

“智慧政务”中的文本挖掘与综合分析

“智慧政务”中的留言信息的分析与挖掘

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。因此,运用自然语言处理技术和文本挖掘的方法对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研究有着重大的意义。

对于问题 1,通过 pandas 中的 drop_duplicates()函数将群众留言记录表进行去重,得到不重复的群众留言信息。使用正则表达式将数据进行脱敏、去空的处理,再利用中文分词组件 jieba 对群众留言详情进行中文分词,将数据切分为训练集和测试集,并通过 TF-IDF 权重策略将文本向量化表示,最后以训练集的 TF-IDF 权值向量构建出文本分类模型,以测试集的 TF-IDF 权值向量对构建好的模型进行模型评价。以及将数据以数据可视化——词云的形式直观的呈现出来。

对于问题 2,通过 pandas 中的 drop_duplicates()函数将群众留言记录表进行去重,得到不重复的群众留言信息。通过 HanLP 中的命名实体识别功能,识别出留言详情中的地点和群体。通过 synonyms 中的相似度比较功能,将问题进行归类并计算同类问题数目,再结合问题的点赞数和反对数将热点问题的热度指标进行量化,以及提取同类问题的留言时间,使用自定义比较时间函数计算得到最近的时间和最远的时间,以便之后的热点问题表中的时间范围确定,并根据同类问题将相应热点问题对应的留言信息写入热点问题明细表中。

对于问题 3,自定义出对于相关部门对留言的答复意见的质量的一套评价方案,从答复意见与问题的是否相关、答复意见是否详细完整、答复意见是否使用了书面语、答复意见中是否使用正确的格式:对于网友的称呼是否有“尊敬”或者与“尊敬”近似的形容词,答复意见中是否有“你好”或者与“你好”近似的文明用语。对答复内容和留言详情进行关键词提取,并对其二者关键词进行比对,则能量化出答复意见的相关性指标。

关键词: 去重 中文分词 TF-IDF 算法 文本向量化 命名实体识别 关键词提取

Analysis and mine of message information in "smart government"

Abstract

In recent years, with the online questioning platforms such as WeChat, Weibo, mayor's mailbox, and sunshine hotline, etc., it has gradually become an important channel for the government to understand public opinion, gather people's wisdom, and gather people's popularity. The work of the relevant departments, which mainly relied on manual work to divide the message and organize hotspots, brought great challenges. At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of a smart government system based on natural language processing technology has become a new trend of social governance innovation and development, which has a great impact on improving the government's management level and governance efficiency. Promote the role. Therefore, the use of natural language processing technology and text mining methods has a great significance for the study of the public's question and answer message records and the relevant departments' responses to some of the people's comments.

For question 1, the mass message record table is deduplicated by the `drop_duplicates()` function in pandas to obtain non-duplicate mass message information. Use regular expressions to desensitize and empty the data, and then use the Chinese word segmentation component jieba to perform Chinese word segmentation on the details of the people's message, divide the data into training and test sets, and use the TF-IDF weight strategy to convert the text vector Representation, and finally build a text classification model with the TF-IDF weight vector of the training set, and evaluate the constructed model with the TF-IDF weight vector of the test set. And visualize the data as data—intuitively presented in the form of word clouds.

For question 2, the mass message record table is deduplicated by the `drop_duplicates()` function in pandas to obtain non-duplicate mass message information. Through the named entity recognition function in HanLP, identify the location and group in the message details. Through the similarity comparison function

in synonyms, classify the problems and calculate the number of similar problems, then combine the number of likes and oppositions of the problem to quantify the heat index of hot issues, and extract the message time of similar problems, use custom The comparison time function calculates the nearest time and the farthest time, so as to determine the time range in the subsequent hot-spot question table, and writes the message information corresponding to the corresponding hot-spot question to the hot-spot question list according to the same kind of questions.

For question 3, customize a set of evaluation plans for the quality of the response comments of the relevant departments on the message, from whether the response comments are related to the question, whether the response comments are detailed and complete, whether the response comments are written, and whether they are used in the response comments The correct format: whether there is an adjective of "respect" or similar to "respect" for the netizen's title, and whether there is a civilized phrase "hello" or similar to "hello" in the reply. Keyword extraction is performed on the content of the reply and the details of the message, and the keywords of the two are compared, so that the relevance index of the reply opinion can be generated.

Key words: remove duplicates Chinese text segmentation TF-IDF weighted
 Text vectorization Named entity recognition Keyword extraction

目 录

1.	挖掘目标	- 5 -
2.	分析方法与过程	- 5 -
2.1	问题 1 分析方法与过程	- 7 -
2.1.1	问题 1 流程图	- 7 -
2.1.2	数据预处理	- 8 -
2.1.3	文本分类模型的处理	- 10 -
2.1.4	数据可视化处理	- 13 -
2.2	问题 2 分析方法与过程	- 15 -
2.2.1	问题 2 流程图	- 15 -
2.2.2	数据预处理	- 15 -
2.3	问题 3 分析方法与过程	- 17 -
2.3.1	问题 3 流程图	- 17 -
2.3.2	问题 3 分析	- 17 -
2.3.3	定义答复意见质量的评价方案	- 18 -
2.3.4	答复意见质量的评价方案的各个维度的解决方法	- 18 -
2.3.5	答复意见质量的评价方案的各个维度代码实现	- 20 -
2.3.6	TF-IDF 算法提取关键词	- 23 -
3.	智慧政务系统的益处	- 23 -
3.1	能更科学的指定相关政策和法规	- 24 -
3.2	政府能提高监管和服务的效率	- 24 -
3.3	社会风险的及时预警	- 24 -
4.	结论	- 24 -
5.	参考文献	- 25 -

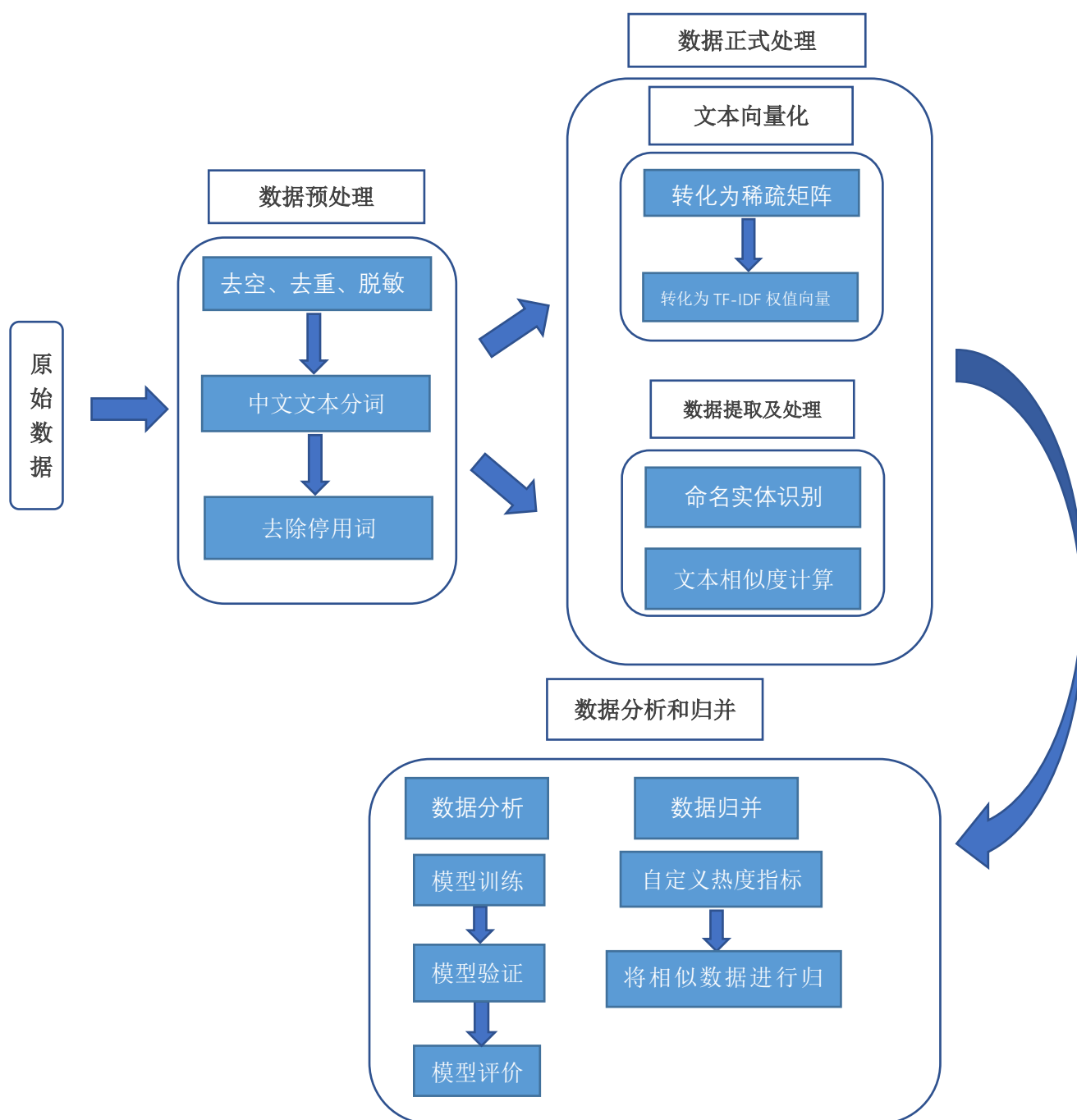
1. 挖掘目标

本次建模目标是利用“智慧政务”系统中的群众问政留言记录，利用 jieba 中文分词工具对群众问政留言记录中的留言详情进行分词、利用 Synonym 中文近义词工具进行留言主题计算文本相似度、利用 HanLP 自然语言处理工具对留言详情进行命名实体识别，达到以下三个目标：

- 1) 利用文本分词和文本去重以及将文本转化为 TF-IDF 权值向量进行构建文本分类模型，以及对分类模型进行评价
- 2) 对于文本进行文本相似度计算，并提取命名实体（地点/群体）、时间，并对热度指标进行量化，最后对热点问题归并
- 3) 自定义答复意见评价指标，从答复意见的相关性、完整性等角度对政府部门的答复意见进行全面的评价。

2. 分析方法与过程

本文的总体架构及思路如下：



图表 1 总体流程图

步骤一：数据预处理，对附件 2 非结构化文本数据去除重复项以及空格、中文文本分词、停用词过滤、脱敏处理，对附件 3 非结构化文本数据进行关键数据的提取，以便后续的数据分析和数据归并。

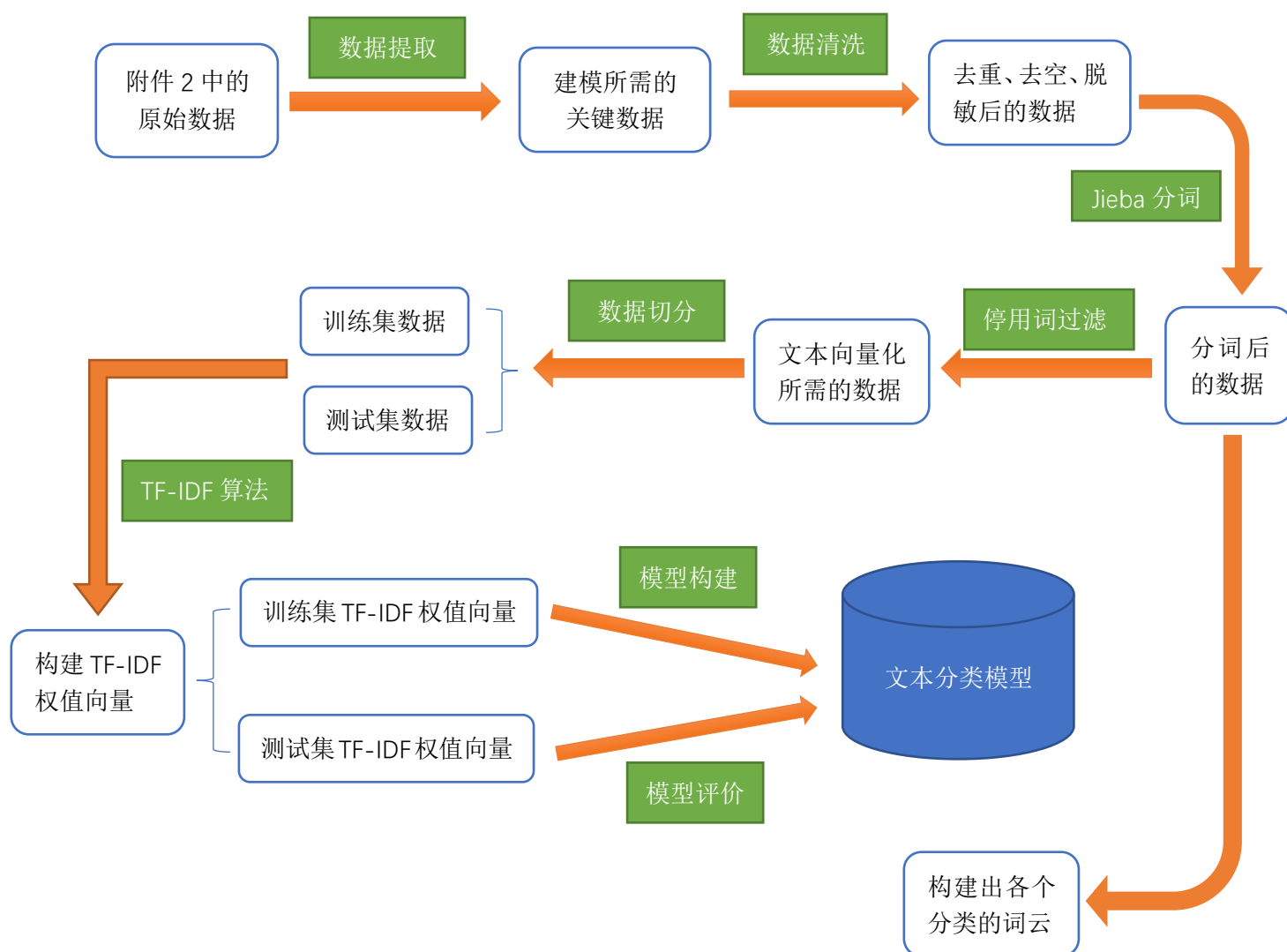
步骤二：数据正式处理，对于附件 2 中的预处理后的数据，采取 TF-IDF 权重策略将其文本向量化，使之成为能够被计算机识别的结构化信息，对附件 3 非结构化的文本数据进行文本

相似度计算，命名实体的识别与提取，以便后续的文本分类模型的建立和相同文本数据的归并。

步骤三：数据分析和归并，根据经过 TF-IDF 算法后得出的文本向量进行对文本多分类模型的构建，以及对模型的泛化能力和性能进行验证和评价。根据提取出的相关数据和文本相似度进行数据的归并，数据的顺序则根据自定义的热度评价指标进行相对的数据排名以及数据详情写入。

2.1 问题 1 分析方法与过程

2.1.1 问题 1 流程图



图表 2 问题 1 流程图

2.1.2 数据预处理

2.1.2.1 群众留言的去重、去空、脱敏

去重：

在题目给出的数据中，出现了重复的数据。例如留言详情中的数据有重复的留言内容。为了避免数据的冗余导致之后的数据建模的影响，则使用 `pandas` 库中的 `drop_duplicates()` 函数将重复的数据进行去除。

去空、脱敏：

在题目给出的数据中，虽然提供的数据已经对数据进行了处理，将一些真实的敏感信息进行了处理，并且用户的留言内容格式较正确，导致有空格的存在，数据读取过程中计算机会将空格识别为转义字符，所以敏感信息处理和转义字符会对之后的构建 TF-IDF 权值向量，甚至是模型构建、模型评价都是有一定的影响，为避免此问题的发生，我们需要使用正则表达式将每条数据中的进行去空和脱敏处理。

2.1.2.2 对建模所需的关键数据进行中文分词

在对网络问政平台的群众留言数据时，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。为了方便转换，要对附件 2 中的留言详情进行中文分词。这里采用了 `python` 的中文分词组件 `jieba` 进行中文分词。`jieba` 采用了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)，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，并且，`jieba` 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，使其分词变得更加准确。

2.1.2.3 文本的向量化表示

为了使非结构化的文本信息转化为计算机能够识别的结构化信息，我们需要将文本信息进行文本向量化表示，文本特征向量化方法通常有：1、词集模型：one-hot 编码向量化文本 2、词袋模型+IDF：TF-IDF 向量化文本。前者是将每个词表示为一个很长的向量，长度是由词袋长度决定的（词袋：所有词的不重复构成），词袋中的词语在句子中有存在则用 1 表示，

否则用 0 表示,从而将文本转化为词向量矩阵。One-hot 编码向量化文本的例子如下图表示:

词袋: [a, ate, cat, dolphin, dog, homework, my, sandwich, the]
文本 1: My dog ate my homework.
文本 1 的向量表示: [0 1 0 0 1 1 1 0 0]
文本 2: My cat ate the sandwich.
文本 2 的向量表示: [0 1 1 0 0 0 1 1 1]

图表 3 One-hot 编码向量化示例

One-hot 编码向量化文本的缺点是忽略了句子的词频信息。而 TF-IDF 权重策略相较于 One-hot 编码方式增加了词频信息和避免了句子长度不一致的问题。

2.1.2.4 TF-IDF 算法

在对留言详情信息进行分词后,需要将这些词语转换为向量,以供挖掘分析使用。

这里采用的是 TF-IDF 算法,把留言详情信息转换为权重向量。TF-IDF 算法的具体原理如下:

第一步,计算词频,即 TF(Term frequency)

词频(TF) = 某个词在文章中出现次数

考虑到文章长短之分,为了便于不同文章的比较,进行“词频”标准化。

$$\text{词频 (TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{文章的总次数}}$$

或者

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{该文出现次数最多的词的出现次数}}$$

第二步，计算逆文档频率，即 IDF(Inverse document frequency)

需要建立一个语料库(corpus)，用来模拟语言的使用环境。

$$\text{逆文档频率(IDF)} = \log \left(\frac{\text{语料库的文档总数}}{\text{包含该词的文档数} + 1} \right)$$

如果一个词越常见，那么分母就越大，逆文档频率就越小越接近 0。分母之所以要加 1，是为了避免分母为 0（即所有文档都不包含该词）。log 表示对得到的值取对数。

第三步：计算 TF-IDF 值(Term frequency-Inverse document frequency)

$$\text{TF-IDF} = \text{词频(TF)} \times \text{逆文档频率(IDF)}$$

TF-IDF 的值与一个词在文档中的出现次数成正比，与该词在整个语言中的出现次数成反比。

2.1.2.5 生成 TF-IDF 权值向量

生成 TF-IDF 权值向量的具体步骤如下：

- (1) 对分词后的留言详情的每条数据的每个词计算出 TF-IDF 权值，TF-IDF 权值计算公式如下：

$$\text{TF-IDF} = \text{词频(TF)} \times \text{逆文档频率(IDF)}$$

- (2) 将每条数据的词的 TF-IDF 权值放入的位置与每个词的位置相同，从而构成 TF-IDF 权值向量；

2.1.3 文本分类模型的处理

将切分后的训练集数据生成 TF-IDF 权值向量后，建立关于留言内容的一级标签分类模型。本题经过模型性能评价后最终使用的是高斯朴素贝叶斯模型算法，该模型算法在文本分类方面简单易行，且取得不错的分类效果。

2.1.3.1 高斯朴素贝叶斯模型算法

算法原理:

朴素贝叶斯分类(NBC)是以贝叶斯定理为基础并且假设特征条件之间相互独立的方法,先通过已给定的训练集,以特征词之间独立作为前提假设,学习从输入到输出的联合概率分布,再基于学习到的模型,输入 X 求出使得后验概率最大的输出 Y 。^[1]

设有样本数据集 $D = \{d_1, d_2, \dots, d_n\}$, 对应样本数据的特征属性集为 $X = \{x_1, x_2, \dots, x_d\}$ 类变量为 $Y = \{y_1, y_2, \dots, y_m\}$, 即 D 可以分为 y_m 类别。其中 x_1, x_2, \dots, x_d 相互独立且随机, 则 Y 的先验概率 $P_{prior} = P(Y)$, Y 的后验概率 $P_{post} = P(Y|X)$, 由朴素贝叶斯算法可得, 后验概率可以由先验概率 $P_{prior} = P(Y)$ 、证据 $P(X)$ 、类条件概率 $P = P(X|Y)$ 计算出:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)}$$

朴素贝叶斯基于各个特征之间相互独立, 在给定类别为 y 的情况下, 上式可以进一步表示为下式:

$$P(X|Y = y) = \prod_{i=1}^d P(x_i|Y = y)$$

由以上两式可以计算出后验概率为:

$$P_{post} = P(Y|X) = \frac{P(Y) \prod_{i=1}^d P(x_i|Y)}{P(X)}$$

由于 $P(X)$ 的大小是固定不变的, 因此在比较后验概率时, 只比较上式的分子部分即可。因此可以得到一个样本数据属于类别的朴素贝叶斯计算如下图所示:

$$P(y_i|x_1, x_2, \dots, x_d) = \frac{P(y_i) \prod_{j=1}^d P(x_j|y_i)}{\prod_{j=1}^d P(x_j)}$$

高斯朴素贝叶斯：

原始的朴素贝叶斯只能处理离散数据，当 x_1, \dots, x_n 是连续变量是，我们可以使用高斯朴素贝叶斯(Gaussian Naive Bayes)完成分类任务。

当处理连续数据时，一种经典的假设是：与每个类相关的连续变量的分布是基于高斯分布的，故高斯贝叶斯的公式如下：

$$P(x_i = v|y_k) = \frac{1}{\sqrt{2\pi\sigma_{y_k}^2}} \exp\left(-\frac{(v - \mu_{y_k})^2}{2\sigma_{y_k}^2}\right)$$

其中 μ_{y_k} ， $\sigma_{y_k}^2$ 表示全部属于类 y_k 的样本中变量 x_i 的均值和方差。

2.1.3.2 文本分类模型的评价

通常使用*F1 - Score*对分类方法进行评价，*F1 - Score*公式如下：

$$F1 = \frac{2PR}{P + R}$$

其中， P 为精确率（也叫查准率，precision），即正确预测为正的占全部预测为正的占全部预测为正的的比例，（真正正确的占所有预测为正的的比例）， R 为召回率(recall)，即正确预测为正的占全部实际为正的的比例（真正正确的占所有实际为正的的比例）。

以上*F1 - Score*通常适用于二分类模型评价，如果需要对于多分类模型评价，需要将 n 分类的评价拆分成 n 个二分类的评价，计算每个二分类的*F1 - Score*，则 n 个*F1 - Score*的平均值即为*Macro F1*。所以*Macro F1*可作为多分类模型评价指标，*Macro F1*公式如下：

$$Macro F1 = \frac{1}{n} \sum_{i=1}^n \frac{2P_i R_i}{P_i + R_i}$$

其中， P_i 为第 i 类的查准率， R_i 为第 i 类的查全率。

2.1.4 数据可视化处理

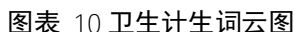
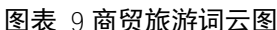
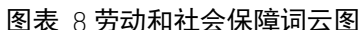
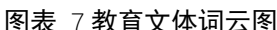
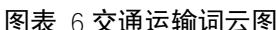
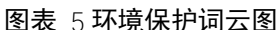
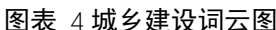
为了更加清晰直观的看出数据预处理后的数据结果,将预处理后的数据进行了词云图的绘制。

词云图过滤掉大量的文本信息,使浏览网页者只要一眼扫过文本就可以领略文本的主旨。^[2]
具体关键词词云图绘制过程如下:

第一步,构建一个空字典,将每个一级标签下的数据进行词频统计(字典中以词为键(key),以该词出现的次数为值(value))

第二步,词云图的绘制,在同一一级标签下的字典中的值(value)越大的,会将其相对应的键(key)——词语的字体变得越大。

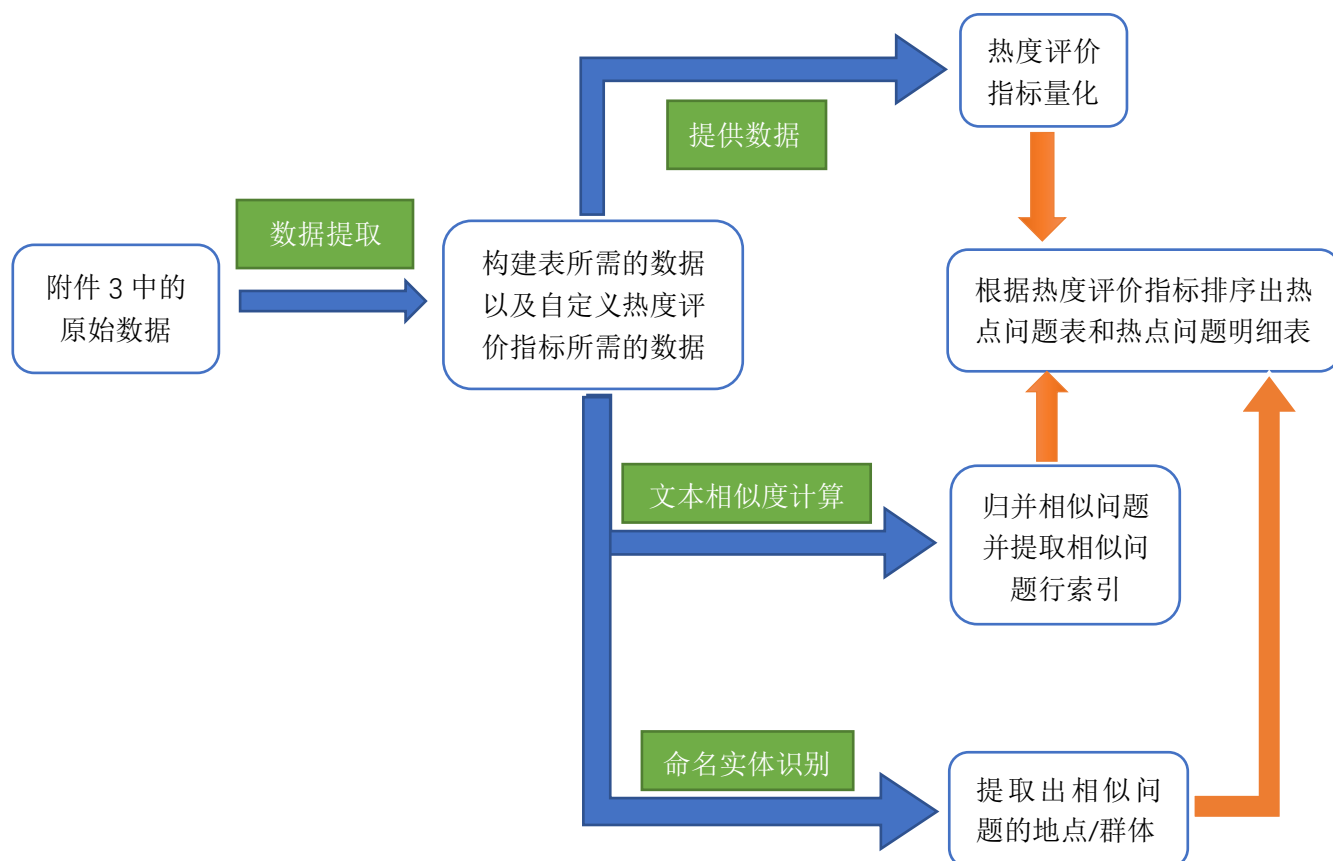
词云图的绘制效果如下所示:



- 14 -

2.2 问题 2 分析方法与过程

2.2.1 问题 2 流程图



图表 11 问题 2 流程图

2.2.2 数据预处理

2.2.2.1 定义合理的热度评价指标

热点问题的定义是某一时段内群众集中反映的某一问题。又经过对附件 3 的表的查看，内有对每个问题的网友点赞数与反对数，网友对于一个问题的点赞，也反映了网友认同此问题的存在，对一个问题的反对反映网友对此问题的存在有不认可。那么我们可以结合群众反映相同问题的数量，以及网友对问题的点赞数和反对数对热点问题的热度评价指标进行定义，如下：

$$\text{热度评价指标} = \text{群众反映相同问题的数量} \times 0.6 + (\text{点赞数} - \text{反对数}) \times 0.4$$

2.2.2.2 文本相似度计算

文本相似度计算的目的是将相同问题进行归并，而一个留言的留言主题能够体现出一个留言详情的大致内容，则使用每个留言的留言主题进行文本相似度计算并且比较其的相似度，根据相似度进行问题归并。这里采用了 python 的中文近义词工具包 Synonyms 进行对留言主题的相似度计算，Synonyms 中句子的相似度计算使用了词向量和编辑距离；根据词向量距离，采取梯度方式设置权重；并采取了平滑策略。

2.2.2.3 命名实体识别

命名实体识别的目的是将留言详情中的问题发生的地点和问题发生的相关群体识别出来，并对命名实体进行提取，以便后续对热点问题表中的列名为“地点/群体”的列进行相应的数据写入，这里采用了 python 的汉语言处理包 HanLP 进行对留言详情的命名实体识别并提取，HanLP 的命名实体识别是基于 HMM 角色标注的命名实体识别和线性模型的命名实体识别。

2.2.2.4 留言的时间范围提取

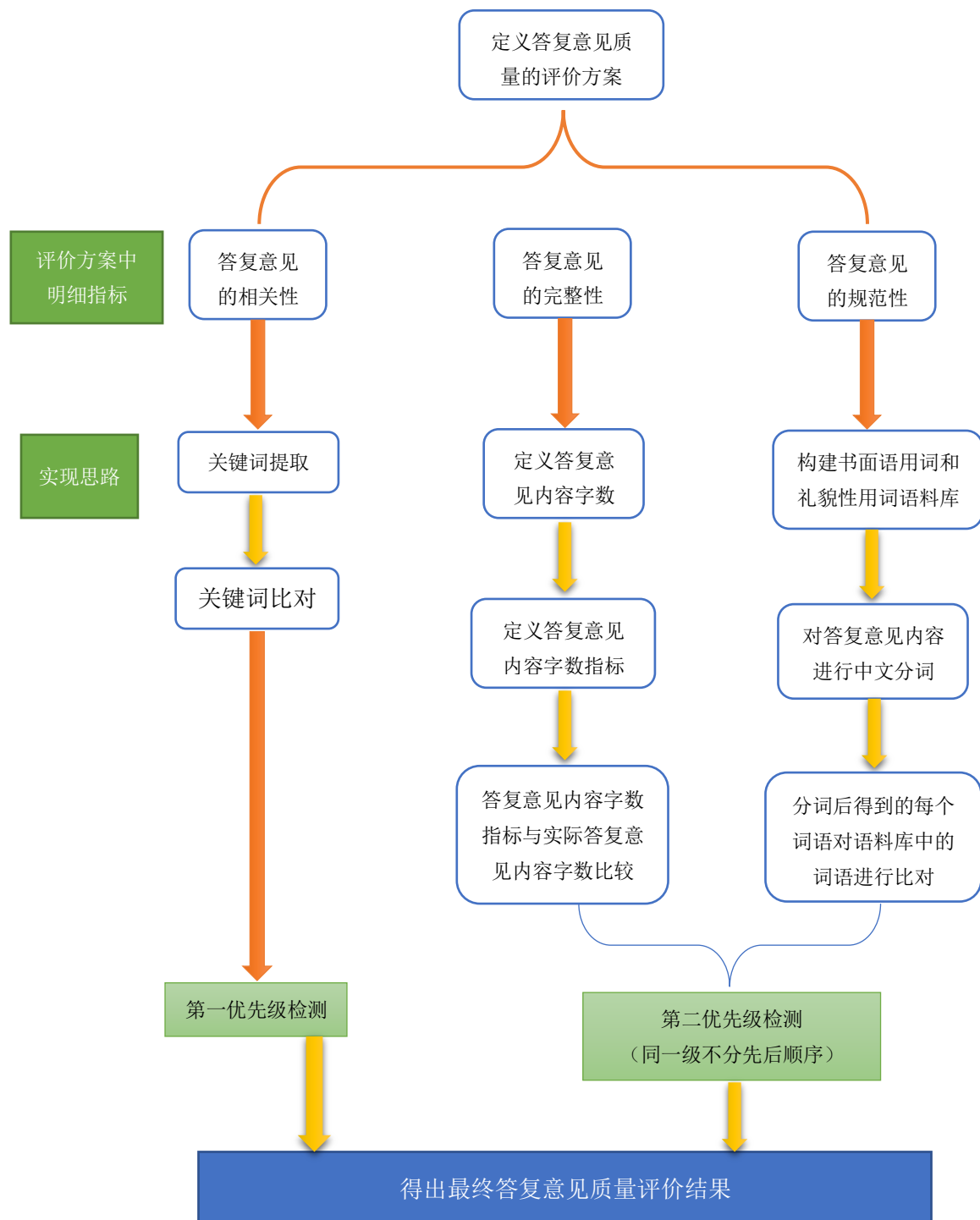
对附件 3 中的留言信息经过分析发现，每条留言都带有留言时间，则同类问题就有多个不同的留言时间，需要将其对每个时间进行对比得出最近的一个时间和最远的一个时间，以便确定时间范围，具体步骤如下：

第一步，由于时间提取后是字符串形式，要先将时间进行格式化这里采用了 time 库中的 strptime()和 mktime()，将字符串形式的时间进行格式化，返回后得到的是用秒数来表示时间的浮点数

第二步，将格式化后的时间进行比较，比较大小后即可返回最近的时间和最远的时间，即时间范围。

2.3 问题3 分析方法与过程

2.3.1 问题3 流程图



图表 12 问题3 流程图

2.3.2 问题3 分析

问题3 内容：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。

分析：该题是个开放性问题，开放性程度很高。对于答复意见的评价方案内容，首先我们需要对答复意见进行类比，例如，对客服咨询问题后，客服对顾客问题进行答复后，顾客对客服的答复进行评价，其中对客服答复评价的选项可运用到此问题中相关部门对留言的答复意见的质量评价方案中。

2.3.3 定义答复意见质量的评价方案

定义答复意见质量的评价方案，需要从多个角度进行对答复意见质量的评价方案，以下从多个维度介绍答复意见质量的评价方案：

- (1) 答复意见的相关性：即答复意见是否与群众留言详情中的问题相互关联、相互有联系；该维度的设定的目的是检查相关部门是否出现“答非所问”的现象。
- (2) 答复意见的完整性：即答复意见内容是否完整，是否详细，此维度的设定的目的是检查相关部门是否对群众留言详情中的问题进行抱有应付的心态或者是敷衍的心态去答复群众留言。
- (3) 答复意见的规范性：规范性是个较广的定义，内容较多，其中包括答复意见内容词语是否使用得当，即答复意见内容是否使用书面语，答复意见内容对于网友、群众是否使用礼貌用语。

2.3.4 答复意见质量的评价方案的各个维度的解决方法

我们将各个维度的解决方法进行了文字表述，并尝试进行在代码上的实现，以下为对答复意见质量的评价方案的各个维度的解决方法的文字表述内容：

- (1) 答复意见的相关性：相关性即是问题与答复是否有关联之处。这里将群众留言详情和答复意见内容中的关键词进行提取，关键词提取出后进行比对，查找关键词是否有相同或相似之处，其中关键词提取的方法采用的是 TF-IDF 算法。
- (2) 答复意见的完整性：完整性即答复意见内容的是否详细，是否完整。能够衡量该维度的指标目前只想到了字数比较。这里将群众留言详情内容和答复意见内容进行字数统计，然后进行比较，得出结果该答复意见内容是否完整、详细。答复意见内容字数应有个合理的指标，以便后续的答复意见内容是否完整、详细的判断，以下为

答复意见内容字数指标公式：

$$\text{答复意见内容字数} = \text{群总留言详情字数} \times 120\%$$

说明：当实际答复意见内容字数<该指标时，则判定为答复意见的完整性不足；当实际答复意见内容字数≥该指标时，则判定为答复意见较完整、详细。

- (3) 答复意见的规范性：规范性即答复意见内容中用词、格式等的使用规范。答复意见内容用词规范，用词规范包括书面语用词和礼貌性用词，这里需要对群众留言详情内容进行是否有书面语用词和礼貌性用词的判断。答复意见的格式规范，格式规范包括答复意见内容结尾的需要对群众、网友的致信感谢等，以及内容结尾需要有时进行标注。如下图中红色划线区域：

苑物业管理有问题”的调查核实情况向该网友答复如下：您好，首先感谢您对我们工作的信任。调查了解，针对来信所反映的“小区停车收费问题”，景蓉华苑业委会于2019年4月10日至4月15日出的意见和建议进行了认真梳理归纳并进行了反馈，业委会制定的停车收费标准不高于周边小区，区住房和城乡建设局也参加了会议。在综合各方面的意见后，辖区桂花坪街道、区住房和城乡建设局再执行相应的程序。再次感谢您对我区工作的理解和关心。2019年5月9日

图表 13 答复意见格式规范示例

在此之前需要构建一个书面语用词和礼貌性用词的语料库，再对答复意见内容使用python中的中文分词组件jieba进行分词，将分词后得到的每个词语对语料库中的词语进行比对，如有则说明该答复意见规范性较好，反之，则不好。

说明：该三个维度的检测应该有优先级，第一优先级应为答复意见的相关性检测；因为当答复意见的相关性检测不通过，则说明该留言答复可能出现了“答非所问”现象，所以后续的答复意见的完整性与规范性可不进行检测，直接对该“答非所问”的答复意见的完整性与规范性判定为不良。第二优先级应为答复意见的完整性的检测或答复意见的规范性的检测，属于同一优先级中的二者不分先后，即都需进行检测。

2.3.5 答复意见质量的评价方案的各个维度代码实现

- (1) 答复意见的相关性：关键词的提取这里采用的是 python 中的中文分词组件 jieba 中的 analyse 方法进行关键词提取，为避免大概率的相关性误判，经过多次的测试后，将关键词提取数量设定为 15 个，其中关键词提取的方法采用的是 TF-IDF 算法。然后将留言详情关键词与答复意见关键词进行对比，对比规则：当二者有一关键词相匹配，即可认为该答复意见相关性较好；如无一匹配则认为相关性不良。

```
for i in range(len(data3)):
    flag = 0
    question_keys = jieba.analyse.extract_tags(data3_question.loc[i, '留言详情'], topK=15)
    answer_keys = jieba.analyse.extract_tags(data3_answer.loc[i, '答复意见'], topK=15)
    for x in range(len(question_keys)):
        for y in range(len(answer_keys)):
            if question_keys[x] == answer_keys[y]:
                evaluation_df.loc[i, '相关性'] = '√'
                flag = 1
                break
    if flag == 0:
        evaluation_df.loc[i, '相关性'] = '×'
```

图表 14 答复意见的相关性部分代码示例

- (2) 答复意见的完整性：完整性的衡量本质是答复意见内容字符数与指标的比较，为防止留言详情、答复意见中的“\t、\n、\u3000、\xa0”以及空格的空白字符导致的字符数计算错误。需要先对文本进行去空操作，在进行字符计数。其中去空操作采用的是 replace()方法，去空操作代码示例如下：

```
#####留言详情、答复意见去空#####
for i in range(len(data3)):
    data3_question.loc[i, '留言详情'] = data3_question.loc[i, '留言详情'].replace('\n', '')
    data3_question.loc[i, '留言详情'] = data3_question.loc[i, '留言详情'].replace('\t', '')
    data3_question.loc[i, '留言详情'] = data3_question.loc[i, '留言详情'].replace('\u3000', '')
    data3_question.loc[i, '留言详情'] = data3_question.loc[i, '留言详情'].replace('\xa0', '')
    data3_question.loc[i, '留言详情'] = data3_question.loc[i, '留言详情'].replace(' ', '')
    data3_answer.loc[i, '答复意见'] = data3_answer.loc[i, '答复意见'].replace('\n', '')
    data3_answer.loc[i, '答复意见'] = data3_answer.loc[i, '答复意见'].replace('\t', '')
    data3_answer.loc[i, '答复意见'] = data3_answer.loc[i, '答复意见'].replace('\u3000', '')
    data3_answer.loc[i, '答复意见'] = data3_answer.loc[i, '答复意见'].replace('\xa0', '')
    data3_answer.loc[i, '答复意见'] = data3_answer.loc[i, '答复意见'].replace(' ', '')
```

图表 15 留言详情、答复意见去空操作部分代码示例

经过去空操作后，对去空后的留言详情、答复意见进行字符统计，根据留言详情计算出

答复意见内容字数（以下称为指标），指标计算公式如下：

$$\text{答复意见内容字数} = \text{群总留言详情字数} \times 120\%$$

将指标与答复意见字符数进行比较，比较规则：实际答复意见内容字数<指标时，则判定为答复意见的完整性不足；实际答复意见内容字数≥指标时，则判定为答复意见较好。

```
#====完整性=====
for i in range(len(data3)):
    if evaluation_df.loc[i,'相关性'] == '×':
        evaluation_df.loc[i,'完整性'] = '×'
        evaluation_df.loc[i,'规范性'] = '×'
    elif evaluation_df.loc[i,'相关性'] == '√':
        index = len(data3_question.loc[i,'留言详情']) * 1.2
        if len(data3_answer.loc[i,'答复意见']) < index:
            evaluation_df.loc[i,'完整性'] = '×'
        else:
            evaluation_df.loc[i,'完整性'] = '√'
```

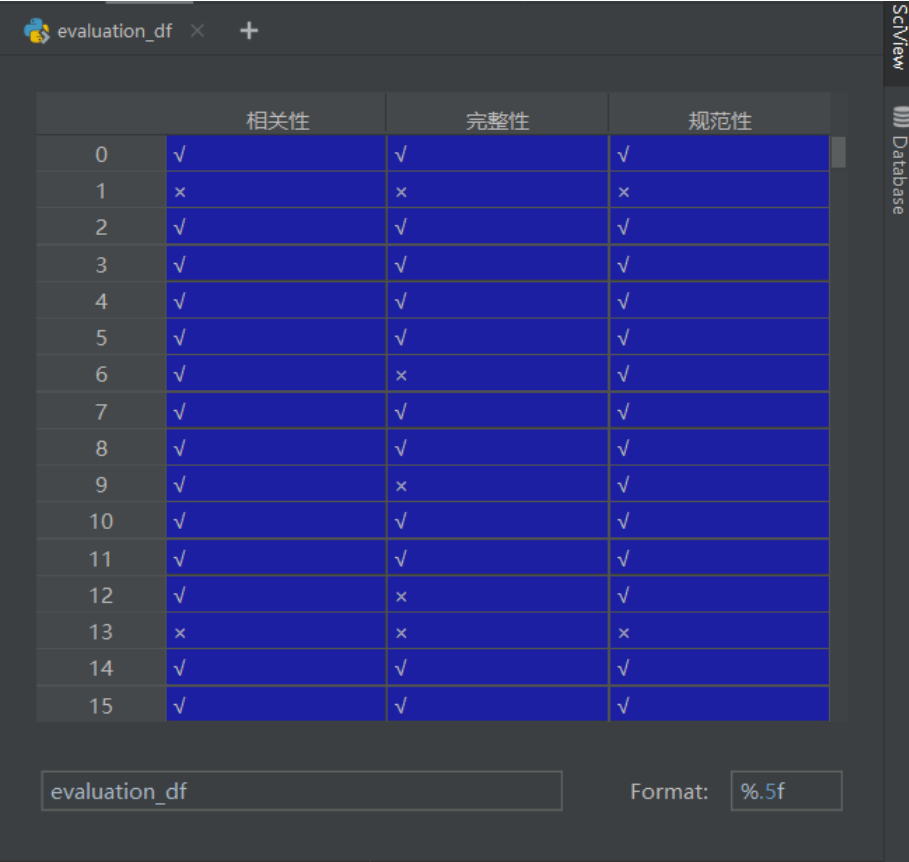
图表 16 指标与答复意见字符数比较操作部分代码示例

- (3) 答复意见的规范性：构建书面语和礼貌性用词的语料库后，再对每一条的答复意见使用 python 中的中文分词组件 jieba 进行中文分词，对分词后的结果与语料库中的词进行比对，比对规则：如二者有相同，即答复意见中存在语料库中的词汇，则规范性较好；反之，当答复意见中不存在语料库中的词汇时，则判定为该答复意见的规范性较差。以下为答复意见的规范性代码实现部分：

```
for x in range(len(data3_answer_cut_list)):
    if evaluation_df.loc[x,'规范性'] == 'x':
        continue
    else:
        flag2 = 0
        for y in data3_answer_cut_list[x]:
            for i in words:
                if y == i:
                    evaluation_df.loc[x,'规范性'] = '√'
                    flag2 = 1
                    break
        if flag2 == 0:
            evaluation_df.loc[x,'规范性'] = 'x'
```

图表 17 答复意见的规范性检测部分代码示例

- (4) 对于答复意见的评估会专门生成一张评估结果表，里面包含了答复意见的各个维度的检测结果，评估结果表的效果图如下：



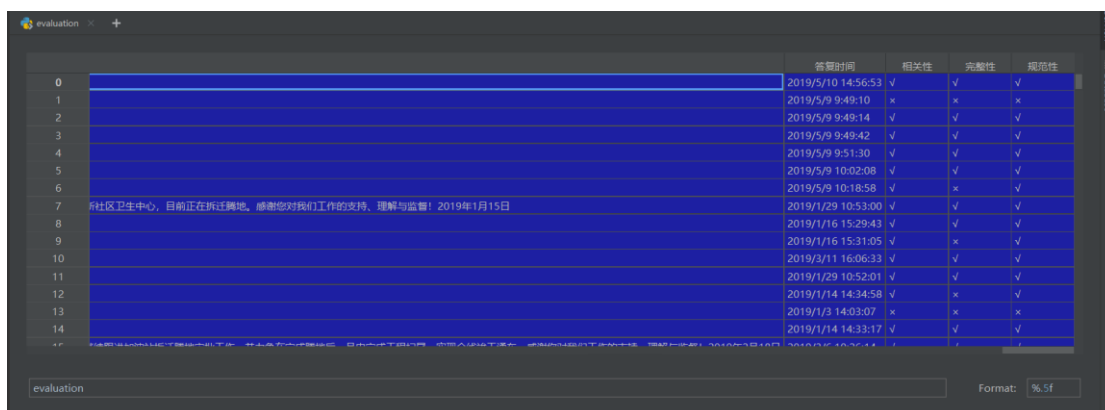
	相关性	完整性	规范性
0	√	√	√
1	x	x	x
2	√	√	√
3	√	√	√
4	√	√	√
5	√	√	√
6	√	x	√
7	√	√	√
8	√	√	√
9	√	x	√
10	√	√	√
11	√	√	√
12	√	x	√
13	x	x	x
14	√	√	√
15	√	√	√

图表 18 评估结果表效果图

说明：所得到的评估结果表保存在“答复意见质量评价表.xlsx”中，行索引与该条的答复意

见行索引相同，随着留言详情与答复意见内容及时更新。

为了更加直观的看到每条答复意见内容与答复意见质量的评价结果，又将群众留言详情、部门答复意见、答复意见质量评估三张表进行合并，得到结果效果图如下：



		答复时间	相关性	完整性	规范性
0		2019/5/10 14:56:53	√	√	√
1		2019/5/9 9:49:10	×	×	×
2		2019/5/9 9:49:14	√	√	√
3		2019/5/9 9:49:42	√	√	√
4		2019/5/9 9:51:30	√	√	√
5		2019/5/9 10:02:08	√	√	√
6		2019/5/9 10:18:58	√	×	√
7	社区卫生中心，目前正在拆迁腾地。感谢您对我们工作的支持、理解与监督！2019年1月15日	2019/1/29 10:53:00	√	√	√
8		2019/1/16 15:29:43	√	√	√
9		2019/1/16 15:31:05	√	×	√
10		2019/3/11 16:06:33	√	√	√
11		2019/1/29 10:52:01	√	√	√
12		2019/1/14 14:34:58	√	×	√
13		2019/1/3 14:03:07	×	×	×
14		2019/1/14 14:33:17	√	√	√

图表 19 三表合并结果效果图

说明：所得到的三表合并的表保存在“直观答复意见质量评价表.xlsx”中，行索引与该条的答复意见行索引相同，随着留言详情与答复意见内容及时更新。

2.3.6TF-IDF 算法提取关键词

使用 TF-IDF 算法提取关键词时，关键词起初默认为在该文本中出现次数最多的词，但单独根据词频（TF）进行关键词提取，可能会出现关键词的提取。所以需要一个重要性调整系数，衡量一个词是不是常见词。如果某个词比较少见，但是它在该文本中多次出现，那么它很可能就反映了该文本的特性，正是该文本所需要的关键词。用统计学语言表达，就是在词频基础之上，对每个词分配一个“重要性”权重，则该权重称为“逆文档频率”（IDF）。根据 TF-IDF 算法得出每个词的 TF-IDF 值（TF-IDF 算法详情请见该篇文章中的 [2.1.2.4TF-IDF 算法](#) 部分）。某个词对文章重要性越高，则它的 TF-IDF 值就越大。所以，只需将文本中的每个词根据 TF-IDF 值进行排序，排名靠前的几个词语，即可作为该文本中的关键词。

3、智慧政务系统的益处

建立智慧政务平台，基于该平台可以实现政府部门为社会民众的智慧服务功能，同时为政府部门提供智慧预警和智慧决策。

3.1 能更科学的指定相关政策和法规

通过各级政府和部门将数据进行共享，对数据形成集聚效应。再使用大数据分析工具对一些宏观数据进行分析，建立一些经济模型，从而对经济社会的发展形成预判，这样有利于政府部门科学的指定相关的政策和法规，从而更好的为社会提供服务。

3.2 政府能提高监管和服务的效率

通过大数据平台将问题进行分类统计，对分类后的问题派发到相关处理问题的部门，进而大大提高了政府的服务的效率，以及政府通过大数据的应用能实现对市场的监管。

3.3 社会风险的及时预警

利用大数据可以对社会舆论风险进行预警，通过爬虫程序在各大论坛和相关网页上的查找，可以实现对一些敏感性词汇进行统计，对一些社会舆论反应激烈的事件，政府部门在事件爆发前就能预知，从而可以提前处理，维持社会的稳定。

4、结论

总结本次比赛，我们基于 TF-IDF 权重策略将文本特征向量化，构建出文本多分类模型，通过 *Macro F1* 分数，对模型进行性能量化，再根据 *Macro F1* 分数对模型进行调整，得到最终质量较高的模型；对文本进行相似度计算后进行问题归并，并进行了中文命名实体的识别与提取，再根据自定义的热度指标进行热点问题的挖掘；我们基于 TF-IDF 算法提取文本中的关键词，构建出书面语、礼貌性用词的语料库，将答复意见质量进行全方面的评估。

但是我们认为自定义的答复意见质量评价方案不够完善，对评价方案包含的三个维度中的评价方法略有简单，在评估结果上就只有好坏之分，应该再将结果进一步精细化，使评价方案层次化；以及我们在文本相似度计算的结果准确度不是特别的好，这可能是由于语料库不够完善与 word2vec 根据上下文做推断无法区分同义词和反义词造成的，这也涉及到当今中文文本挖掘的不足，我们后期也会进一步对文本挖掘进行深入探讨。

5、参考文献

- [1] 张航. 基于朴素贝叶斯的中文文本分类及 Python 实现. 山东师范大学. 2018
- [2] 紫竹. “词云”——网络内容发布新招式. 人民网. 2006
- [3] 国务院办公厅关于运用大数据加强对市场主体服务和监管的若干意见[R]. 国办发(2015) 51 号. 2015-7-1.
- [4] 国务院办公厅关于运用大数据加强对市场主体服务和监管的若干意见[R]. 国办发(2015) 51 号. 2015-7-1.