基于 FastText 的智慧政务系统

摘 要

随着网络问政平台的发展，汇聚社情民意的文本数据量不断攀升，给相关部门的工作带来了极大挑战。网络问政平台作为政府和群众之间的"民意连心桥"，如何高效准确地对平台的群众留言进行分类，有关部门对热点问题及时处理和对留言的答复显得尤为重要。本文通过自然语言处理技术与文本挖掘建立了一个智慧政务系统，有助于提升政府的管理水平和施政效率。

对于问题一，本文先对附件 2 的数据进行特征预处理，即正则表达式替换，jieba 分词和去除停用词。针对 jieba 分词中后存在未划分的比较长的属于词，本文利用逆向最大匹配分词算法 BMM 借助自建词表实现最佳分词加以改进。接着，采用 TF-IDF 算法进行关键词提取后打标签，再利用 Fasttext 文本分类框架构建分类模型，实现有监督训练的文本分类。使用 F-Score 对分类方法进行评价，得到 $F_1 = 0.97$ 。

对于问题二，本文提出了留言热度指数的衡量，包括点赞数和反对数、关注度和文本热度指数。首先，第一个指标可以直接从附件 3 获取，在一定程度上反应这个问题的关注度情况。第二，文本热度指数需要从留言具体内容的角度分析，对文本数据进行同样的预处理，针对预处理之后留言文本进行词频统计分析，再根据词频共现算法获取关键词指数 Jaccard 系数，从而量化文本之间的相关关系。最后对文本相关系数矩阵进行 K-means 聚类，留言文本到中心簇的距离即为文本热度指数，将距离聚类中心簇较远的留言视为热点问题。综合三个指标，我们得到排名前 5 的热点问题及详细信息汇总到附件数据文件夹中。

对于问题三，我们从相关性，完整性以及可解释性的角度制定一套评价方案。首先，对于答复意见的相关性质量，通过利用前文的关键词抽取算法，抽取出留言的关键词文本和答复意见文本计算莱文斯坦相似度，将计算结果归一化获取答复意见与问题的相关性系数。此外，对于答复意见的完整性评价，我们利用前文的文本分词算法，通过文本词长度统计来衡量，一般来说文本含有的词语越多，回复意见越完整。其次，对于答复意见的可解释性评估，利用字符串匹配来获取，对答复意见中高频词和出现在问题中高频词的次数进行统计。最终量化出三个评价指标，然后建立熵权综合评价模型，对每个答复意见的质量进行评价。

关键词：组合分词；Fasttext；K-means 聚类；留言热度指数；熵权综合评价模型

Smart government system based on FastText

Abstract

With the development of the online questioning platform, the amount of textual data that gathers the feelings and opinions of the society continues to rise, which brings great challenges to the work of relevant departments. As a "public opinion bridge" between the government and the masses, the network questioning platform efficiently and accurately classifies the masses' messages on the platform. It is particularly important for relevant departments to deal with hot issues in a timely manner and reply to the messages. This article establishes a smart government system through natural language processing technology and text mining, which helps to improve the government's management level and governance efficiency.

For question one, this article first performs feature preprocessing on the data in Annex 2, namely regular expression replacement, jieba word segmentation and removal of stop words. Aiming at the jieba word segmentation, there is a relatively long belonging word that is not divided, this paper uses the reverse maximum matching word segmentation algorithm BMM to improve the best word segmentation by means of a self-built word list. Then, the TF-IDF algorithm is used for keyword extraction and labeling, and then the Fasttext text classification framework is used to build a classification model to implement supervised training text classification. Use F-Score to evaluate the classification method and get $F_1 = 0.97$.

For question two, this article proposes the measurement of the message popularity index, including likes and dislikes, attention and text popularity index. First of all, the first indicator can be obtained directly from Annex 3, reflecting the degree of attention to this issue to a certain extent. Second, the text heat index needs to be analyzed from the perspective of the specific content of the message, the same preprocessing is performed on the text data, and the word frequency statistical analysis is performed on the message text after preprocessing. According to the word frequency co-occurrence algorithm, the keyword index is obtained to quantify the correlation between the texts. Finally, the text correlation coefficient matrix is K-means clustered. The distance from the message text to the central cluster is the text heat index. Messages farther from the center cluster are regarded as hot issues. Combining the three indicators, we get the top 5 hot issues and detailed information in the attachment data folder.

For question three, we formulate an evaluation plan from the perspective of relevance, completeness and interpretability. First, for the relevance quality of the reply opinions, the keyword text of the message and the reply opinion text are extracted by using the keyword extraction algorithm above to calculate the Levinstein similarity, and the calculation result is normalized to obtain the correlation between the reply opinion and the question Sexual coefficient. In addition, for the

evaluation of the completeness of the reply opinion, we use the text segmentation algorithm of the previous text to measure it through the text word length statistics. Generally speaking, the more words the text contains, the more complete the reply opinion. Secondly, for the interpretability evaluation of reply opinions, use string matching to obtain, and count the frequency of high-frequency words and high-frequency words appearing in the question. Finally, three evaluation indicators were quantified, and then an entropy weight comprehensive evaluation model was established to evaluate the quality of each response.

Key words: Combined word segmentation; Fasttext; K-means clustering; message popularity index; entropy weight comprehensive evaluation model.

目录

1. 挖掘目标

随着网络问政平台上各类社情民意相关的文本数据量不断攀升，只依靠人工来进行留言划分和热点整理的工作难度加大。本次建模是利用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，先对数据进行清洗，提取关键词，利用 FastText，建立关于留言内容的一级标签分类模型；接着，对于某一时段内反映特定地点或特定人群问题的留言，采用多个指标评价热度，得到排名前 5 的热点问题；最后，针对相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等进行评价。

2. 分析方法与过程

2.1. 总体流程

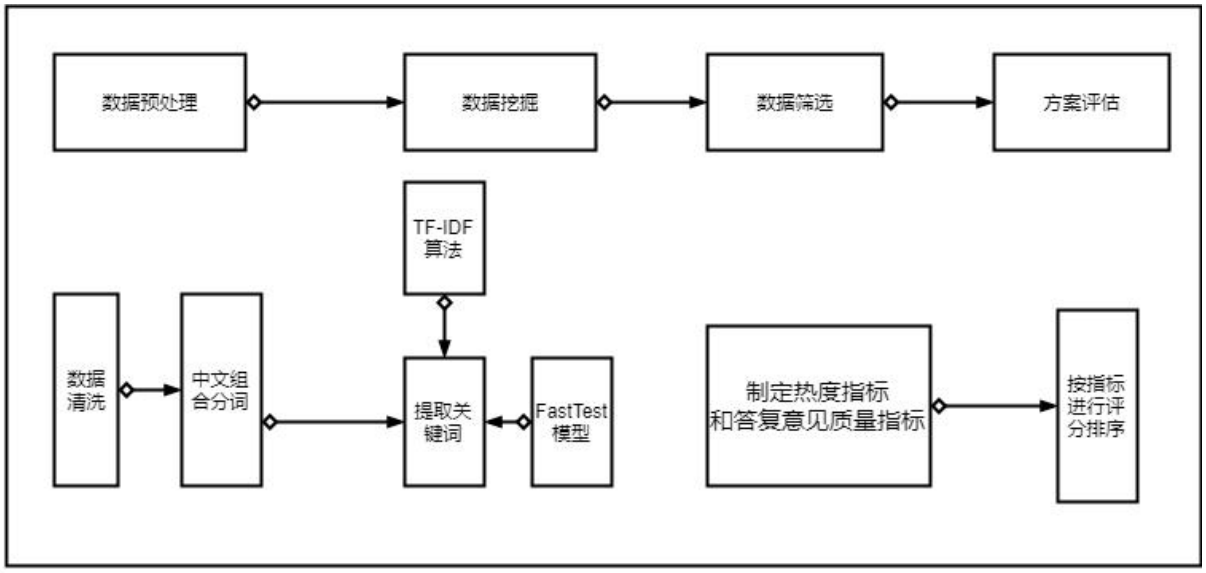


图 1：总体流程图

本用例主要包括如下步骤：

步骤一：在题目所给数据中，我们需要提取中文字符，需要用正则表达式替换，jieba 和逆向最大匹配分词算法（BMM）组合分词，去除停用词

步骤二：预处理得到的数据，使用 TF-IDF 算法提取关键词，在将其用于 FastTest 模型训练

步骤三：对于热点问题挖掘和答复意见的评估方案，则需要对其进行指标的确定和量化，

再进行评分和排序得到结果。

2.2. 问题一的分析方法与过程

2.2.1. 流程图

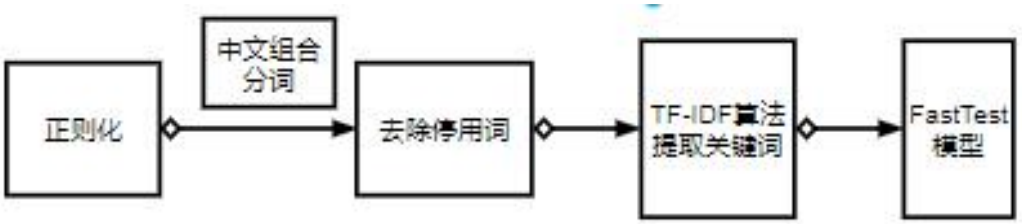


图 2：问题 1 流程图

本题要求针对文本留言内容建立一个一级标签分类模型。为了能够较好的应用分类模型，我们首先针对文本数据进行特征预处理，利用正则替换，jieba 分词，去除停用词等手段来清洗数据。进一步利用清洗之后的特征数据进行分类建模。对于分词部分，虽然利用了 jieba 智能分词模块，但仍旧有很多比较长的属于词很难完整的划分出来，因此我们利用逆向最大匹配分词算法 BMM 借助自建词表实现最佳分词。

利用预处理之后的分类特征，我们首先尝试利用关键词提取以及关键词和标签词的最大相似度匹配来实现文本分类。其中关键词提取主要采用 TFIDF 算法来实现文本关键词抽取，但是考虑到这种方法是一种无监督的分类算法，准确率可能比较低，因此我们考虑利用表征学习进行词嵌入，进一步利用开源 Fasttext 文本分类框架构建分类模型，从而实现有监督训练的文本分类。

2.2.2. 数据预处理

预处理是文本分类中的重要一环，首先对题目所给的文本数据进行特征预处理，主要包括正则表达式替换，jieba 分词，去除停用词等手段来清洗数据。

✧ 正则表达式

正则表达式是一种可以用于模式匹配和替换的规范，它是由普通的字符及特殊字符组成的文字模式，用于描述在查找文字主体时待匹配的一个或多个字符串。它可以用来验证字符串是否符合指定特征并用来查找字符串。在本题中我们采用正则表达式，去除所有非中文字符，提取中文字符有利于后续处理。

✧ 分词

我们采用 Python 的中文包 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。我们发现仍旧有很多比较长的属于词很难完整的划分出来，因此利用逆向最大匹配分词算法 BMM 借助自建词表实现最佳分词。BMM 大致过程为：

1.输入最大词长 maxWordLength，字典 wordDict，待分句子。

2.从待分句子的末尾开始向前截取长度为 maxWordLength 的子句，进行分词。

3.对一个子句的分词过程为，首先判断子句是否在字典中，若在，则保存这个子句，并从原句中删除这个子句，转到 2。

若不在，则判断子句长度是否为 1，若为 1，则将单字保存，从原句中删除单字，转到 2。若不为 1，则将子句中最右边的一个字删除，形成新的子句，转到 3。

通过以上组合分词，我们可以得到更为精确的分词结果。

✧ 去除停用词

停用词库中主要包括英文字符、数字、数字字符、标点符号及停用频率特高的单汉字，如语气助词、副词、介词、连词等并无明确意义的词。像“由于”“马上”“因此”等，去除后可以加快对分类提取特征时的过程。

2.2.3. 关键词提取

利用预处理之后的分类特征，我们首先尝试利用关键词提取以及关键词和标签词的最大相似度匹配来实现文本分类。关键词提取主要采用 TF-IDF 算法来实现文本关键词抽取，具体原理如下：

第一步：计算词频，即 TF 权重（Term Frequency）

词频是某个词在文本中出现的次数，考虑到文段有长短之分，为了便于描述，需要进行“词频”标准化，除以文本中的总词数或者除以该文本中出现次数最多的词的出现次数即:

$$\text{词频}=\frac{\text{某个词在文本中的出现次数}}{\text{文本的总词数}}$$

或

$$\text{词频}=\frac{\text{某个词在文本中的出现次数}}{\text{该文本出现次数最多的词的出现次数}}$$

第二步，计算 IDF 权重，即逆文档频率（Inverse Document Frequency），需要建立一个语料库（corpus），用来模拟语言的使用环境。IDF 越大，此特征性在文本中的分布越集中，说明该分词在区分该文本内容属性能力越强。

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

第三步，计算 TF-IDF 值（Term Frequency Document Frequency）

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

实际分析得出 TF-IDF 值与一个词在留言内容文本出现的次数成正比，某个词文本的重要性越高，TF-IDF 值越大。计算文本中每个词的 TF-IDF 值，进行排序，次数最多的即为要提取的留言文本关键词。

2.2.4. FastText 文本分类器

考虑到以上方法是一种无监督的分类算法，准确率可能比较低，因此我们考虑利用表征学习进行词嵌入，进一步利用开源 Fasttext 文本分类框架构建分类模型，从而实现有监督训练的文本分类。

FastText 是 Facebook 开发的一款快速文本分类器，旨在协助创建文本表达和分类的可伸缩解决方案的资料库，提供简单而高效的文本分类和表征学习的方法，性能比肩深度学习而且速度更快。FastText 模型输入一个词的序列（一段文本或者一句话)，输出这个词序列属于不同类别的概率。输入序列中的词和词组组成特征向量，特征向量通过线性变换映射到中间层，中间层再映射到标签。并且，在预测标签时使用了非线性激活函数，但在中间层不使用非线性激活函数。FastText 不需要预训练好的词向量，会自己训练词向量；同时有两个重要的优化：Hierarchical Softmax、N-gram。

FastText 模型架构和 Word2Vec 中的 CBOW 模型很类似。不同之处在于，FastText 预测标签，而 CBOW 模型预测中间词。

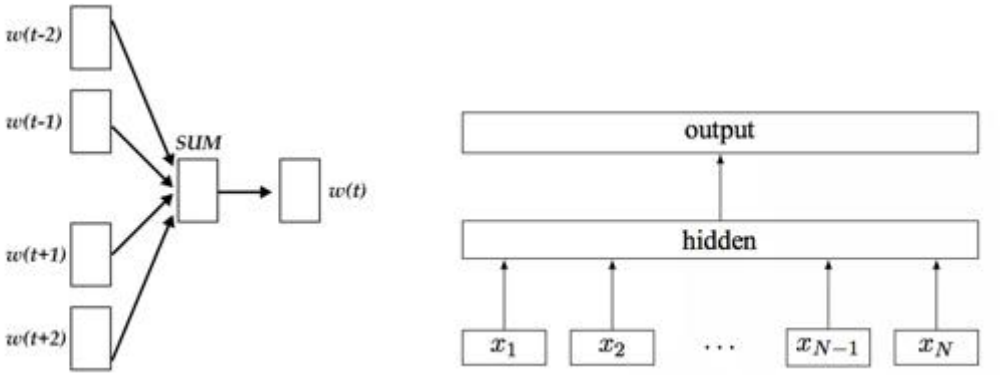


图 3：左图为 CBOW 模型，右图为 fastText 模型架构

word2vec 将上下文关系转化为多分类任务，进而训练逻辑回归模型，这里的类别数量|V|词库大

小。通常的文本数据中，词库少则数万，多则百万，在训练中直接训练多分类逻辑回归并不现实。word2vec 中提供了两种针对大规模多分类问题的优化手段， negative sampling 和 hierarchical softmax。在优化中，negative sampling 只更新少量负面类，从而减轻了计算量。hierarchical softmax 将词库表示成前缀树，从树根到叶子的路径可以表示为一系列二分类器，一次多分类计算的复杂度从|V|降低到了树的高度。

fastText 模型架构中 $x_1, x_2, \ldots, x_{N-1}, x_N$ 表示一个文本中的 n-gram 向量，每个特征是词向量的平均值。这和前文中提到的 CBOW 相似，CBOW 用上下文去预测中心词而此处用全部的 n-gram 去预测指定类别。

层次 softmax 函数常在神经网络输出层充当激活函数，目的就是将输出层的值归一化到 0-1 区间，将神经元输出构造成概率分布，主要就是起到将神经元输出值进行归一化的作用。在标准的 softmax 中，计算一个类别的 softmax 概率时，我们需要对所有类别概率做归一化，在这类别很大情况下非常耗时，因此提出了分层 softmax (Hierarchical Softmax)，思想是根据类别的频率构造霍夫曼树来代替标准 softmax，通过分层 softmax 可以将复杂度从 N 降低到 logN，下图给出分层 softmax 示例：

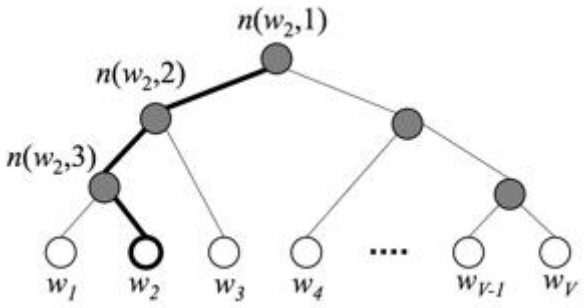


图 4：softmax 模型

在层次 softmax 模型中，叶子结点的词没有直接输出的向量，而非叶子节点都有响应的输在在模型的训练过程中，通过 Huffman 编码，构造了一颗庞大的 Huffman 树，同时会给非叶子结点赋予向量。我们要计算的是目标词 w 的概率，这个概率的具体含义，是指从 root 结点开始随机走，走到目标词 w 的概率。因此在途中路过非叶子结点（包括 root）时，需要分别知道往左走和往右走的概率。例如到达非叶子节点 n 的时候往左边走和往右边走的概率分别是：

$$p(n, left) = \sigma({\theta_n}^T \cdot h)$$

$$p(n, right) = 1 - \sigma({\theta_n}^T \cdot h) = \sigma(-{\theta_n}^T \cdot h)$$

以上图中目标词为 w_2 为例，

$$\begin{aligned} p(w_2) &= p(n(w_2,1), left) \cdot p(n(w_2,2), left) \cdot p(n(w_2,3), right) \\ &= \sigma({\theta_{n(w_2,1)}}^T \cdot h) \cdot \sigma({\theta_{n(w_2,2)}}^T \cdot h) \cdot \sigma(-{\theta_{n(w_2,3)}}^T \cdot h) \end{aligned}$$

到这里可以看出目标词为 w 的概率可以表示为：

$$p(w)=\prod_{j=1}^{L(w)-1}\sigma(sign(w,j)\cdot\theta_{n(w,j)}{}^{T}h)$$

其中 $\theta_{n(w,j)}$ 是非叶子结点 $n_{(w,j)}$ 的向量表示（即输出向量）；h 是隐藏层的输出值，从输入词的向量中计算得来；$sign(x,j)$ 是一个特殊函数定义为

$$sign(w,j)=\begin{cases}1, & \text{若}n(w,j+1)\text{是}n(w,j)\text{的左孩子}\\ -1, & \text{若}n(w,j+1)\text{是}n(w,j)\text{的右孩子}\end{cases}$$

此外，所有词的概率和为 1，即

$$\sum_{i=1}^{n}p(w_i)=1$$

最终得到参数更新公式为

$$\theta_j^{(new)}=\theta_j^{(old)}-\eta(\sigma(\theta_j{}^{T}h)-t_j)h$$

其中，$j=1,2,\ldots,L(w)-1$

n-gram 是基于语言模型的算法，基本思想是将文本内容按照子节顺序进行大小为 N 的窗口滑动操作，最终形成窗口为 N 的字节片段序列。而且需要额外注意一点是 n-gram 可以根据粒度不同有不同的含义，有字粒度的 n-gram 和词粒度的 n-gram。对于文本句子的 n-gram 来说，可以是字粒度或者是词粒度，同时 n-gram 也可以在字符级别工作，使用 n-gram 有如下优点：

1. 为罕见的单词生成更好的单词向量：根据上面的字符级别的 n-gram 来说，即是这个单词出现的次数很少，但是组成单词的字符和其他单词有共享的部分，因此这一点可以优化生成的单词向量

2. 在词汇单词中，即使单词没有出现在训练语料库中，仍然可以从字符级 n-gram 中构造单词的词向量

3. n-gram 可以让模型学习到局部单词顺序的部分信息, 如果不考虑 n-gram 则便是取每个单词，这样无法考虑到词序所包含的信息，即也可理解为上下文信息，因此通过 n-gram 的方式关联相邻的几个词，这样会让模型在训练的时候保持词序信息.

附件 2 所给的留言数据，进行正则表达式替换，分词和去除停用词，再提取关键词后得到的数据，按照训练集：验证集 = 8:2 直接进行 FastText 训练得到模型。程序见附件 fasttest.py，数据集为 question1_alldata.txt。

2.3. 问题二的分析方法与过程

2.3.1. 流程图

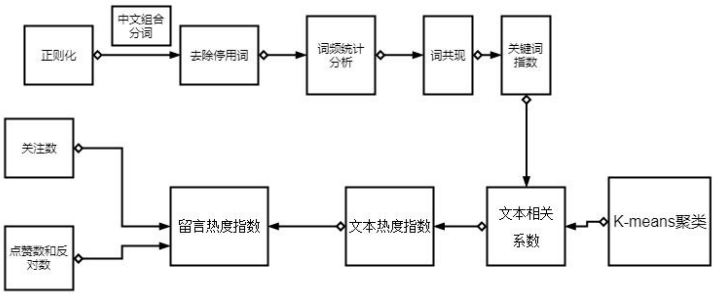


图 5：问题 2 流程图

本题要求针对热点问题进行挖掘，主要目的是从群众留言中挖掘出热点问题。也就是给每一条留言都量化一个热度指数。并且根据热度指数进行排序，从而获取热度较高的评价问题。

对于热度指数的量化，我们通过对附件三数据可以发现问题点赞数与反对数可以在一定程度上反应这个问题的关注度情况。因此问题点赞数与反对数也是衡量问题热度的一个重要指标。比如问题点赞数量越多，就越说明这个问题反应人民群众的心声。进一步我们考虑从留言具体内容的角度来研究留言热度。首先对文本数据进行预处理，同样包括正则字符处理，jieba 分词，然后针对预处理之后留言文本进行词频统计分析。进一步根据词频共现算法来获取关键词指数 Jaccard 系数。根据关键词指数量化文本之间的相关关系，然后根据文本相关系数进行聚类。从而将距离聚类中心簇较远的留言视为热点问题。

文本热度指数可以根据留言到中心簇的距离公式来量化，再综合考虑点赞数与反对数指标，从而加权归一化得到整体的留言热度指数。进一步排序获取最终的结果。

2.3.2. 数据预处理

在本题中，我们同样需要对数据进行预处理，包括正则表达式处理非中文字符，jieba 分词和去停用词，然后针对预处理之后留言文本进行词频统计分析。

2.3.3. 词频统计分析

词频分析(Word Frequency Analysis)是对文献正文中重要词汇出现的次数进行统计与分析，是文本挖掘的重要手段。它是文献计量学中传统的和具有代表性的一种内容分析方法，基本原理是通过词出现频次多少的变化，来确定热点及其变化趋势。

2.3.4. 词共现模型

词共现算法是建立在词频统计算法的基础上，将词语及其语义关系映射到词语共现图上，利用在词共现图上形成的主题信息和不同主题之间的连接特征信息，自动的提取文档中的主题词，主要目的是找出一些非高频并且对主题贡献大的词作为关键词。

在自然语言文本中普遍存在词共现现象，而在特定的某一类文本中这种现象更加明显。词共现即某些相关词汇会出现在一定的文本范围内，本题将该范围规定为留言内容内，相关词汇比单个高频词汇更具有代表性，更能代表留言的内容和思想。在文本集中，任意的两个词多次出现在多个文本范围内都可被认定为共现词，词条 t1 与词条 t2 可组成共现词对（t1，t2）。

现在我们引入共现度来评价词条 t1 与词条 t2 的语义相关性。共现词对的相关性越大就越能表示该共现词对在文章中越重要。共现词对（t1，t2）的共现度计算公式如下:

$$C(t_1,t_2)=\alpha\times C(t_1\mid t_1,t_2)+\beta\times C(t_2\mid t_1,t_2)$$

其中，C（t1|t1，t2）和 C（t2|t1，t2）分别是在关键词 t1，t2 出现的条件下，共现词对（t1，t2）出现的次数，称为条件共现度。α 和 β 分别是关键词 t1 和 t2 共现度的加权参数，在共现度计算公式中起着非常重要的作用。其中条件共现度公式如下:

$$C(t_1\mid t_1,t_2)=\frac{f(t_1,t_2)}{f(t_1)}$$

$$C(t_2\mid t_1,t_2)=\frac{f(t_1,t_2)}{f(t_2)}$$

$f(t_1)$，$f(t_2)$，$f(t_1,t_2)$分别表示词 t_1出现的次数，词 t_2出现的次数和在一定文本范围内词 t_1与 t_2共同出现的次数。

基于前一步的词频统计，我们获取到词频较高的统计词，根据词共现算法构造共现矩阵。

构建过程如下：

1. 对附件 3 留言数据作词频统计，得到全部词频统计结果）

2. 对词频统计结果求并集，结果存入一个字典中，keys()为词，values()为每个词的词频。再将所有特征词存入 Full_Feature_word 列表中，其对应的词频存入 Full_Feature_weight 列表中。

3. 建一个二维矩阵 Common_matrix，其大小为： 总特征词词数 x 总特征词词数 （也就是共词矩阵）。其横竖分别对应总特征词中的每个词，例如矩阵第 3 行第 5 列的数值即代表，特征词第 3 个与特征词第 5 个的关系程度，同时它的值也等于该矩阵第 5 行第 3 列的值。

4. 将共词矩阵对角线上元素赋值为它的出现次数。

5. 循环遍历特征词列表，构建全部两个词之间的组合，再遍历每一篇文章的切词结果，如果该两个词在同一片文章中出现，则该两词的权重+1，再将其存入共词矩阵的对应位置中。

形成的矩阵形式如下：

$$A=\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{22} & \cdots & f_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ f_{N1} & f_{N2} & \cdots & f_{NN} \end{bmatrix}$$

该矩阵为一个对角线对称矩阵，其中 f_{ij} 为第 i 个关键词和第 j 个关键词出现的频次，$f_{ij}=f_{ji}$ 。

根据 Van Raan AFJ 等人提出的相关概念，共现矩阵反映的是一种绝对的表象。 因为两个关键词的共现频次是直接两个关键词各自出现的频次影响。要想真正揭示关键词之间的共现关系，还需引入关键词共现相对强度的指标，这就需要按照特定的计算公式计算关键词共现的强度。在文献计量学中，常用的表示关键词之间的关联强度的统计指数有 Jaccard 指数和 Salton 指数两种。根据统计学的原理，这两种指数都可以使两个本来关系就密切的关键词显得更密切，使两个关系疏远的关键词显得更疏远，从而将某一学科领域的核心和非核心区分开来。

对于问题二，我们选用 Jaccard 系数作为关键词指数，因为 Jaccard 使用的是集合操作，句子的向量长度由两个句子中独特的词汇数目决定，重复多次的词汇不会对其产生影响，计算较为简单快速。定义两个集合 S,T 的 Jaccard 相似度: Sim(S,T) = |S,T 的交集| / |S,T 的并集|。

2.3.5. K-means 聚类

根据上述的关键词指数量化文本之间的相关关系，然后将文本相关系数矩阵进行 k-means 聚类。

K-mean 聚类的原理如下：

假设有一个包含 n 个 d 维数据点的数据集 $X=\{x_1,x_2,\ldots,x_i,\ldots,x_n\}$，其中 $x_i\in R^d$，K-means 聚类将数据集 X 组织为 K 个划分 $C=\{c_k,i=1,2,\ldots,K\}$。每个划分代表一个类别中心 μ_i。选取欧氏距离作为相似性和距离判断准则，计算该类内个点到聚类中心 μ_i 的距离平方和

$$J(c_k)=\sum_{x_i\in c_k}\|x_i-\mu_k\|^2$$

聚类目标是使得各类总的距离平方和 $J(c)=\sum_{k=1}^{K}J(c_k)$ 最小，

$$J(c)=\sum_{k=1}^{K}J(c_k)=\sum_{k=1}^{\mathrm{K}}\sum_{x_i\in c_k}\left\|x_i-\mu_k\right\|^2=\sum_{k=1}^{\mathrm{K}}\sum_{i=1}^{n}d_{ki}\left\|x_i-\mu_k\right\|^2$$

其中，$d_{ki}=\begin{cases}1,若x_i\in c_i\\0,若x_i\in c_i\end{cases}$，所以根据最小二乘法和拉格朗日原理，聚类中心 μ_i 应该取为类别

c_k 类各数据点的平均值。

K-mean 聚类的算法步骤如下：

1、从 X 中随机取 K 个元素，作为 K 个簇的各自的中心。

2、分别计算剩下的元素到 K 个簇中心的相异度，将这些元素分别划归到相异度最低的簇。

3、根据聚类结果，重新计算 K 个簇各自的中心，计算方法是取簇中所有元素各自维度的算术平均数。

4、将 X 中全部元素按照新的中心重新聚类。

5、重复第 4 步，直到聚类结果不再变化。

6、将结果输出。

完成聚类后，将距离聚类中心簇较远的留言即为热点问题。我们将文本热度指数用留言到中心簇的距离公式来量化，再综合考虑点赞数与反对数指标，为了使不同单位或量级的指标能够进行比较和加权，我们进行归一化，并进行加权得到整体的留言热度指数，排序后获取最终的结果。

2.4. 问题三的分析方法与过程

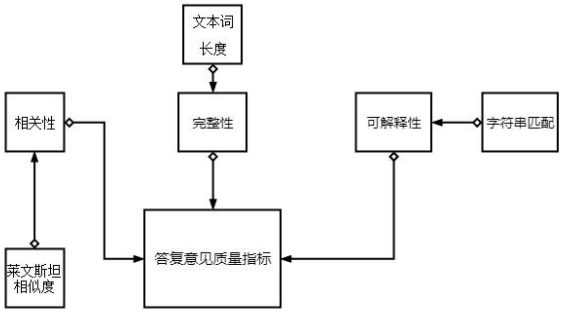


图 6：问题 3 流程图

本题要求根据部门对于留言的答复意见给出一套意见的质量评价。我们尝试从各种角度来评估答复意见的质量，主要包括从相关性，完整性以及可解释性等角度。

对于答复意见的相关性质量，我们考虑利用文本相似度计算来衡量，通过利用前文的关键词抽取算法，抽取出留言的关键词文本和答复意见文本计算余弦相似度或者是莱文斯坦相似度。其中，莱文斯坦相似度描述的是两端文本之间的形体相似性。最终利用上述相似度计算结果归一化获取答复意见与问题的相关性系数。对于答复意见的完整性评价，我们利用前文的文本分词算法，通过文本词长度统计来衡量，一般来说文本含有的词语越多，回复意见越完整。此外对于答复意见的可解释性评估，主要考虑利用字符串匹配来获取，主要是统计答复意见中高频词出现在问题中高频词的次数进行统计。最终量化出三个评价指标，然后建立熵权综合评价模型，最终给出每个答复意见的质量评价得分。

2.4.1. 莱文斯坦距离

莱文斯坦距离(LD)用于衡量两个字符串之间的相似度。 以下我们称这两个字符串分别为 s (原字符串) 和 t (目标字符串)。莱文斯坦距离被定义为”将字符串 s 变换为字符串 t 所需的删除、插入、替换操作的次数”。

在数学上，两个字符串 a,b 的莱文斯坦距离记作 $lev_{a,b}(|a|,|b|)$ 。这里

$$lev_{a,b}(i,j)=\begin{cases} max(i,j) & if\ min(i,j)=0, \\ min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i\neq b_j)} \end{cases} & otherwise. \end{cases}$$

这里，|a|和 |b| 分别表示字符串 a 和 b 的长度，$1_{(a_i\neq b_j)}$ 是当 a≠b 时值为 1，否则值为 0 的示性函数。这样，$lev_{a,b}(|a|,|b|)$ 是 a 的前 i 个字符和 b 的前 j 个字符之间的距离。

因此，我们通过莱文斯坦距离量化抽取出的留言关键词文本和答复意见文本之间的相似度，将其计算结果归一化获取答复意见与问题的相关性系数，用于衡量答复意见的相关性。

2.4.2. 熵权综合评价模型

熵最先由申农引入信息论，目前已经在工程技术、社会经济等领域得到了非常广泛的应用。熵权法的基本思路是根据指标变异性的大小来确定客观权重。

一般来说，若某个指标的信息 E_j 越小，表明指标值得变异程度越大，提供的信息量越多，在合评价中所能起到的作用也越大，其权重也就越大。相反，某个指标的信息熵 E_j 越大，表明指标值得变异程度越小，提供的信息量也越少，在综合评价中所起到的作用也越小，其权重也

就越小。

熵权法赋权步骤如下：

1. 数据标准化

将各个指标的数据进行标准化处理。

假设给定了个指标 $X_1,X_2,\ldots,X_k$，其中 $X_i=\{x_1,x_2,\ldots,x_n\}$。假设对各指标数据标准化后的值为 $Y_1,Y_2,\ldots,Y_n$，那么 $Y_{ij}=\dfrac{x_{ij}-\min(X_i)}{\max(X_i)-\min(X_i)}$

2.求各指标的信息熵

根据信息论中信息熵的定义，一组数据的信息熵

$$E_j=-\ln(n)^{-1}\sum_{i=1}^{n}p_{ij}\ln p_{ij}$$

其中

$$p_{ij}=\frac{Y_{ij}}{\sum_{i=1}^{n}Y_{ij}}$$

如果 $p_{ij}=0$，则定义

$$\lim_{P_{ij}\to 0}p_{ij}\ln p_{ij}=0$$

3.确定各指标权重

根据信息熵的计算公式，计算出各个指标的信息熵为 $E_1,E_2,\ldots,E_k$。通过信息熵计算各指标的权重：

$$W_i=\frac{1-E_i}{k-\sum E_i}(i=1,2,\ldots,k)$$

3. 结果分析

3.1. 问题 1 结果分析

数据预处理的结果如下：

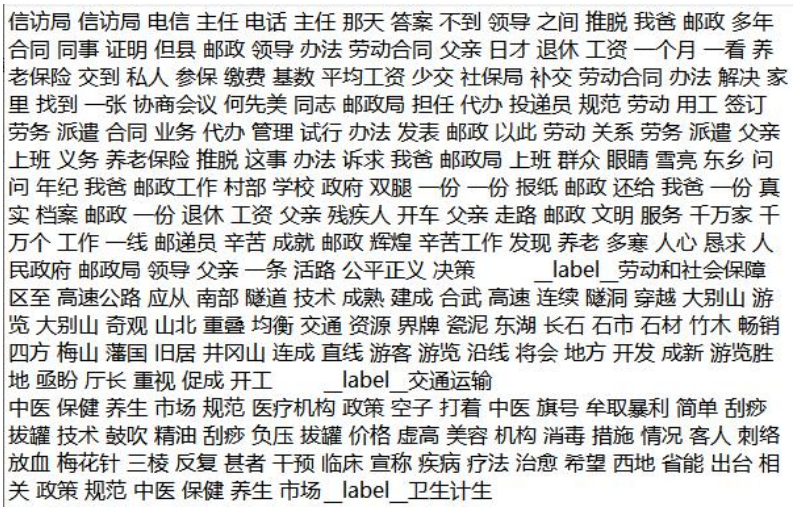


图 7：预处理后的留言内容截图

对数据进行预处理后按 8:2 划分训练集和验证集，进行 FastText 训练得到模型，进行验证，利用 F-Score 对分类方法进行评价，得到的结果如下：

其中 $F_1=\frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$，是准确率和召回率的调和平均值，最佳值为 1，最差值为 0。我们得到的模型 F_1 值为 0.97，说明组合分词处理后的数据训练的模型具有良好的分类性能。

3.2. 问题 2 结果分析

根据前文叙述的三个指标，即关注度、文本热度以及点赞数和反对数，分别取出前两个个量化指标的前 20 个数据，具体可参照附件中 result_attention.csv， result_heat.xlsx 文件。再结合点赞数对其进行归一化加权处理，排序后得出以下热点问题表和详情表，具体可参照附件中同名文件。我们可以看出：只关注单个留言是“A 市 A5 区汇金路五矿万境 K9 县存在一系列问题”热度最高；但前五个留言有三个是反映了“58 车贷案”，综合起来该问题热度最高；最后是“金毛湾配套入学的问题”。

表 1：热点问题表

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	0.93	2019/8/19 11:34	A 市 A5 区汇金路	A 市 A5 区汇金路五矿万境 K9 县存在一系列问题
2	2	0.87	2019/2/25 9:58	几十位 58 车贷受害者	严惩 A 市 58 车贷特大集资诈骗案保护伞
3	3	0.81	2019/4/11 21:02	金毛湾 2800 户业主	反映 A 市金毛湾配套入学的问题

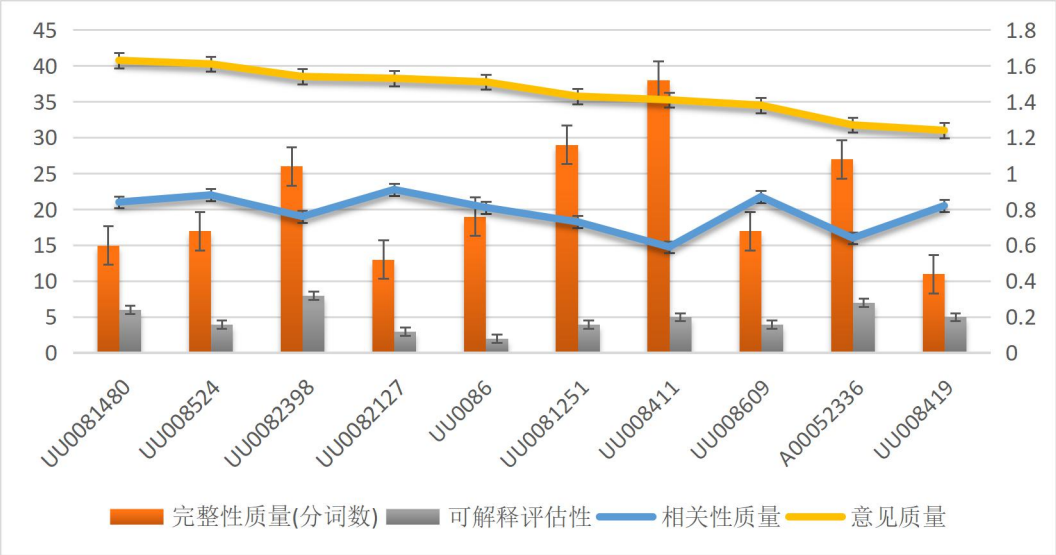
4	2	0.73	2019/2/21 18:45	A 市 A4 区	请书记关注 A 市 A4 区 58 车贷案
5	2	0.69	2019/3/1 22:12	58 车贷案	承办 A 市 58 车贷案警官应跟进关注留言

表 2：热点问题留言明细表

问题 ID	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
1	208636	A00077171	A 市 A5 区汇金路五矿万境 K9 县存在一系列问题	2019-08-19 11:34:04	我是 A 市 A5 区汇金路五矿万境 K9 县 24 栋的一名业主…	0	2097
2	217032	A00056543	严惩 A 市 58 车贷特大集资诈骗案保护伞	2019-02-25 09:58:37	胡市长：您好！西地省展星投资有限公司设立 58 车贷…	0	790
3	223297	A00087522	反映 A 市金毛湾配套入学的问题	2019-04-11 21:02:44	书记先生：您好！我是梅溪湖金毛湾的一名业主…	5	1762
2	220711	A00031682	请书记关注 A 市 A4 区 58 车贷案	2019-02-21 18:45:14	尊敬的胡书记：您好!A4 区 p2p 公司 58 车贷，非法经营近四年…	0	821
2	194343	A000106161	承办 A 市 58 车贷案警官应跟进关注留言	2019-03-01 22:12:30	胡书记：您好！58 车贷案发，引发受害人举报投诉…	0	733

3.3. 问题 3 结果分析

本题要求根据对于留言的答复意见给出一套意见的质量评价，我们从文本相似度量化相关性，即莱文斯坦相似度归一转换为相关性系数；通过统计答复的文本词长度统计衡量完整性；用字符串匹配统计留言和答复意见出现的高频词次数，衡量回复的可解释性。通过建立熵权综合评价模型，最终给出每个答复意见的质量评价得分。下面是答复意见的质量排名前十的图，详细数据可见附件中文件：排名前十质量问题回复.xlsx。



在我们评价模型中，三个指标组成意见质量的高低，缺一不可，一般来说，有两个指标较高的，意见质量评估得分就较高，相关性质量相较于其他两者，与意见质量关联性较大，一般意见质量高的，相关性就比较高。

4. 结论

网络问政平台作为政府和群众之间的"民意连心桥"，我们通过建立了一个智慧政务系统，高效准确地对平台的群众留言进行分类，发现热点问题和对留言答复的评价，有助于提升政府的管理水平和施政效率。

本文采用组合分词，改进以往只用 jieba 分词的方法，采用 TF-IDF 算法进行关键词提取，利用 Fasttext 文本分类框架构建分类模型,实现有监督训练的文本分类,提高了准确率,使用 F-Score 对分类方法进行评价，得到 $F_1 = 0.97$ 。

对于留言热度指数的衡量，通过三个不同的角度得到对应的热点问题，再将其归一加权得到最终的排名前 5 的热点问题及详细信息。对于答复意见的评价，我们从相关性，完整性以及可解释性的角度制定一套评价方案。然后建立熵权综合评价模型，对每个答复意见的质量进行评价。通过对处理后的数据进行查看，发现热点问题和评价模型的适用性较好。

随着大数据、 云计算、人工智能等新技术的发展，今后在这方面的研究前景会更广阔的，在推动政府的管理水平和施政效率的发展和进步。

5. 参考文献

[1]和志强,王丽鹏,张鹏云.基于词共现的关键词提取算法研究与改进[J].电子技术与软件工程,2018(01):144-146.

[2]郭树行, 谈斯奇. 关键词共现研究趋势分析[J]. 科技资讯, 2011(32):210-211.

[3]CSDN 博主「逆水舟行」https://blog.csdn.net/u013782172/article/details/78345675

[4]CSDN 博主「悟乙己」https://blog.csdn.net/sinat_26917383/article/details/54850933

[5]fastText 原理和文本分类实战，看这一篇就够了

https://blog.csdn.net/feilong_csdn/article/details/88655927

[6]CSDN 博主「feilong_csdn」原文链接：https://blog.csdn.net/feilong_csdn/article/details/88655927

[7]CSDN 博主「这是一个死肥宅」原文链接：

https://blog.csdn.net/qq_28840013/article/details/89575548

[8]Jaccard 相似度、minHash、Locality-Sensitive Hashing(LSH)

http://www.voidcn.com/article/p-qfljdngb-s.html

[9]Jaccard 与 cosine 文本相似度的异同 - mountain blue 的文章 - 知乎

https://zhuanlan.zhihu.com/p/60723017

[10]莱文斯坦距离

https://baike.baidu.com/item/%E8%8E%B1%E6%96%87%E6%96%AF%E5%9D%A6%E8%B7%9D%E7%A6%BB/14448097?fr=aladdin

[11]【综合评价方法 熵权法】指标权重确定方法之熵权法

CSDN 博主「开心果汁」

https://blog.csdn.net/u013421629/article/details/81221559