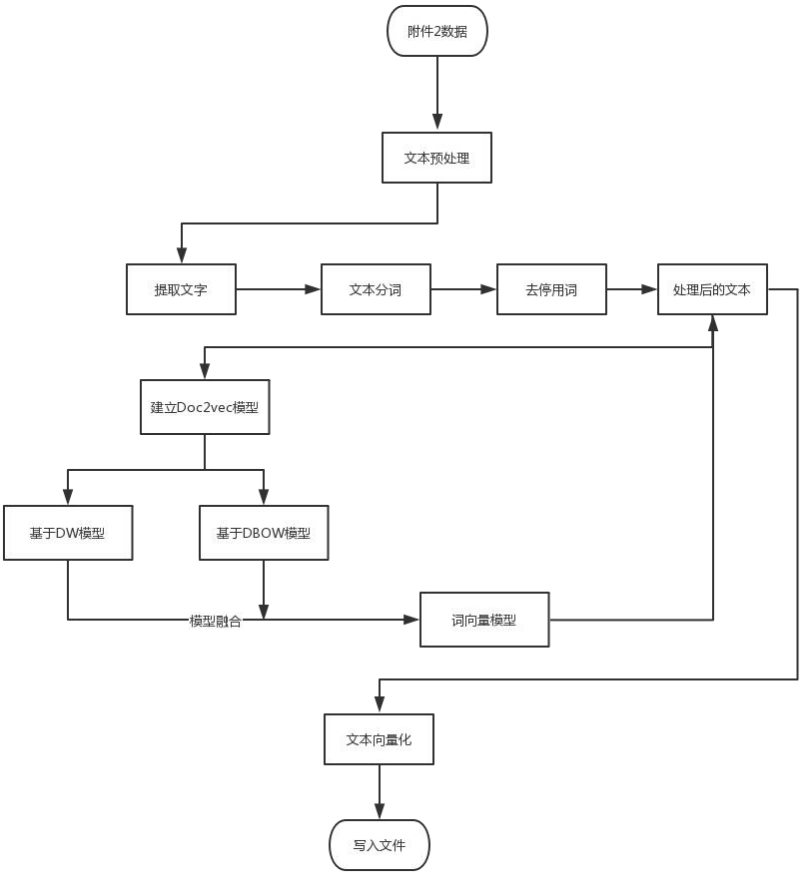


图 1 问题一词向量模型流程图

3.2 数据预处理

3.2.1 提取中文

提取附件二数据中的汉字，并去掉不需要的标点，空格等特殊字符，将数据格式进行统一。

3.2.2 使用 jieba 模块分词

由于中文文本的特点是词与词之间没有明显的界限，从文本中提取词语时需要分词，本文采用 Python 开发的一个中文分词模块——jieba 分词。

jieba 分词属于概率语言模型(Probabilistic Language Modeling)分词。如下列公式所示，其设计思想是从统计的角度来看待分词问题，即输入的是字符串 S=S1,S2,…,Sm,输出的是词串 W=W1,W2,…,Wn 其中 n≤m。那么对某个特定的字符串 S，将有很多个切分方案 W 与其对应，如何在这些切分方案中找出概率最大的一个。即对输入字符串切分出可能性最大的词序列，就是分词过程需要完成的任务。

$$\mathrm{Seg}(s)\arg\max_{Seg} P(W\mid S)=\arg\max_{Seg}\frac{P(S\mid W)P(W)}{P(S)} \tag{3-1}$$

对附件二的留言列表和留言主题数据进行中文分词，jieba 用到的算法有：

第 1 种基于 Trie 树的词图扫描，生成有向无环图(DAG)；

第 2 种采用动态规划(Drrnamic Programming)查找最大概率路径；

第 3 种对于未登录词，采用基于汉字成词能力的隐藏马尔科夫模型，使用 Viterbi 算法。

jieba 分词系统提供分词、词性标注、未登录词识别，支持用户自定义词典，关键词提取等功能。

3.2.3 去掉文章的停用词

分词结束后仍然有大量标点及表达无意义的字词，对后续分析会造成很大影响，因此接下来需要进行停用词过滤。

去除停用词可以大大减小特征词的数量，进而提高文本分类的准确性。停用词主要有两种类型：一是人类语言中包含的功能词。这些功能词非常常见，类似虚词，与其他词相比，没什么实际意义。另一类词是词汇词，像“的”“和”“在”“是”等副词、量词、介词、叹词、数词。对文本分类来说，这些词汇几乎在所有文本中都会出现，不具有特殊性，没有区分度，反而会稀释那些有区分度的词，所以通常会把这些词从问题中移去，从而提高分类性能。

建立一个停用词词典，在分词后将每个词与停止词字典中的条目进行匹配。

如果匹配成功，该单词将被删除。去除停用词后数据保存在附件二的数据处理.txt,部分结果示例：

图 2 数据预处理结果

3.3 模型的建立

3.3.1Doc2vec 模型的建立

Doc2vec [1]是一个无监督框架，学习文本段落的连续分布向量表示，文本可以是可变长度的，从句子到文档。该方法可以应用于可变长度的文本，任何句子或大型文档。通过该方法将文本进行数字化表示，每个词语使用几百维的实数向量表示，然后使用欧式公式计算两个词语的语义相似性。本文依赖于 PV-DM 模式与 PV-DBOW 模式分别建立了 DM.model 和 DBOW.model。

3.3.1.1PV-DM 模式

PV-DM 下段落向量的框架[2]，该框架类似 word2vec 中的连续词袋模型框架。唯一的不同点在于：通过矩阵 D，额外的段落分段被安置到单个向量中。在此模型中，该向量及另外三个语境向量的拼接或者平均结果被用于预测第四个词。该段落向量表示了上下文缺失的信息，同时也充当了关于该段落话题的一份记忆。

模型中，上下文长度是固定的，并且是在段落上的滑动窗口上取到的样本。段落向量是被分享于所有由同一段落生成的上下文，而不是所有段落都实现共享。而词向量矩阵 W 是跨段落分享的。

在预测期间，模型需要实现一个推理步骤，给新段落计算段落向量。这个也是由梯度下降得到的，在这个过程中，模型的其它部分词向量 W 和 softma 权重都是固定住的。

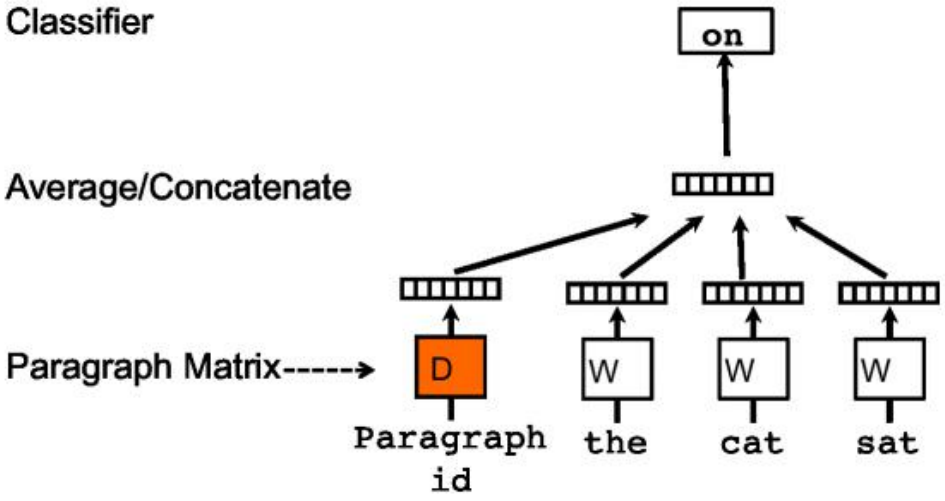


图 3 PV-DM 下段落向量的框架

3.3.1.2 PV-DBOW 模式

PV-DBOW，它与 PV-DM 相反。它忽略输入的上下文词，但强迫模型去预测从段落上随机样本出的词。实际上，这意味着在每次随机梯度下降循环里，我们样本出一个文本窗，然后从文本窗中样本出一个随机的词，之后在给定段落向量下去完成一个分类任务。分布词袋版本的段落向量如下图所示,除了概念简单外，此模型不需要储存过多的数据。我们只需要储存 softmax 权重,而不需要像 PV-DM 模型一样存 softmax 权重和词向量。这个模型类似于 word2vec 中的 skip-gram 模型。

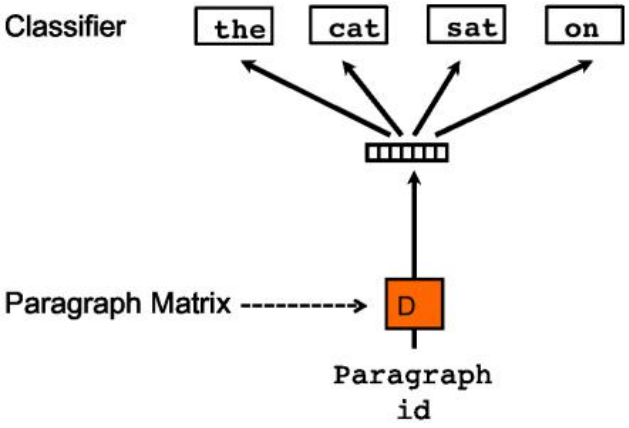


图 4 分布词袋版本的段落向量 PV-DBOW

3.3.2 分类模型的建立和预测

3.3.2.1 流程图

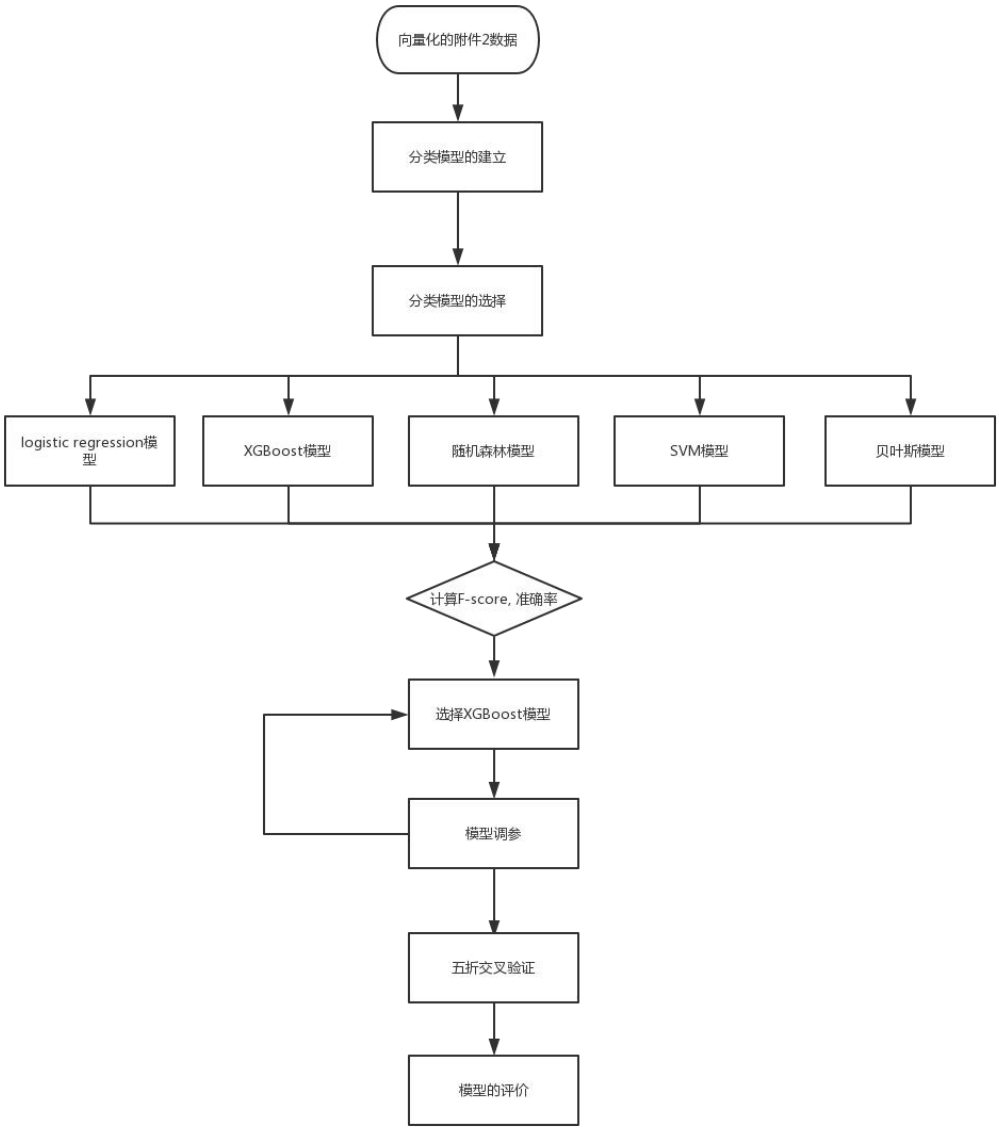


图 5 分类模型建立的流程图

3.3.2.2XGBoost 算法原理

特征提取完毕后，需要进行分类，本文采用的算法是 XGBoost。

极端梯度提升（extreme gradient boosting，XGBoost)是 boosting 算法的其中一种。Boosting 算法的主要思想：汇集多个弱分类器，从而形成一个强分类器。XGBoost 本质上是一种提升树模型，该算法主要汇集多个分类树模型，得

到一个很强的分类器。

XGBoost[3]算法思想：该算法通过对特征进行分类，从而生长出一颗树，并且不断的进行数的添加。每次添加一颗树，本质上是对一个新的函数进行学习，并用学习到的结果对上次预测的残差进行拟合。在对数据集进行训练之后，得到 k 棵树，对于每个样本的预测分数，需要根据该样本的特征，找到每棵树中对应的叶节点的分数，将每棵树中对应的叶节点 分数相加就是该样本的预测值。预测值函数为：

$$\hat{y} = \varphi(x_i) = \sum_{k=1}^{K} f_k(x_i) \\ where F = \{f(x) = w_{q(x)}\}(q : R^m \to T, w \in R^T) \tag{3-2}$$

其中，$w_{q(x)}$为叶子节点 q 的分数，f(x)为其中一棵回归树。 XGBoost 目标函数为：

$$Ob_j = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3-3}$$

其中，$\sum_{i=1}^{n} l(y_i, \hat{y}_i)$为训练时的损失函数，$\sum_{k=1}^{K} \Omega(f_k)$为数的复杂性。

目标函数包含两个部分：第一部分是模型的训练误差，第二部分是正则化项，正则化项是由 K 棵树的正则化项相加而来的。XGBoost 算法包含了多棵树，每棵树的复杂程度定义为：

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{3-4}$$

其中，T 表示叶子结点的个数，w 表示叶子节点的分数。γ可以控制叶子结点的个数，λ可以控制叶子节点的分数不会过大，防止过拟合。

新生成的树需要拟合上次预测的残差的，当生成 t 棵树后，预测分数可以写成：

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_i(x_i) \tag{3-5}$$

由此，目标函数可写为：

$$L^{(l)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(l-1)} + f_l(x_i)\right) + \Omega(f_l) \tag{3-6}$$

目的是能够找到一个 Z 能够使目标函数最小化。XGBoost 算法主要是利用目标函数在 Z=0 处 的泰勒二阶展开近似它，从而找到目标函数的近似 函数，目标函数近似为：

$$L^{(l)} \cong \sum_{i=1}^{n}[l(y_i,\hat{y}_i^{(l-1)}+g_i f_l(x_i))+\frac{1}{2}h_i f_l^2(x_i)]+\Omega(f_l) \tag{3-7}$$

其中，g_i 为一阶导数，h_i 为二阶导数。

$$g_i=\partial_{\hat{y}^{(l-1)}} l(y_i,\hat{y}_i^{(l-1)}+g_i f_l(x_i)), h_i=\partial^2_{\hat{y}^{(l-1)}} l(y_i,\hat{y}_i^{(l-1)}) \tag{3-8}$$

因此改写之后，最终可以将目标函数改写成关于叶子结点分数 w 的一个一元二次函数，可用顶点公式求解最优 w 和最优目标函数值因此，最优的 w 和目标函数公式为：

$$w_j^*=-\frac{G_j}{H_j+\lambda}, Ob_j=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T \tag{3-9}$$

3.3.2.3 模型的选取

文本分类[4]从根本上说是一个映射过程，所以评估文本分类系统的标志是映射的准确程度和映射的速度，这也就是分类器的准确度和运算速度。分类系统的运算速度主要取决于分类算法的时间复杂度和空间复杂度，而其准确度的衡量通常用到的是，准确率，查准率、查全率和 F1 值。

表 1 评价指标相关参数

	基准分类中与类别相关	基准分类中与类别无关
自动分类中与类别相关	A	C
自动分类中与类别无关	B	D

查准率和查全率定义如下：

$$P_i=\frac{A}{A+B} \tag{3-10}$$

$$R_i=\frac{A}{A+C} \tag{3-11}$$

查准率是所有分类的文本中分类正确的比率，而查全率是分类正确的占应该被分类正确的比率。通常情况下两者呈互补状态，单纯提高一个指标会导致另一个下降。所以，需要一个指标综合考虑这两个因素，这就是 F1 值：

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i} \tag{3-12}$$

准确率为:

$$\text{accuracy}\_score = \frac{A+D}{A+B+C+D} \tag{3-13}$$

计算每一个分类模型的 F-score 和准确率如下表：

表 2 分类模型的 F-score 和准确率

模型的种类/评价指标	F-score	准确率
随机森林模型	0.8863799072452417	0.8944082519001085
XGBoost 模型（调参后）	0.9229098805646037	0.921761968323841
logistic regression 模型	0.9052478564492102	0.9090662323561346
高斯贝叶斯模型	0.5840151866801477	0.522584147665581
SVM 模型	0.8905120808380308	0.895522541476655

3.3.2.4XGBoost 模型的调参

机器学习模型参数众多，为了提高模型的精度，同时提升模型的泛化能力，即在模型过拟合和欠拟合中寻求最优，调参过程[5]不可缺少。

对于本文所选 XGBoost 模型，我们把所有的参数分为 3 类：

通用参数：宏观函数控制。

Booster 参数：控制每一步的 booster(tree/regression)。

学习目标参数：控制训练目标的表现。

通过各参数之间的组合，能够对模型的精度提高和过拟合进行有效的控制，调参方案如下：

第一步：为了确定 boosting 参数，我们先给其它参数一个初始值。固定每个参数的初始值，再调整最优化。其中设置的初始值见调参结果详情表。

第二步：max_depth 和 min_weight 参数调优。我们先对这两个参数调优，是因为它们对最终结果有很大的影响。首先，我们先大范围地粗调参数，然后再小范围地微调。

第三步：gamma 参数调优。在已经调整好其它参数的基础上，我们可以进行 gamma 参数的调优了。Gamma 参数取值范围可以很大，我们在第一步调参时设置的初始 gamma 值就是比较合适的。也就是说，理想的 gamma 值为 0。

第四步：调整 subsample 和 colsample_bytree 参数。尝试不同的 subsample 和 colsample_bytree 参数。我们分两个阶段来进行这个步骤。这两个步骤都取 0.6,0.7,0.8,0.9 作为起始值。

第五步：正则化参数调优。应用正则化来降低过拟合。由于 gamma 函数提供了一种更加有效地降低过拟合的方法，大部分人很少会用到这个参数。但是我们在这里也可以尝试用一下这个参数。我会在这里调整’reg_alpha’参数。

第六步：降低学习速率。我们使用较低的学习速率，以及使用更多的决策树。在进行以上六步的调参过程后，调整汇总结果详情如下表所示：

表 3 参数调整汇总结果表

参数名称	参数简称	参数含义	初始值	调参结果
树的最大深度。	max_depth	树的最大深度。这个值是用来避 免过拟合的。max_depth 越大， 模型会学到更具体更局部的样 本,但深度越大可能过拟合。	3	6
树的个数	n_estimators	总共迭代的次数,即决策树的个数	100	200
特征占全部特征的比例	colsample_bytree	训练每棵树时,使用的特征占全部特征的比例。防止过拟合。	1	0.9
权重的 L2 正则化项	reg_lambda	控制 XGBoost 的正则化部分。	1	2
权重的 L1 正则化项	reg_alpha	可以应用在很高维度的情况下,使得算法的速度更快。	0	0.001

控制对于每棵树随机采样的比例	subsample	减小这个参数的值,算法会更加保守，避免过拟合。但是，如果这个值设置得过小,它可能会导致欠拟合。	1	0.6
学习率	learning_rate	控制每次迭代更新权重时的步长	0.1	0.005
决定最小叶子节点样本权重和。	min_child_weight	避免过拟合。当它的值较大时,可以避免模型学习到局部的特殊样本。但是如果这个值过高,会导致欠拟合。	1	2
惩罚项系数	gamma	在节点分裂时,只有分裂后损失函数的值下降了，才会分裂这个节点。Gamma 指定了节点分裂所需的最小损失函数下降值。这个参数的值越大，算法越保守。	0	0

3.3.2.5 XGBoost 模型的交叉验证

交叉验证的基本思想是把在某种意义下将原始数据(dataset)进行分组,一部分做为训练集(train set),另一部分做为验证集(validation set or test set),首先用训练集对分类器进行训练,再利用验证集来测试训练得到的模型(model),以此来做为评价分类器的性能指标。K 折交叉验证，初始采样分割成 K 个子样本，一个单独的子样本被保留作为验证模型的数据，其他 K-1 个样本用来训练。交叉验证重复 K 次，每个子样本验证一次，平均 K 次的结果或者使用其它结合方式，最终得到一个单一估测。这个方法的优势在于，同时重复运用随机产生的子样本进行训练和验证，每次的结果验证一次，10 折交叉验证是最常用的。本文采用 5 折交叉验证。

表 4 五折交叉验证结果

次数	1	2	3	4	5
F-socre	0.90678733	0.91493213	0.9239819	0.92488688	0.91131222

五折交叉验证结果如上表，可以看出模型的效果较好。

3.4 模型的评价和分析

3.4.1 模型的分类结果

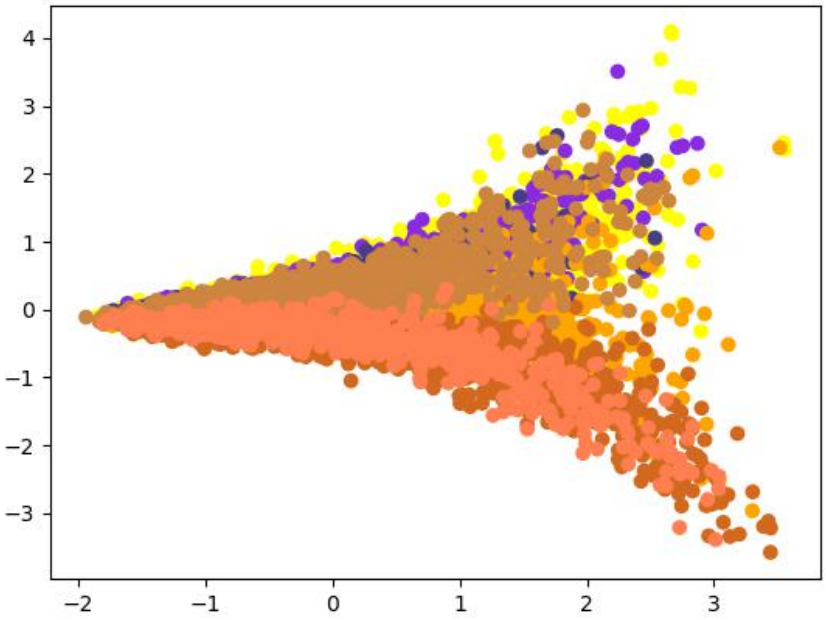


图 6 全部数据的分类

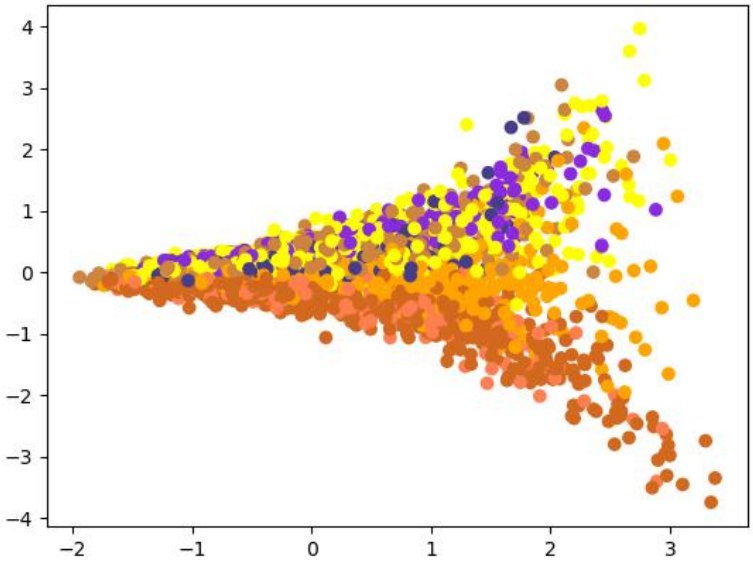


图 7 测试集数据的分类

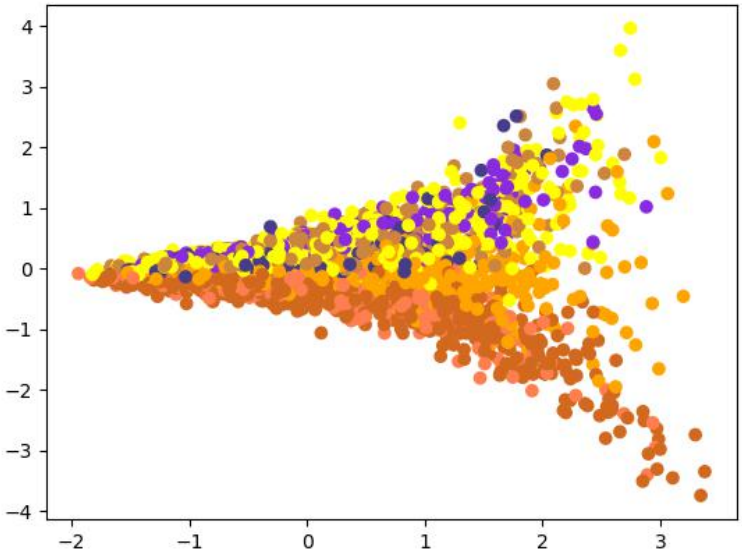


图 8 使用 XGBoost 模型分类

可以从图中看出使用 XGBoost 模型分类，有较好的结果。

3.4.2 模型优点

1） Doc2vec 模型克服了词袋模型的劣势，首先，特征向量能够继承到词的语义信息，例如”powerful”比“Paris”更接近“strong”。第二个好处是：至少在小的上下文基础上，它考虑了词顺序。并且由于高维表示会倾向于低的通用性，因此这段落向量模型好。

2） PV-DM 在多数任务上经常能取得较好的表现，但是如果再结合 PV-DBOW 的话，能够在很多的任务中入的始终如一的良好表现。因此本文使用它们的结合体

3） XGBoost 不仅能防止过拟合，还能降低计算；

4） XGBoost 控制了模型的复杂度，正则化项包含全部叶子节点的个数，每个叶子节点输出的 score 的 L2 模的平方和。从贝叶斯方差角度考虑，正则项降低了模型的方差，防止模型过拟合；

5） XGBoost 在每次迭代之后，为叶子结点分配学习速率，降低每棵树的权重，减少每棵树的影响，为后面提供更好的学习空间；

3.4.3 模型缺点

1） XGBoost 模型虽然利用预排序和近似算法可以降低寻找最佳分裂点的计算量，但在节点分裂过程中仍需要遍历数据集；

2) XGBoost 预排序过程的空间复杂度过高，不仅需要存储特征值，还需要存储特征对应样本的梯度统计值的索引，相当于消耗了两倍的内存.

4.问题二的分析与解决

4.1 数据的预处理

4.1.1 地名提取

根据附件 3，利用 jieba 模块分词的词性标注功能对词性进行筛选，具体方式为：对留言主题提取地名、英文字母、人名；对留言详情去掉停用词，提取地名，然后去掉重复出现的地名、字母等。将处理后留言的主题和留言内容合并写入文件：附件 3 的数据处理(地名)，部分数据如图 9。

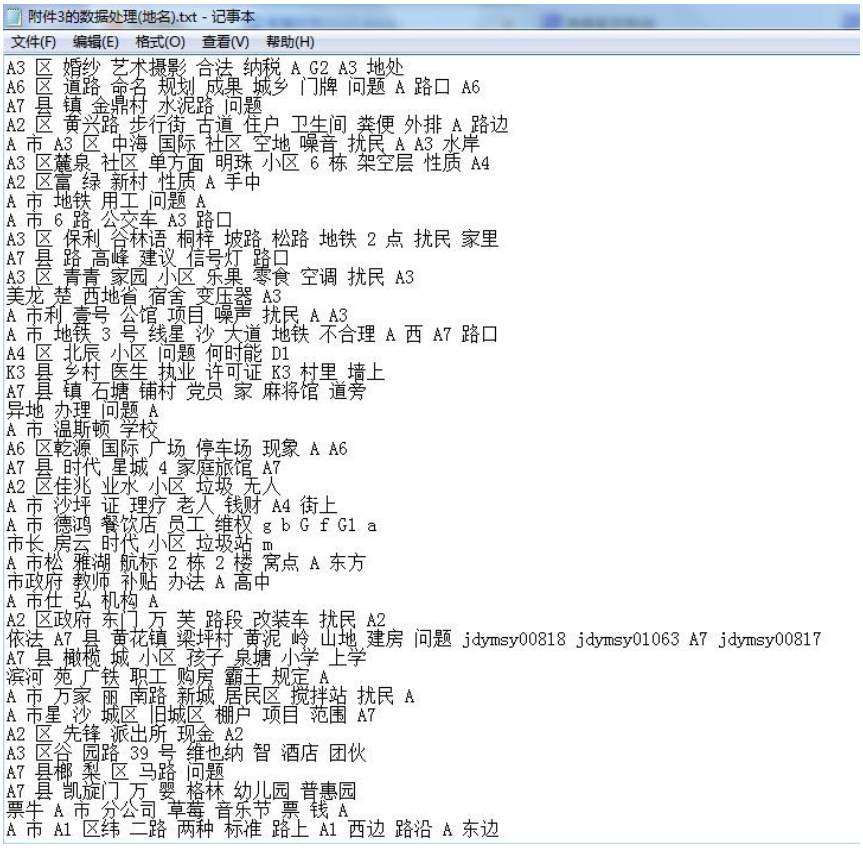

附件3的数据处理(地名).txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
A3 区 婚纱 艺术摄影 合法 纳税 A G2 A3 地处
A6 区 道路 命名 规划 成果 城乡 门牌 问题 A 路口 A6
A7 县 镇 金鼎村 水泥路 问题
A2 区 黄兴路 步行街 古道 住户 卫生间 粪便 外排 A 路边
A 市 A3 区 中海 国际 社区 空地 噪音 扰民 A A3 水岸
A3 区麓泉 社区 单方面 明珠 小区 6 栋 架空层 性质 A4
A2 区富 绿 新村 性质 A 手中
A 市 地铁 用工 问题 A
A 市 6 路 公交车 A3 路口
A3 区 保利 谷林语 桐梓 坡路 松路 地铁 2 点 扰民 家里
A7 县 路 高峰 建议 信号灯 路口
A3 区 青青 家园 小区 乐果 零食 空调 扰民 A3
美龙 楚 西地省 宿舍 变压器 A3
A 市利 壹号 公馆 项目 噪声 扰民 A A3
A 市 地铁 3 号 线星 沙 大道 地铁 不合理 A 西 A7 路口
A4 区 北辰 小区 问题 何时能 D1
K3 县 乡村 医生 执业 许可证 K3 村里 墙上
A7 县 镇 石塘 铺村 党员 家 麻将馆 道旁
异地 办理 问题 A
A 市 温斯顿 学校
A6 区乾源 国际 广场 停车场 现象 A A6
A7 县 时代 星城 4 家庭旅馆 A7
A2 区佳兆 业水 小区 垃圾 无人
A 市 沙坪 证 理疗 老人 钱财 A4 街上
A 市 德鸿 餐饮店 员工 维权 g b G f G1 a
市长 房云 时代 小区 垃圾站 m
A 市松 雅湖 航标 2 栋 2 楼 寓点 A 东方
市政府 教师 补贴 办法 A 高中
A 市仕 弘 机构 A
A2 区政府 东门 万 芙 路段 改装车 扰民 A2
依法 A7 县 黄花镇 梁坪村 黄泥 岭 山地 建房 问题 jdymsy00818 jdymsy01063 A7 jdymsy00817
A7 县 橄榄 城 小区 孩子 泉塘 小学 上学
滨河 苑 广铁 职工 购房 霸王 规定 A
A 市 万家 丽 南路 新城 居民区 搅拌站 扰民 A
A 市星 沙 城区 旧城区 棚户 项目 范围 A7
A2 区 先锋 派出所 现金 A2
A3 区谷 园路 39 号 维也纳 智 酒店 团伙
A7 县椰 梨 区 马路 问题
A7 县 凯旋门 万 婴 格林 幼儿园 普惠园
票牛 A 市 分公司 草莓 音乐节 票 钱 A
A 市 A1 区纬 二路 两种 标准 路上 A1 西边 路沿 A 东边

图 9 附件三的数据处理（地名）

4.1.2 全部信息的提取

根据附件 3，首先基于正则表达式进行判断，对全部留言的内容匹配汉字，对于全部留言的内容，提取地名、人名等信息，然后去掉停用词，对文本长度大于 200 的通过 jieba 分词进行关键字提取，使提取的信息不大于 200 个关键字；

对于全部留言的主题，匹配汉字，然后去掉停用词。

4.2 词向量模型的建立

文本信息提取后，由于任意两个词之间都是孤立的，根本无法表示出在语义层面上词语词之间的相关信息，所以需要建立神经网络语言模型。Word2vec 模型根据词语在语料库中的上下文信息，为每个词语训练一个相同维数的实向量。Word2vec 包含两种模式：连续词带模型（continuous bag-of-word model，CBOW）和 Skip-gram 模型。本文采用建立 CBOW 模型，快速高效的将词语表达成向量。

4.2.1 word2vec 模型的 CBOW 模式[6]原理

$v_i \in R^d$，词典中索引为 i 的词的背景词向量；

$u_i \in R^d$，词典中索引为 i 的词的中心词向量；

设中心词为w_C，背景词$w_{O_1}, \cdots, w_{o_{2m}}$，m 表示窗口大小。给定北京词生成中心词的条件概率：

$$P(w_c \mid w_{o_1},...,w_{o_{2m}}) = \frac{\exp(\frac{1}{2m} u_c^T (V_{o_1} + ... + V_{o_{2m}}))}{\Sigma_{j \in v} \exp(\frac{1}{2m} u_j^T (V_{o_1} + ... + V_{o_{2m}}))} \tag{4-1}$$

简写为：

$$P(w_c \mid w_o) = \frac{\exp(u_c^T v_o)}{\Sigma_{j \in v} \exp(u_j^T v_o)} \tag{4-2}$$

其中

$$w_o = w_{o_1},...,w_{o_{2m}}, v_o = V_{o_1} + ... + V_{o_{2m}} \tag{4-3}$$

给定一个 T 个字的字符串 s，根据语言模型有：

$$P(s) = P(w_1, w_2, ..., w_T) = \prod_{i=1}^{T} p(w_i \mid context(w_i)) = \prod_{i=1}^{T} p(w_i \mid w_{i-m},...,w_{i-1},w_{i+1},w_{i+m}) \tag{4-4}$$

我们的目标是让 P(s)最大化，这就变成最大似然估计问题，等价于最小化损失函数：

$$-\sum_{i=1}^{T} \log p(w_i \mid w_{i-m},...,w_{i-1},w_{i+1},w_{i+m}) \tag{4-5}$$

对背景向量v_{O_k}求导：

$$\frac{\partial L}{\partial v_{o_k}} = --\sum_{i=1}^{T}\frac{1}{2m}(u_i - \frac{\Sigma_{j\in v}\exp(u_j^T v_o)\cdot u_j}{\Sigma_{j\in v}\exp(u_j^T v_o)}) = -\sum_{i=1}^{T}\frac{1}{2m}(u_i - \sum_{j\in v} p(w_j \mid w_o)u_j) \tag{4-6}$$

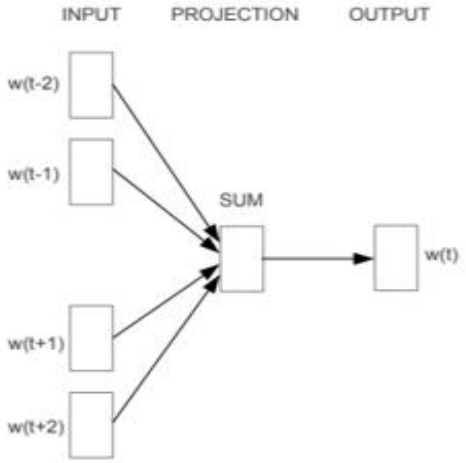


图 10 CBOW 模型的结构

4.2.2 建立对数据的 word2vec 模型

根据附件 3 的数据处理(地名)和，建立 word2vec 模型，分别为 word2vec。考虑到对留言地名的提取文本较短，对全部内容的提取文本较长：对于 word2vec1 选取词向量的维数为 200 部分词向量如图 11。

```
'农田': array([-0.02184939,  0.01756459, -0.00073701, -0.01379986,  0.028731
        0.01684092, -0.01425861, -0.00315147, -0.00232089, -0.01674174,
        0.0137148 ,  0.00334678, -0.00481856, -0.0040674 ,  0.01477192,
       -0.02786038,  0.01912196,  0.02457625,  0.00802181, -0.04421898,
       -0.04276991,  0.00323226,  0.00749699,  0.0122284 ,  0.00096224,
       -0.04047934,  0.01116333,  0.00121045, -0.01640369, -0.02591496,
        0.00764834,  0.03195404, -0.02958549, -0.00790827, -0.02804381,
       -0.01981271,  0.01230751, -0.01461409, -0.01072253,  0.02774361,
       -0.01594143,  0.00095171, -0.00704923,  0.00474442, -0.01382947,
       -0.02824643, -0.02064274,  0.00601545, -0.01381438,  0.02663404,
       -0.01300145, -0.00580628, -0.01259649, -0.02556014,  0.0105354 ,
       -0.01439515,  0.00792145, -0.01108157, -0.01393114,  0.04393332,
       -0.0080325 ,  0.02232183,  0.0155916 , -0.01073198, -0.00724717,
        0.00877314, -0.01006632,  0.04612237, -0.03519718, -0.03574158,
       -0.00053514, -0.02131504, -0.02131518, -0.05701074, -0.00404728,
        0.00568981,  0.00476507,  0.00024507,  0.00594525, -0.03201064,
```

图 11 部分词向量

```
    dtype=float32),
'湖西': array([-0.01743976,  0.01332924, -0.00164181, -0.00871713,  0.01837206,
        0.01264558, -0.01097266,  0.00037642, -0.00132515, -0.01362758,
        0.00946894,  0.00059178, -0.00187979, -0.00052406,  0.00989944,
       -0.02152917,  0.0130587 ,  0.01595931,  0.00642883, -0.03396081,
       -0.02973302,  0.0053263 ,  0.00545492,  0.00984502, -0.00219211,
       -0.02949544,  0.00914398,  0.00225904, -0.01347751, -0.01762814,
        0.00462456,  0.02361821, -0.01912217, -0.00649173, -0.02156757,
       -0.01343677,  0.00719605, -0.00939434, -0.0082774 ,  0.01819611,
       -0.00950316, -0.00204943, -0.00421982,  0.00350881, -0.01102456,
       -0.01770507, -0.01382847,  0.00259712, -0.00832022,  0.01797601,
       -0.00743253, -0.00109356, -0.00869507, -0.01540287,  0.00611176,
       -0.01106329,  0.00648648, -0.00795985, -0.01008559,  0.029974  ,
       -0.00490327,  0.01573534,  0.01319113, -0.00509978, -0.0053465 ,
        0.00787374, -0.00677605,  0.03430019, -0.02434982, -0.0258511 ,
       -0.00100781, -0.01735872, -0.01455666, -0.03776245, -0.00182007,
        0.00299174,  0.00133452,  0.00058029,  0.00593748, -0.02124994,
        0.0224007 , -0.02595714, -0.01757234,  0.0146307 , -0.02294468,
[5]:
```

图 12 部分词向量

4.3 聚类地名

4.3.1 聚类模型的建立

建立 word2vec1 词向量模型后，用 Mini Batch K-Means 模型进行聚类。K-means 算法是一种较典型的基于样本间相似性度量的逐点修改迭代的动态聚类算法，属于机器学习中方法中的非监督学习。此算法依据设定的参数 k，把 n 个对象划分到 k 个簇中，每个簇中心为簇中对象的平均值，使得每一个对象与所属簇的中心具有较高的相似度，而与不同簇的中心相似度较低。考虑到经典的 K-means 算法对大规模数据集处理效率低，本文所使用的 Mini Batch K-Means 算法是 K-Means 算法的一种优化变种，采用小规模的数据子集（每次训练使用的数据集是在训练算法的时候随机抽取的数据子集）减少计算时间，同时试图优化目标函数；Mini Batch K-Means 算法可以减少 K-Means 算法的收敛时间，而且产生的结果效果只是略差于标准 K-Means 算法。Mini Batch K-Means 算法步骤如下：

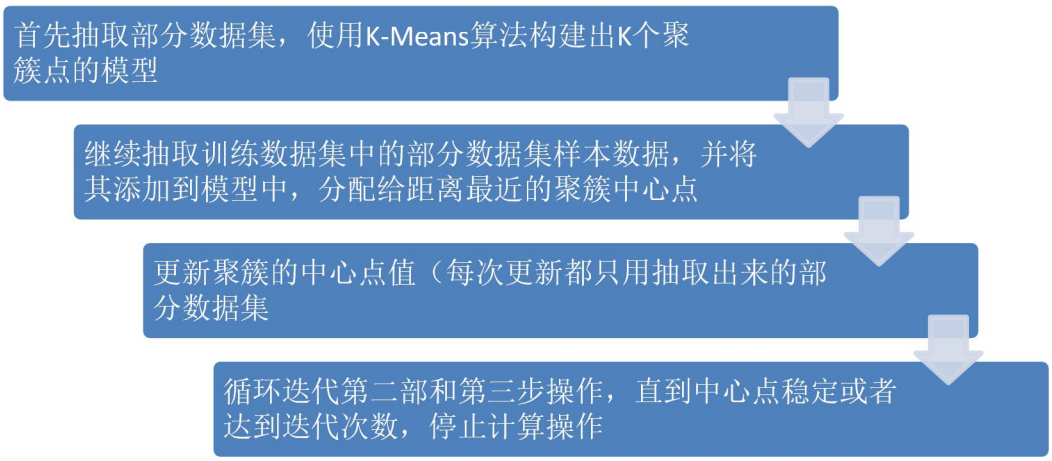


图 13　Mini Batch K-Means 算法步骤

对于聚类类别个数的选取，用 CH 分数（Calinski Harabasz Score）进行选取。CH 分数[7]是通过评估类之间方差和类内方差来计算得分。$s = \frac{SS_B}{k-1} \Big/ \frac{SS_w}{N-k}$，其中 k 代表聚类类别数，N 代表全部数据数目。　SS_B是类间方差，SS_w　是类内方差。

SS_B的计算方式是：

$$SS_B = \mathrm{tr}(B_k) \tag{4-7}$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E c_q - c_E)^T \tag{4-8}$$

trace 只考虑了矩阵对角上的元素，即类中所有数据点到类的欧几里得距离；

SS_w的计算方式是：

$$SS_W = \mathrm{tr}(W_k) \tag{4-9}$$

$$W_k = \sum_{q=1}^{k} \sum_{x_q} (x - c_q)(x - c_q)^T \tag{4-10}$$

其中是类中所有数据的集合，是类的质点，是所有数据的中心点，是类数据点的总数。评价聚类的指标有很多如轮廓系数，CH 分数等。

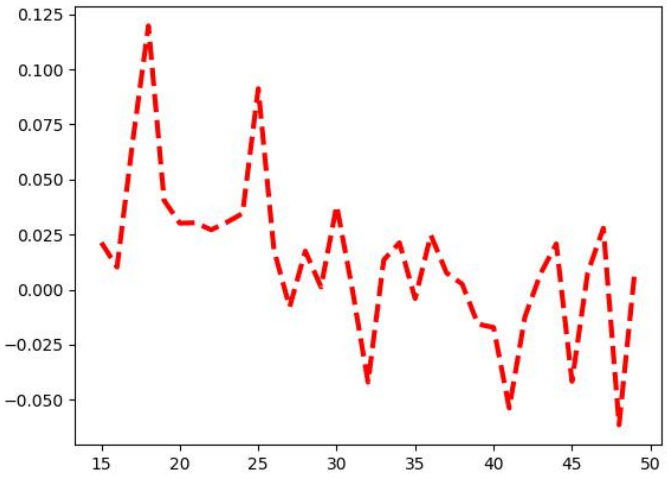


图 14 轮廓系数-聚类的个数

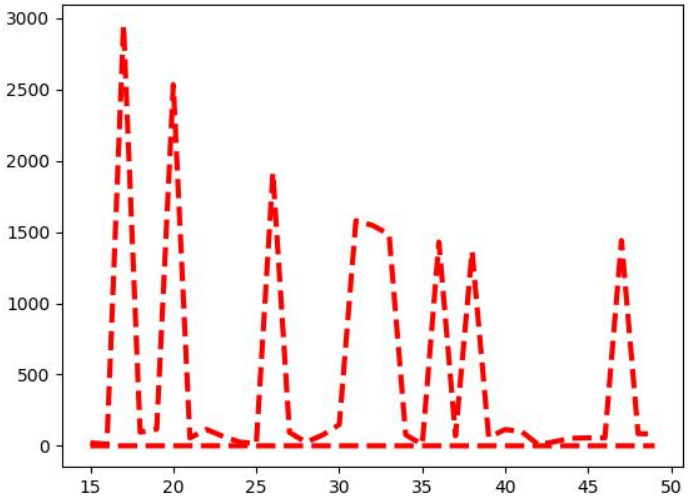


图 15 CH 分数—聚类的个数

本文选取 CH 分数最大的类，获得聚类结果。

4.3.2 聚类结果的处理

使用聚类结果对热点问题地区进行总结，数据归并。

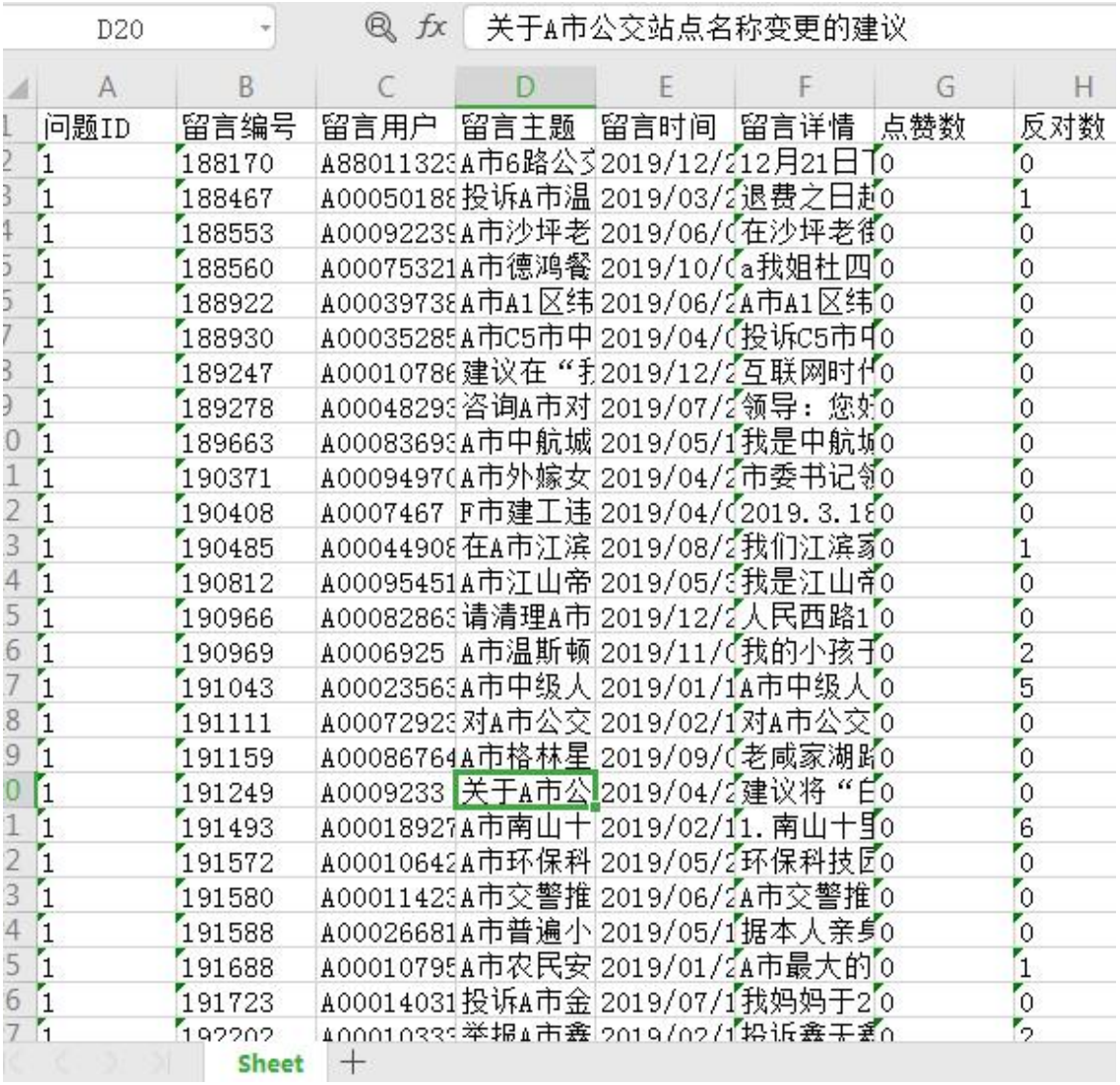
4.3.2.1 地名的等级指数

对于上文获得的每一聚类结果，计算类的大小以及类里面所有元素的反对数和赞成数的加权和，即类中元素的个数+0.5*所有的赞成数+0.5*反对数，由此得到地名的等级指数。然后将所有的类的等级指数进行排序，取前五个地名。

4.3.2.2 热点问题地区等级表

根据附件 3，将每一个类的留言内容按照留言用户的 ID 进行排序写入文件得到表：热点问题地区等级表。

如图部分数据：



D20　　关于A市公交站点名称变更的建议

	A	B	C	D	E	F	G	H
1	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
2	1	188170	A88011323	A市6路公交	2019/12/2	12月21日下	0	0
3	1	188467	A00050188	投诉A市温	2019/03/2	退费之日起	0	1
4	1	188553	A00092239	A市沙坪老	2019/06/0	在沙坪老很	0	0
5	1	188560	A00075321	A市德鸿餐	2019/10/0	a我姐杜四	0	0
6	1	188922	A00039738	A市A1区纬	2019/06/2	A市A1区纬	0	0
7	1	188930	A00035285	A市C5市中	2019/04/0	投诉C5市中	0	0
8	1	189247	A00010786	建议在“我	2019/12/2	互联网时代	0	0
9	1	189278	A00048293	咨询A市对	2019/07/2	领导：您好	0	0
10	1	189663	A00083693	A市中航城	2019/05/1	我是中航城	0	0
11	1	190371	A0009497(	A市外嫁女	2019/04/2	市委书记领	0	0
12	1	190408	A0007467	F市建工违	2019/04/0	2019.3.18	0	0
13	1	190485	A00044908	在A市江滨	2019/08/2	我们江滨豪	0	1
14	1	190812	A00095451	A市江山帝	2019/05/3	我是江山帝	0	0
15	1	190966	A00082863	请清理A市	2019/12/2	人民西路1	0	0
16	1	190969	A0006925	A市温斯顿	2019/11/0	我的小孩于	0	2
17	1	191043	A00023563	A市中级人	2019/01/1	A市中级人	0	5
18	1	191111	A00072923	对A市公交	2019/02/1	对A市公交	0	0
19	1	191159	A00086764	A市格林星	2019/09/0	老咸家湖路	0	0
20	1	191249	A0009233	关于A市公	2019/04/2	建议将“白	0	0
21	1	191493	A00018927	A市南山十	2019/02/11	.南山十里	0	6
22	1	191572	A00010642	A市环保科	2019/05/2	环保科技园	0	0
23	1	191580	A00011423	A市交警推	2019/06/2	A市交警推	0	0
24	1	191588	A00026681	A市普遍小	2019/05/1	据本人亲身	0	0
25	1	191688	A00010795	A市农民安	2019/01/2	A市最大的	0	1
26	1	191723	A00014031	投诉A市金	2019/07/1	我妈妈于2	0	0
27	1	192202	A0001033(	举报A市鑫	2019/02/1	投诉鑫天鑫	0	2

Sheet

图 16 热点问题地区等级表

4.4 聚类热点问题

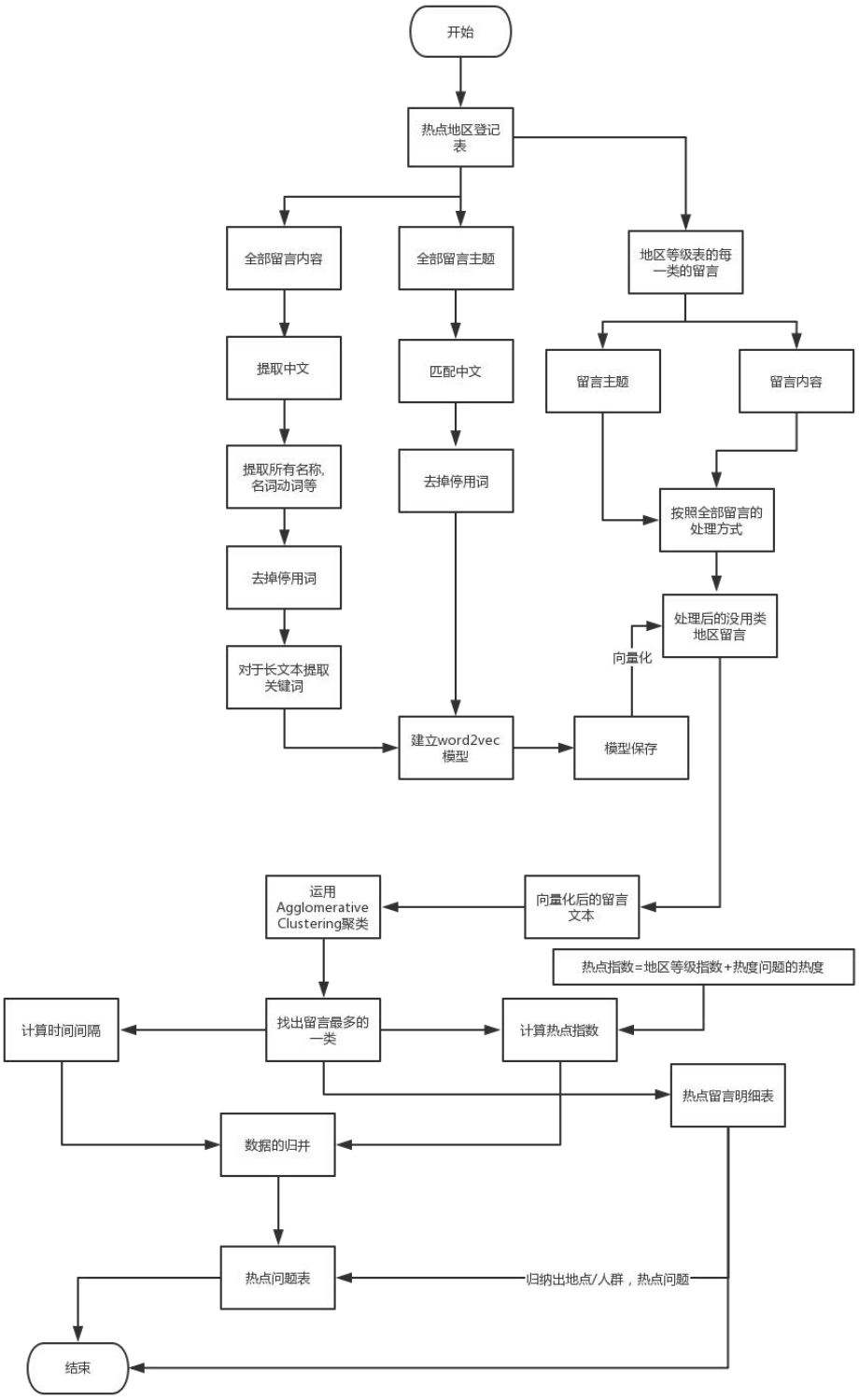


图 17 聚焦热点问题流程图

4.4.1 热点问题聚类模型的建立

对于附件 3 的数据，提取全部的信息然后建立词向量模型 word2vec 记为 word2vec2，对于 word2vec2 选取词向量的维数为 400 ，最大的窗口为 8。

利用生成的 word2vec2 模型对热点问题地区等级表中的留言内容和留言主题进行向量化，然后对每一类地区的问题进行聚类。本文用 Agglomerative Clustering（层次聚类）模型进行聚类。

层次聚类[8]是一种自下而上的算法，首先将每个样本都视为一个簇，然后开始按一定规则，将相似度高的簇进行合并，最后所有样本都形成一个簇或达到某一个条件时，算法结束。确定簇与簇之间相似度是该算法的要点，而这里的相似度是由簇间距离来确定的，簇间距离短(小)的相似度高，簇间距离长(大)的相似度低。

簇间距离：

MIN，又称为单链，即取两个簇之间距离最近的两个点的距离（两个簇间的最小边长）。

$$d(u,v)=\min(dist(u[i],v[j])) \tag{4-11}$$

MAX，又称为“全链”，即取两个簇之间距离最远的两个点的距离（两个簇间的最大边长）。

$$d(u,v)=\max(dist(u[i],v[j])) \tag{4-12}$$

组平均，这里简单介绍两种。

① 非加权组平均（UPGMA）

$$d(u,v)=\sum_{ij}\left(\frac{dist(u[i],v[j])}{(|u|*|v|)}\right) \tag{4-13}$$

其中|u|和|v|分别表示簇 u 和簇 v 的元素个数。

② 加权组平均（WPGMA）

$$d(u,v)=\frac{dist(s,v)+dist(t,v)}{2} \tag{4-14}$$

其中 u 是由 s 和 t 合并形成的。

质心距离，指的是簇质心与簇质心之间的距离。这里简单描述两种质心距离。

①UPGMC，即

$$dist(u,v)=\| c_u-c_v \|_2 \tag{4-15}$$

其中c_u和c_v分别为簇 u 和簇 v 的聚类中心。

②WPGMC，即新簇质心取合并的两个簇的质心均值(取质心的方法不同)。

使用质心距离计算簇间相似度的方法，质心数据是会随着簇合并发生变化的，质心变化后，质心数据就可能不存在于样本数据里。因而，在簇不断合并的过程中，其他方法计算簇间距离，簇间距离是递增的，而使用质心距离计算，簇间距离不一定递增。

沃德方差最小化：

$$d(u,v)=\sqrt{\frac{|s|+|v|}{T}dist(s,v)^2+\frac{|t|+|v|}{T}dist(t,v)^2-\frac{|v|}{T}d(s,t)^2} \tag{4-16}$$

其中 u 是 s 和 t 合并形成的，而 T=|s|+|t|+|v|，|*|等于簇*的元素个数

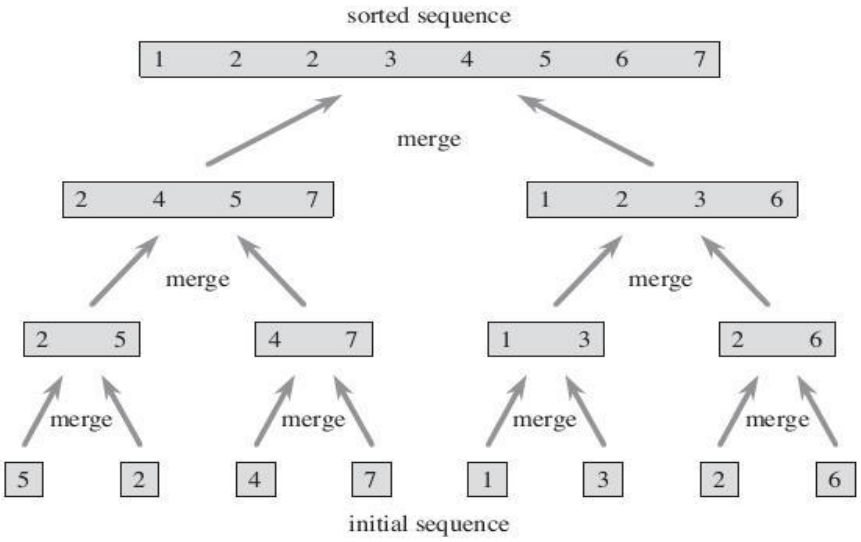


图 18　Agglomerative Clustering 的原理图

将向量化后的热点问题地区等级表中的留言内容和留言主题分了 30 簇，选取其中元素个数最多的一簇作为这个地区的热点问题。

4.4.2 分析排名

4.4.2.1 热点问题明细表的建立

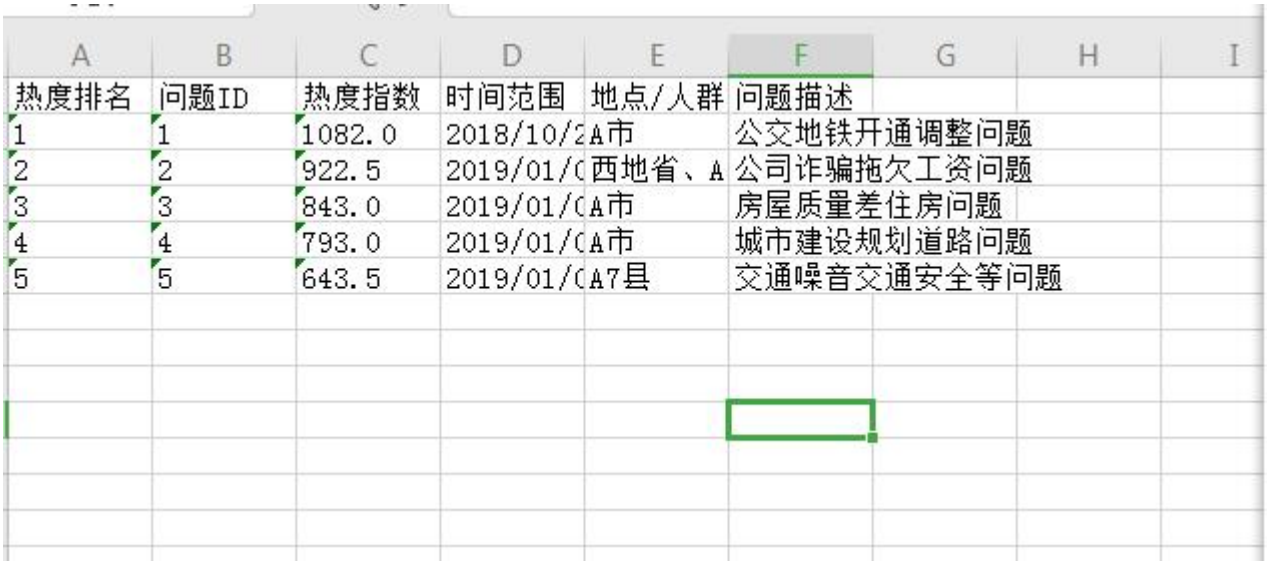
将按照地名的等级指数取的前 5 个地区的热点问题作为 5 类热点问题，对数据进行归并，就能得到热点问题明细表。部分结果如图：

5	194554	A00075486	A7县三一大	2019/11/0	三一大道西	0	0
5	197479	A00063376	直通A8县高	2019/03/0	直通A8县高	0	0
5	197527	A00010625	给A7县松开	2019/03/2	尊敬的领导	2	9
5	197964	A00052075	请解决A7县	2019/02/2	A7县黄星大	0	3
5	199330	A00092277	A7县海伦春	2019/02/1	A7县黄星大	0	0
5	205332	A00028151	反映A7县身	2019/11/2	尊敬的领导	0	4
5	207651	A00057795	A7县楚龙得	2019/11/2	尊敬的沈书	0	11
5	209828	A00013276	恳请A7县城	2019/12/3	我是泉塘的	0	0
5	218267	A00074105	请将A7县公	2019/02/2	尊敬的县交	1	26
5	219432	A00083527	A7县城区考	2019/05/0	A7县城城区	0	1
5	224045	A0006300	A7县楚龙得	2019/01/0	位于楚龙得	0	1
5	230954	A00080322	请求解决A	2019/07/0	导致漓楚路	0	1
5	248113	A00080330	对A7县东六	2019/04/2	1、东六路	0	2
5	250979	A00042344	A7县漓楚路	2019/12/0	因东六路未	0	2
5	259422	A00053043	建议在A7县	2019/03/2	整个黄星大	0	6
5	259821	A00076669	A7县宁华路	2019/09/2	尊敬的张县	0	1
5	260126	A00080323	希望A7县8	2019/06/2	809路公交	0	2
5	261258	A00042422	A7县春华钟	2019/02/2	尊敬的各位	0	0
5	263100	A00087136	A7县A5区六	2019/12/1	A5区大道老	0	0
5	272588	A00042422	A7县春华钟	2019/02/2	尊敬的各位	0	0
5	273328	A00080330	建议取消A	2019/05/2	尊敬的领导	0	4
5	277066	A00075486	请问A7县时	2019/11/0	每天经三一	0	2
5	277554	A00080323	A7县漓楚东	2019/06/2	漓楚路作为	0	1
5	284907	A0009233	A7县天华路	2019/09/2	天华路凉城	0	1
5	287507	A00044417	反映A7县河	2019/05/0	尊敬的领导	0	3

图 19 热点问题明细表的建立

4.4.2.2 热点问题表的建立

对于每一个分类好的每个地区，人为概括其热点问题，并提取问题描述中的人名、地名、留言的主题，作为热点问题表中的人名、地名、问题详情。



	A	B	C	D	E	F	G	H	I
1	热度排名	问题ID	热度指数	时间范围	地点/人群	问题描述			
2	1	1	1082.0	2018/10/2	A市	公交地铁开通调整问题			
3	2	2	922.5	2019/01/0	西地省、A	公司诈骗拖欠工资问题			
4	3	3	843.0	2019/01/0	A市	房屋质量差住房问题			
5	4	4	793.0	2019/01/0	A市	城市建设规划道路问题			
6	5	5	643.5	2019/01/0	A7县	交通噪音交通安全等问题			

图 20 热点问题明细表

4.4.2.3 热点指数的计算

热点指数=地区的等级指数+热度问题的热度

- 地区的等级指数

  对于聚类结果处理得到的前五个地名，规定：地区 1 的等级指数为 1000；地区 2 的等级指数为 900；地区 3 的等级指数为 800；地区 4 的等级指数为 700；地区 5 的等级指数为 600；

- 热点问题的热度

  规定为类中所有热点的点赞数*0.5++反对数*0.5。

5.问题三的分析与解决

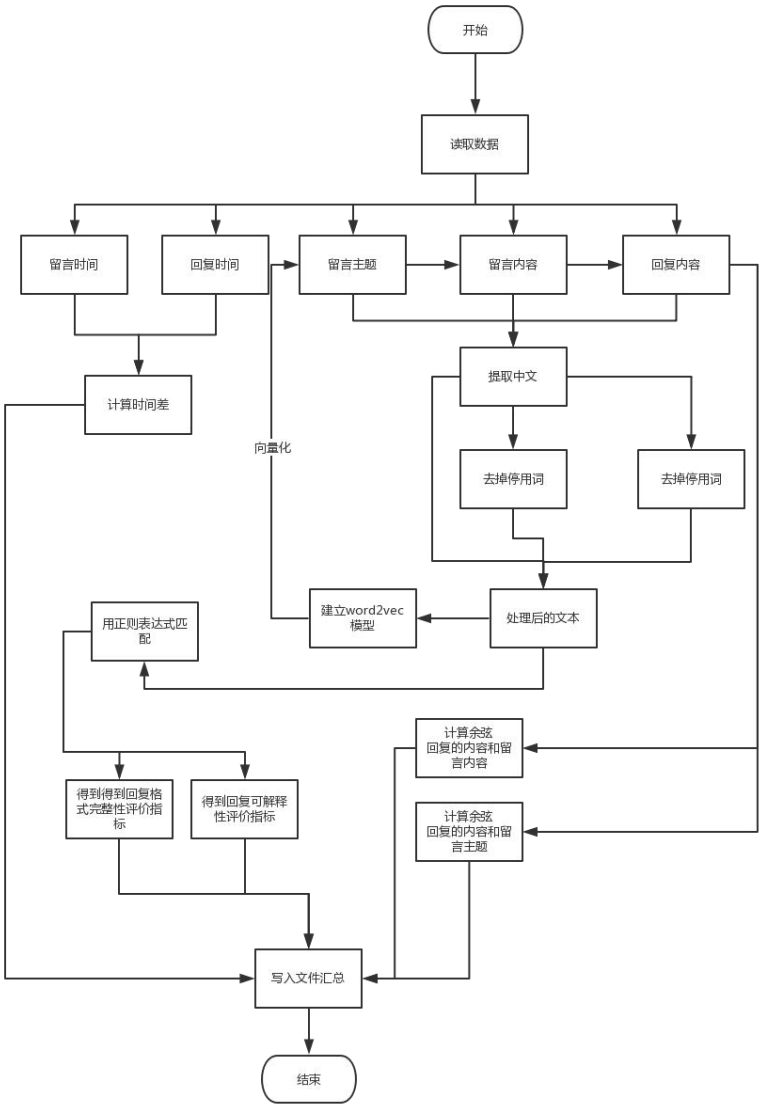


图 21 问题三的流程图

5.1 数据的预处理

对附件四留言的主题，留言的内容，回复的内容提取中文，并去掉停用词。

5.2 词向量模型的建立

将上述的预料建立词向量，建立 word2vec 模型，模型的词向量为 100 维，最大的窗口为 14。

部分词向量如图：

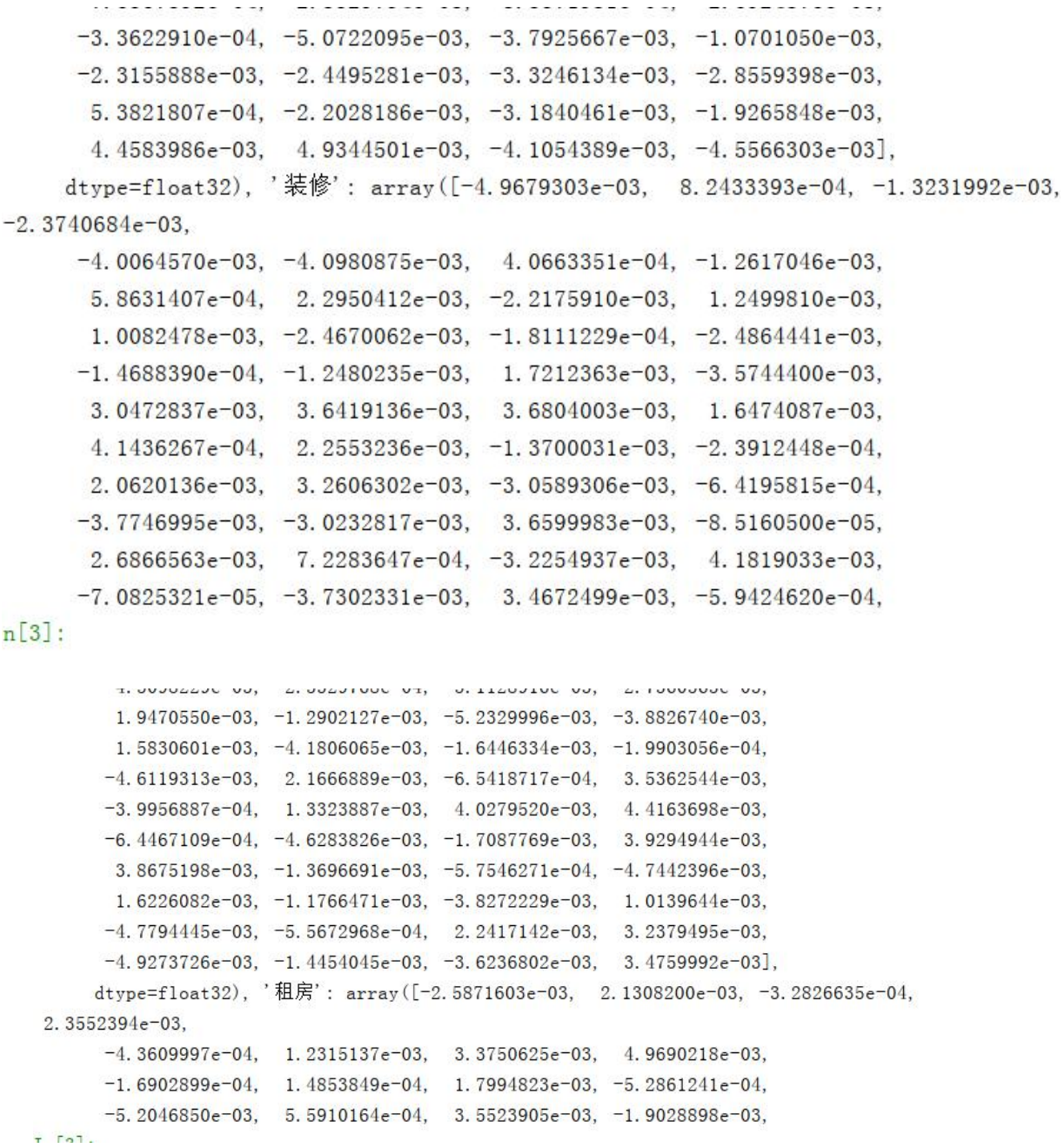


图 22 词向量的建立

5.3 答复意见的评价

针对实际生活中网络留言答复意见存在办理不及时，答非所问避重就轻，回复敷衍了事的问题。本文将分别从从答复的相关性、完整性、可解释性，时效性，规范性对答复意见给出评价方案。

5.3.1 答复的相关性

5.3.1.1 余弦相似度

余弦距离，也称为余弦相似度，是用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小的度量。

余弦值越接近 1，就表明夹角越接近 0 度，也就是两个向量越相似，这就叫"余弦相似性"。

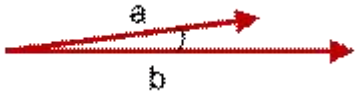


图 23 余弦相似度

上图两个向量 a,b 的夹角很小可以说 a 向量和 b 向量有很高的的相似性，极端情况下，a 和 b 向量完全重合。

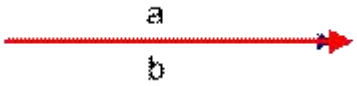


图 23 完全重合的向量

假定 a 和 b 是两个 n 维向量，则 a 与 b 的夹角的余弦等于：

$$\cos(\theta)=\frac{\sum_{i-1}^{n}(x_i\times y_i)}{\sqrt{\sum_{i-1}^{n}(x_i)^2}\times\sqrt{\sum_{i-1}^{n}(y_i)^2}}=\frac{a\bullet b}{\|a\|\times\|b\|} \tag{5-1}$$

余弦值越接近 1，就表明夹角越接近 0 度，也就是两个向量越相似，夹角等于 0，即两个向量相等，这就叫"余弦相似性"。

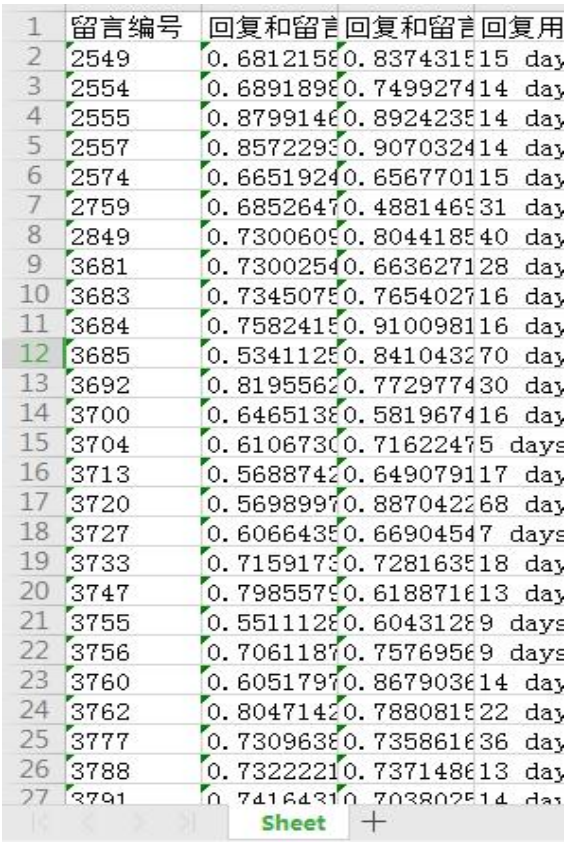
5.3.1.2 计算文本的相似度

我们认为与留言主题内容相关性高的答复，更大可能是高质量的答复。

将留言的主题，留言的内容，回复的内容的各个词向量相加，归一化，建立文本向量。

利用向量的余弦，计算留言的主题和回复的内容的相似度。

计算留言的内容和回复的内容的相似度。



	留言编号	回复和留言	回复和留言	回复用
2	2549	0.6812158	0.8374315	15 day
3	2554	0.6891898	0.7499274	14 day
4	2555	0.8799146	0.8924235	14 day
5	2557	0.8572293	0.9070324	14 day
6	2574	0.6651924	0.6567701	15 day
7	2759	0.6852647	0.4881469	31 day
8	2849	0.7300609	0.8044185	40 day
9	3681	0.7300254	0.6636271	28 day
10	3683	0.7345075	0.7654027	16 day
11	3684	0.7582415	0.9100981	16 day
12	3685	0.5341125	0.8410432	70 day
13	3692	0.8195562	0.7729774	30 day
14	3700	0.6465138	0.5819674	16 day
15	3704	0.6106730	0.7162247	5 days
16	3713	0.5688742	0.6490791	17 day
17	3720	0.5698997	0.8870422	68 day
18	3727	0.6066435	0.6690454	7 days
19	3733	0.7159173	0.7281635	18 day
20	3747	0.7985575	0.6188716	13 day
21	3755	0.5511128	0.6043128	9 days
22	3756	0.7061187	0.7576956	9 days
23	3760	0.6051797	0.8679036	14 day
24	3762	0.8047142	0.7880815	22 day
25	3777	0.7309638	0.7358616	36 day
26	3788	0.7322221	0.7371486	13 day
27	3791	0.7416431	0.7038025	14 day

Sheet

图 24 文本的相似度

5.3.2 答复的完整性

我们认为回复内容越短，答复不完整的概率就越大。因此我们将所有的留言回复的长度计算其分位数得到 10%分位数 ，20%分位数，30%分位数，70%分位数。然后按照下表得到对应的等级。

表 5 答复完整性表

留言回复长度	完整性	等级
0-10	完整性很差	D
10-30	完整性差	C
30-70	完整性一般	B
70-100	完整性好	A

5.3.3 答复的时效性

将回复内容的时间与留言的时间相减。时间差反映了答复是否及时，及时高效的答复对网络反映问题的十分重要，长时间未处理的问题会造成很大的负面影响。

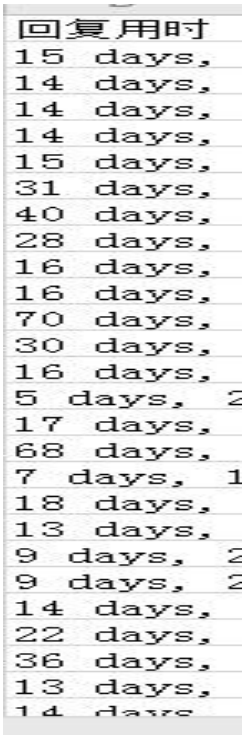


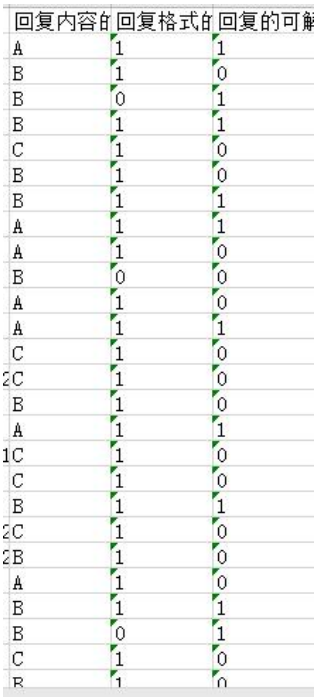
图 25 留言回复时效性

5.3.4 答复的规范性

我们认为答复网民留言时，要规范格式，注意用词，严格遵守相关规定。因此答复的规范性也作为一个留言回复的一个评价指标。我们通过定一个标准的留言格式，用正则表达式匹配，用来评价回复的格式规范性。1 表示格式规范，0 表示格式不规范。

5.3.5 答复的可解释性

我们希望答复网民留言时，特殊问题需要有相关法律条例支撑，因此可解释性也作为留言回复优劣的评价指标。我们利用正则表达式匹配是否提及了相关的条例，法律。1 表示具有可解释性，0 表示不具有可解释性。



回复内容的	回复格式的	回复的可解
A	1	1
B	1	0
B	0	1
B	1	1
C	1	0
B	1	0
B	1	1
A	1	1
A	1	0
B	0	0
A	1	0
A	1	1
C	1	0
2C	1	0
B	1	0
A	1	1
1C	1	0
C	1	0
B	1	1
2C	1	0
2B	1	0
A	1	0
B	1	1
B	0	1
C	1	0
B	1	0

图 26 答复的规范性与可解释性

6.不足与展望

对于问题一：不足:由于数据集较大的原因可能程序运行的速度较慢，然后对于模型的调参没有找到最好的参数，找到的是较好的参数，分类的准确率和 F-score 都还比较高。

展望：从本文的结果可以看出，在使用 Doc2vec 混合模型的情况下，各种机器学习的模型分类的准确率，F-score，都比较好。可能是由于 Doc2vec 混合模型的算法，在处理文本分类中有较好的效果，文本向量的特征性较强，比较适合做文本多分类，希望可以推广这个模型。

对于问题二：不足:在得到的地区等级表中存在一点噪声，另外存在地区的包含关系，比如在生成的地区等级表，A 市中的 A3 区可能也被分类在 A 市里面，而不是所有 A3 区是单独的一类，可能会造成部分数据误分的情况，影响了聚类效果，另一方面，也因为存在地区包含关系，对于被包含的地区，应该赋与较大的权重，而包含的地区赋与较小的权重，以平衡不同区域的差异，但是没有找到较好的方式去找到权重，本文运用了一个线性函数作为对不同地区等级系数。另外由于没有足够大的预料训练模型 word2vec 模型，可能会导致聚类的效果不好。

展望：相比与问题一的 Doc2vec 的混合模型，问题二运用了 word2vec 的模型，因为经过试验，用 word2vec 的模型将一个个的词转换为向量后，然后用整个文本的平均词向量作为文本向量，在文本的聚类方面比 Doc2vec 的模型更好一

点，另外本文是从地区出发，先对文本进行聚类，然后在地区里面再聚类得到一定人群的热点问题，也可以考虑先从人群出发，先按照人群聚类，然后再按照地区聚类，在再聚类出热点的问题，两种方式可能结果差别不是很大，但是相比人群，地区可能更具有特征性，比较容易聚类出满意的结果。

对于问题三：不足：答复意见质量的评价方案不够完善，仅能从各个角度进行评价，而对综合各项指标的回复质量未能给出清晰评价。针对于答复的完整性，本文没有考虑较短字数完整回复的情况，默认为不够完整。针对部分答复网络反映问题相互推诿，不能积极主动地与相关单位协调处理的问题没有解决。

展望：对于回复评论质量评估工作仍然是基于传统的文本分析方法，先做数据预处理，找出特征在进行相应的相关性分析，正则表达式匹配等工作，只能局限于特定领域，相关方法在其它领域的适应性还有待改进。但是本文的质量评估方案从各个方面对于网络回复评论的评价能够提供十分有力的参考。

参考文献

[1]李峰,柯伟扬,盛磊, 等.Doc2vec 在政策文本分类中的应用研究[J].软件,2019,40(8):76-78. DOI:10.3969/j.issn.1003-6970.2019.08.018.

[2]Quoc Le, Tomas Mikolov.Distributed Representations of Sentences and Documents[C].2014International Conference on Machine Learning

[3]段立,徐鸿宇,王懿, 等.基于 word2vec 和 XGBoost 相结合的国网 95598 客服投诉工单分类[J].电力大数据,2019,22(12):50-57.

[4]朱磊.基于 word2vec 词向量的文本分类研究[D].重庆:西南大学,2017.

[5]Complete-guide-parameter-tuning-xgboost-with-codes-python.[DB/OL].https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

[6]Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. A neural prob-abilistic language model. Journal of Machine Learning Research (JMLR), 3:1137-1155,2003

[7]Ethen Liu.Calinski-Harabasz Index and Boostrap Evaluation with ClusteringMethods.[DB/OL].https://ethen8181.github.io/machine-learning/clustering_old/clustering/clustering.html

[8] 余东瑾.基于文本分类与主题模型的用户偏好分析[D].山东:青岛科技大学,2017.