“智慧政务”的文本挖掘

摘 要

因为微信、微博、市长信箱、阳光热线等线上平台逐步实现了网络问政的功能，且使用人数日益增长，成为政府了解民意，解决民生问题的重要渠道，数据量的攀升也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，所以对政务的文本挖掘变成了一个课题，本文主要讲述了实现基于自然语言处理技术的智慧政务系统的实现过程。

针对问题一，首先将读取出的 csv 文件进行数据清洗，运用 unique 函数类属性算法删除相邻的重复元素，将 type 变量转换为因子来进行因子分析，即提取出分类变量，本次实验共有七个分类标签。然后清洗文本数据，对文本分词处理，进行词频统计并降序排序，建立训练集和测试集数据，通过建立文档——词频矩阵和降维来创建指示特征，运用 TFIDF 指标算法进行特征提取，通过制作总词云图来进行可视化处理。最后绘制聚类图，进行聚类分析，再用朴素贝叶斯算法进行分类，可以得到预测精度，再来评估模型的性能和优化。

针对问题二，首先用正则语法匹配留言主题所涉及的地名，并写入地名表，根据地名表用 excle 的文本筛选功能将反馈同一地区的实例归纳在一起，再给语料做文本预处理，使用 jieba 模块和哈工大停词表，给留言详情分词去停用词。然后制作 idf 语料库，用 jieba.analyze 模块的关键词提取，提取重要且有意义的关键词，作为该文档代表。最后利用 lda 模型给这些代表文档主题的关键词集分类，得到该文档主题。

针对问题三，我们分别从相关性、完整性、可解释性来处理。对于相关性，运用 textreuse 就可以比较方面的实现简单的机械分词，只是把文档的内容去掉噪音+分开成字符串。接下来通过 minhash 算法利用 Jaccard 计算距离的方式，计算两个文本之间的相关度。至于可解释性，采用条件式变分自编码器（CVAE）+文本分类生成性解释框架（GEF）的模式，在 CVAE 上应用 GEF，来生成有不同情感倾向（积极、消极和中立）的解释，再用 BLEU 分数对生成的文本解释进行评分，就能得到最终的可解释度。

关键词：朴素贝叶斯算法，LDA 模型, TF-IDF 模型，GEF

Abstract

Because WeChat, such as weibo, mayor mailbox, sun hotline online platform gradu ally realized the function of network asked administration, and the use of the number is growing, as the government to understand public opinion and the important channel to s olve the problem of the people's livelihood, the amount of data up to the past also rely mainly on artificial to leave a message and hot spots of relevant departments work has brought great challenge, so the government affairs text mining has become a subject, this article mainly tells the story of implementation is based on natural language proces sing technology and the implementation process of the wisdom of the e-government sys tem.

To solve the problem one, we first clean the read out csv files, use the unique functi on class attribute algorithm to delete the adjacent repeated elements, and convert the typ e variables into factors for factor analysis, that is, extract the classified variables. There are seven classification labels in this experiment. Then the text data is cleaned, word seg mentation is processed, word frequency statistics is carried out and sorted in descending order, training set and test set data are established, indicating features are created by est ablishing document — word frequency matrix and dimension reduction, feature extracti on is carried out by using TFIDF index algorithm, and visual processing is carried out b y making total word cloud map. Finally, we draw the clustering diagram, carry on the cl ustering analysis, and then use the naive Bayesian algorithm to classify, we can get the p rediction accuracy, we get the accuracy is 80%, and then evaluate the performance and o ptimization of the model. The final accuracy of this experiment is.

To solve the problem two, we first use regular grammar to match the place names i nvolved in the topic of the message, and write the place names table. According to the e xcle text screening function of the place names table, the examples of the same area are summarized together, then the corpus is preprocessed, the jieba module and the stop wor ds table are used, and the word segmentation of the message details is given to stop wor ds. Then we make the idf corpus, extract the key words of the jieba.analyze module, and extract the important and meaningful key words as the representative of the document. Finally, the lda model is used to classify the keyword sets that represent document topic s, and the document topics are obtained.

In response to question three, since three perspectives have been given in the title t o evaluate the response, we deal with it separately in terms of relevance, completeness a nd interpretability. For correlations, use textreuse to compare aspects of the implementat

ion of simple mechanical participles, only to remove the contents of the document noise into strings. Next, the correlation degree between the two texts is calculated by using th e minhash algorithm using Jaccard to calculate the distance.Hamming Distance can be u sed to measure the distance between two strings. JaccardCoefficient is used to measure the similarity of two sets, and then the correlation degree between the two is calculated by using the Jaccard calculation distance formula. with regard to interpretability, the fin al interpretability can be obtained by using the model of conditional variational selfenco der (CVAE) text classification generative interpretation framework (GEF), applying GE F, on the CVAE to generate explanations with different emotional tendencies (positive, n egative and neutral), and then scoring the generated text interpretation with a BLEU sco re.

Keywords: Naive Bayes algorithm, LDA model, TF-IDF model.GEF

目录

一，挖掘目标

1.1 挖掘背景

近年来，科技飞速发展，软件和网站的功能日益完善，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和解决民生问题的重要渠道，各类社情民意相关的文本数据量攀升，以人工来进行留言划分和热点整理的相关部门工作上面临巨大的挑战。与此同时，大数据、云计算、人工智能等技术的高速发展给了我们新的方向，基于自然语言处理技术的智慧政务系统应运而生，极大提升了政府的管理水平和施政效率。

面对繁多且涉及管理部门不一的留言，我们首先要对其做一个分类，以便相关部门能够更有效的处理相关留言，要做到专门的人处理专门的事。其次，留言问题过多，就有可能存在大量问题重复发生或者很多人关心，我们就要对这些提及次数很多的问题做一个汇总，即任务二的热点问题挖掘。等到这些留言被相关部门处理进行回复后，我们也关心这些问题是否得到有效解决，于是再对答复意见进行评价。

1.2 挖掘目标

如上文所说，我们就是在实现智慧政务系统的一些功能。本文是对附件里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来进行文本挖掘处理。我们主要是要解决大量群众留言的分类问题、挖掘留言中在某一时段群众集体反映的一些热点问题以及从相关性、完整性、可解释性等角度来对相关部门针对留言的答复作出评价。

二，问题分析

2.1 群众留言分类的分析

目前大部分电子政务系统还是依靠人工根据经验处理，这样的问题是工作量大且效率低下，我们的想法就是建立一个关于留言内容的一级标签分类模型。

首先对提取的留言进行因子分析来提取出分类变量，然后对文本数据做一个数据清洗，进行分词处理和词频统计，再创建指示特征来进行特征提取。最后绘制聚类图，进行聚类分析，用朴素贝叶斯算法进行分类。

2.2 热点问题挖掘的分析

这个任务里，首先就是要自己制定一个决定热点问题的标准，即在某一时间段里同一地区的群众反映次数更多的问题。

我们的思路是将留言主题涉及到所有的地名制成地名表，先选出同一地区的留言，对这些留言进行分词和去停用词处理，分别得到关键词作为该留言的代表，再利用 LDA 模型给这些留言分类并计数，得到提及次数的排序，最后再通过时间来筛选出热点问题。

2.3 答复意见评价的分析

题目要求里给出了相关性、完整性、可解释性三个角度来评价答复意见，我们也是围绕着这三个方面来处理。

关于相关性，运用 textreuse 就可以比较方面的实现简单的机械分词。接下来通过 minhash 算法利用 Jaccard 计算距离的方式，计算两个文本之间的相关度。海明距离可以用来度量两个串的距离， Jaccard 系数用来度量两个集合的相似度，再利用 Jaccard 计算距离公式计算出两者之间相关度。

完整性，

可解释性，我们自己的理解就是我们能理解这个文本的程度，是否能轻松的获知文本信息的感情色彩和关键信息等。我们的具体思路就是对于一个文本，我们可以从中提取一些关键词作为评价因素，将这些因素放入可解释性模型，来得到可解释性。我们在这里主要用到了条件式变分自编码器、文本分类生成性解释框架和 BLEU 分数。

三，数据预处理

在实现以上的三个问题的功能中，有一个操作经常被提及且非常重要---对数据的预处理，其中包括数据清洗、去停用词和数据分词处理。

3.1 数据清洗

在数据清洗阶段，分为三级清洗：去标点，去内容，去不用词。去标点的具体操作为：①. gsub 函数用于字符串替换，会替换所有满足条件的匹配，此时清洗用字符替换函数，去空格；②. 有时字符之间隔开需要使用\\\n ，用\\\n 来隔开；③. 文中有英文逗号会报错，所以用大写的“，”来代替；④. 替换波浪号（~）和英文单引号（’），它们之间用“|”符号隔开，表示或的关系；⑤. 替换所有的英文双引号（″），因为双引号在 R 中有特殊含义，所以要使用三个斜杠（\\\）来转义。

至于去内容的操作就是：①. 文本内容转化为向量 sentence；②. 清除数字;③. 清除英文字符[a-zA-Z];④. 清除全英文的 dot 符号;⑤. 清除对应 sentence 里面的空值（文本内容）;⑥. `nchar`函数对字符计数，其中，英文叹号为 R 语言里的“非”函数。

最后再来对不用词进行清理：①. 剔除特殊词（剔除一些不需要的词）得出 res；②. 清理文本里的回车，否则每个回车就会被识别成一段文本。

3.2 去停用词

首先在网上下载一个停用词表 stopwords，运用 removeStopWords 的功能，之后利用 lapply 函数，循环处理列表中的每一个元素，遍历每一个分出的词，如下：word < -1apply(word, removeStopWords, stopword)，来去除所有停用词。之后将去除的所有词，一个词构成一个向量形式，有助于后面对数据的操作。

3.3 数据分词处理

分词处理，也就是将文本拆分为词语。利用 Rwordseg 算法对文本进行分词处理。Rwordseg 是一个 R 环境下的中文分词工具，使用 rJava 调用 Java 分词工具 Ansj。Ansj 也是一个开源的 Java 中文分词工具，基于中科院的 ictclas 中文分词算法，采用隐马尔科夫模型（Hidden Markov Model, HMM），进行分词。ICTCLAS 分词关键技术，ICTCLAS 和别的分司系统不一样的地方就是于--N 最短路径分词算法。所谓 N 最短路径其实就是最短路径和最大路径的折中，保留前 N 个最优路径。在 N－最短路径求解之前，ICTCLAS 首先通过二叉分词图表（如下图示）表示出了每个词组之间的耦合关系，每一个节点都表示分词图表中的一条边，它的行值代表边的起点（前驱），它的列值代表边的终点（后驱）。最终通过计算词组之间的耦合关系，来最终确定分词路径。

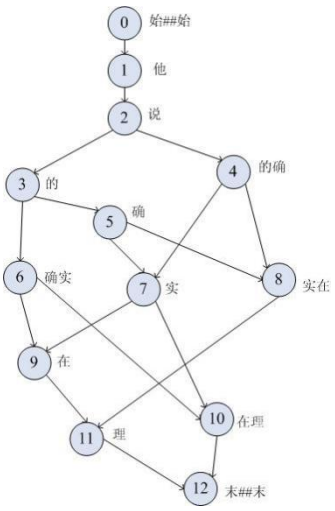


图 3-3-1 二叉分词图表

四．模型假设

4.1 以朴素贝叶斯算法建立分类模型

在处理问题一群众留言问题分类时，我们尝试建立一个分类模型来处理这种

短文本的分类。最终我们采用了朴素贝叶斯算法，朴素表示所有词的特征独立，然后就要在给定词比例的基础上，求各类型文档的比例。先创建分类器，然后进行预测，接下来生成实际与预测交叉表，最后预测精度。

4.2 LDA 模型来挖掘热点问题

性判别式分析(Linear Discriminant Analysis, LDA)，也叫做 Fisher 线性判别(Fisher Linear Discriminant ,FLD)，LDA 模型是词袋模型，不考虑词语上下文的联系。通过对文章的主题分布，词语的主题分布，多次迭代选取该文档主题分布中，概率最大的那个主题。

五．问题求解

5.1 群众留言分类

我们的主要思想是采用出现频率高且有意义的词作为该主题的一系列相关单词。同时根据出现频率给予他们不同的权重。在这些词的权重处理，分为两部分，一个是留言主题，一个是留言详情，由于留言详情的信息量更大，描述得更加具体，所以我们主要是对留言详情进行分类。重复上诉步骤分别统计出七个一级标签的一系列相关词。

其中需要我们重视的一点是，一个短文本里可能会出现多个主题，一个词也可能对应多个主题。这里有个重要概率，单词对于主题的概率和主题对于文本的概率，概率大小有出现频率和词的权重决定。知道这些后我们就可以根据概率的大小判断一个短文本里某个词对应的多个主题，这个词在某几个主题里对应的概率又不同。统计分析，某个主题的系列相关词在某文本里出现的频率，选取频率最高的主题确定为该文本主题。

5.1.1 收集数据

首先，将读取出的 csv 文件进行数据清洗，运用 unique 函数类属性算法，unique 算法表示删除相邻的重复元素，然后重新排列输入范围内的元素，并且返回一个迭代器（容器的长度没变，只是元素顺序改变了），此时表示出来的为无重复的值。

其次，将 type 变量转换为因子，进行因子分析（提取出多个变量潜在的公共因子），在此处即提取出共有哪些个分类变量，我们这里共提取出 7 类：

5.1.2 探索、处理、准备和分析数据

这里开始还是要对文本数据进行预处理，清洗文本数据，去除停用词和分词处理，这里的分词可以自定义词典可以网上下载，也可以手动添加词库，每个词一行写到.txt 文件上，调用 installDict()添加即可（以防删错词），接下来再进行分词+构建数据集（利用 rJava 和 Rwordseg）：①. 进行分词同时可以对分出来的词进行频数统计，以便后续运行；②. 利用 table 函数统计列表中的词频次，遍历列表向量内的每个元素，并且使用指定函数来对其元素进行处理返回列表向量，然后

降序排列，最后利用 data.frame 函数产生 frame 数据集，包含名称 word 和频次 freq 两列；③. 对分出来的数据集进行降序排序，最上方的是频数最大的；④. 最后从表格中过滤掉 1 个字的词 其中 Var1 是分类变量，Freq 是数值型变量。进而为数据建立训练集和测试集数据，将数据控制训练集与测试集之比为 3:1（随机无放回抽取 3/4 样本）。

然后，为频繁的单词创建指示特征：①. 建立文档——词频矩阵：首先在统计 TFIDF 等指数之前，要先处理数据，因为在分词的时候分出了空白符，这种空白符即不能用 is.na、is.null、is.nan 这些函数查出来，也不能使用常见的空白符（空格″″，制表符″\t″，换行符″\n″，回车符″\r″，垂直制表符″\v″，分页符″\f″）包括空白符（″\\s″）等正则规则查出来，然后利用（tm）建立语料库（cropus），前要加载出 NLP 自然语言处理包，利用 Corpus（VectorSource（））对其建立预料。

接下来转换为文档——词频矩阵；然后依次将训练集、测试集转换为数据框；②. 降维（词太多了，不好聚类，所以需要降维，就是减少词的数量，把不重要的词剔除）：首先查看原来有多少词；接下来降维，去除了低于 99% 的稀疏条词，并进行反复测试观察是否符合稀疏度；然后可以再次查看降维后剩下多少词，此时，Sparsity 为 33%。（经过一定的降维这是所能达到的最小的词量）：

紧接着我们可以利用 findFreqTerms 函数查找矩阵中的高频词，也可以利用 findAssocs 函数找到与某个单词具有一定相关系数的单词。

进而特征提取（运用 TFIDF 指标算法）：FIDF 的主要思想是：若某个词或短语在一篇文章中出现的频率 TF 高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。TFIDF 实际上是：TF * IDF，TF 词频(Term Frequency)，IDF 反文档频率(Inverse Document Frequency)。TF 表示词条在文档 d 中出现的频率。IDF 的主要思想是：如果包含词条 t 的文档越少，也就是 n 越小，IDF 越大，则说明词条 t 具有很好的类别区分能力。如果某一类文档 C 中包含词条 t 的文档数为 m，而其它类包含 t 的文档总数为 k，显然所有包含 t 的文档数 n=m + k，当 m 大的时候，n 也大，按照 IDF 公式得到的 IDF 的值会小，就说明该词条 t 类别区分能力不强。所以我们首先要得到归一化数据后的 TF-IDF 矩阵，然后分别计算 tf、idf。利用 TF 公式计算出该矩阵中单词的 tf 量，分别计算分子每个单词在矩阵中的出现次数，而分母则是矩阵中所有词的出现次数之和，做比。得到 tf 矩阵。利用 IDF 公式计算出矩阵中单词的 idf 量，其中分子为语料库中的单词总数，分母计算包含每个单词的矩阵数目，如果该词语不在语料库中，就会导致被除数为零，然后对计算出的分数做 log 处理，得出 idf 矩阵。再利用 diag 计算得出 idf 矩阵的 diagonal matrix 对角矩阵。最后计算 tf_idf = tf * idf 矩阵（这里的规范化是按“行”计算的）。

然后进行可视化处理，我们这里都是利用 RColorBrewer 中的 wordcloud 制作出词云图。

5.1.3 基于数据训练模型

先要绘制聚类图（先进行标准化处理，再生成距离矩阵，再用层次聚类）：首先利用 as.data.frame 转换分析数据为数据框结构，然后利用 dist 函数，对数据进行计算欧氏距离（此处欧氏距离是最易于理解的一种距离计算方法，源自欧氏空

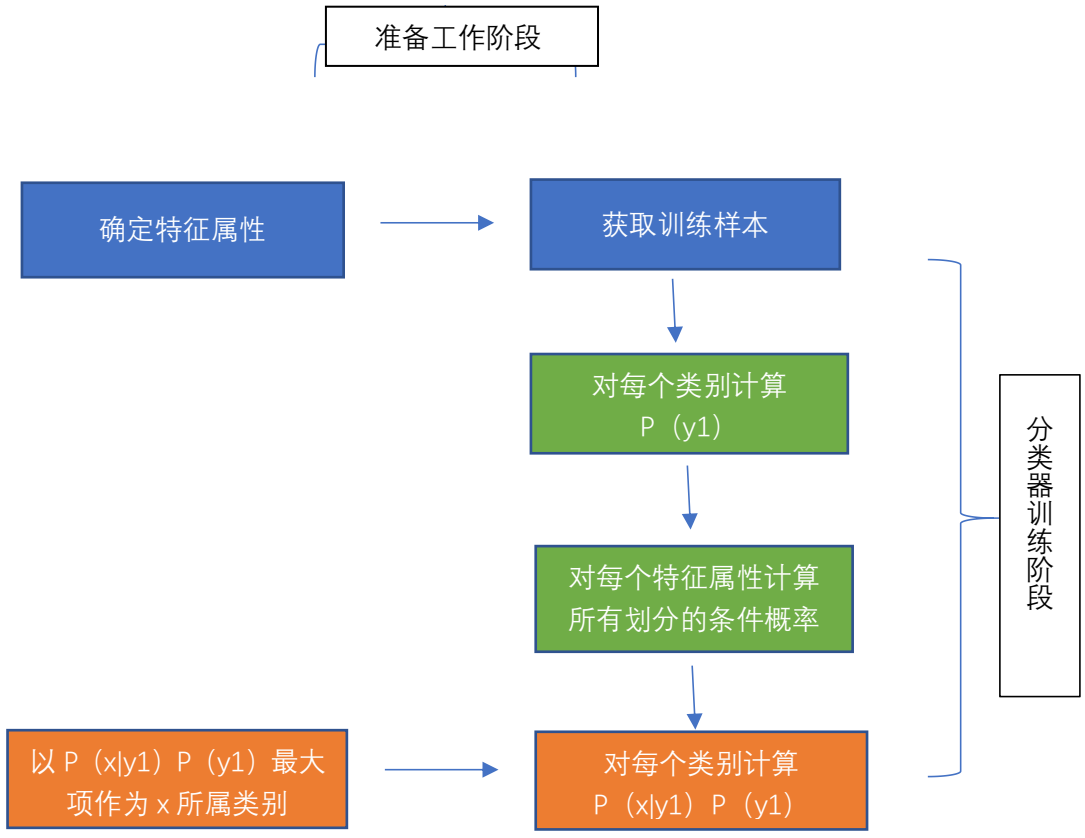
间中两点间的距离公式）。二维平面上两点 a(x1,y1)与 b(x2,y2)间的欧氏距离：利用 $d_{12}=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$ 带入计算得出的 tdm_removed 矩阵，进行计算矩阵中每个单词到中心单词（频数较高单词）的距离。然后进行聚类分析，绘制聚类图（用一张图展示聚类的结果）。

然后我们就来建立模型，处建立模型用到了朴素贝叶斯分类模型及算法，贝叶斯方法是以贝叶斯原理为基础，使用概率统计的知识对样本数据集进行分类，其中，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，先通过前文已分类给定的（train_data），以特征词之间独立作为前提假设在给定词比例的基础上，求各类型文档的比例。需要创建分类器： 其中 train 为包含训练数据集的数据框或矩阵；class 为包含训练数据每一行的分类的一个因子向量此处建立模型假设为分类器训练阶段，这个阶段的任务就是生成分类器利用 naiveBayes(uncsv$Type, train_data)，主要工作是计算每个类别在训练样本中的出现频率及每个特征属性划分对每个类别的条件概率估计，并将结果记录：

已知 uncsv$Type, train_data，

计 算 p（train_data | uncsv$Type）p（train¬_data）p（uncsv$Type） 和 St.p（uncsv$Type | train_data）= p(train_data | uncsv$Type) * p(uncsv$Type)/p(train_data)

详细过程如下图解：



应用阶段

接下来就到了最重要的一步，建立分类模型，也就是应用阶段。这个阶段的任务是使用上述建立好的分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。这一阶段也是机械性阶段，由程序完成。在此，检验叙述分类实现：首先利用（e1071）包构建预测模型：需要讨论的问题是当P（train_data | uncsv$Type）= 0时的情况，当某个类别下某个特征项划分没有出现时，就是产生这种现象，这会令分类器质量大大降低。在此引入 Laplace 校准，即对没类别下所有划分的计数加 1，这样如果训练样本集数量充分大时，并不会对结果产生影响，并且解决了上述频率可能为 0 的问题。在建立好分类器后，对 test 生成混淆矩阵进行预测分类预测，呈现算法性能的可视化效果，运用监督学习。其每一列代表预测值，每一行代表的是实际的类别。这个名字来源于它可以非常容易的表明多个类别是否有混淆（也就是一个 class 被预测成另一个 class）。模型建立中，test 做预测值训练数据集，其中 m 为函数 naiveBayes（）训练的一个模型 m 为 naiveBayes(train, class, laplace = 0) ，type 为标识预测是最可能的类别值或原始预测概率值。接下来生成实际与预测交叉表，是矩阵格式的一种表格，显示变量的（多变量）频率分布。此处交叉表提供了两个变量之间的相互关系的基本画面，方便后续操作和发现它们之间的相互作用。最后通过将交叉表矩阵对角化求和，与交叉表求和计算出分类模型的失误率，进行修改得出预测的精度。此处，我们得到的失误率为 0.1231148，可计算出此处，运用朴素贝叶斯分类算法得到的预测精度为 80%。

5.2 热点问题挖掘

我们利用两种机器学习算法，对附件三的数据，进行分类。根据提供的留言详情，反对数，赞成数，整理规划出一套热度评价指标。根据指标对分类出的热点问题进行排序，得到排名前五的热点问题。

5.2.1 对地点筛选

这里我们主要用到了 Excel 工具，用正则语法匹配涉及留言主题所涉及的地名，并写入地名表，根据地名表用 Excel 的文本筛选功能将反馈同一个地区的的实例归纳一起。

5.2.2 统计每个主题留言次数

给语料做文本预处理，主要用到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) 、基于 TF-IDF 算法的关键词抽取和基于 TextRank 算法的关键词抽取等算法来更准确的分词处理，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。为

了提高待处理语料库的精度，我们通过使用哈工大的停用词表，同时多次使用 jieba.analyze 里的关键词提取算法，查看是否有无效关键词，逐渐完善适用于政务问题分类的停用词表。

在对留言详情的关键词提取时，如果某个词比较少见，但是它在这篇文章中多次出现，那么它很可能就反映了这篇文章的特性，正是我们所需要的关键词。基于前面预处理过后的语料，使用的 TFIDF 算法对单个处理过后的留言详情提取关键词。关键词是能反映该文档特征的词，TFIDF 算法提取的就是这样的词。首先统计语料库里各词语在单个留言详情里出现的次数 ，就是词频 TF,但留言详情里的词数并不一样，为了便于比较，我们将词频标准化。然后计算逆文档频率缩写 IDF，将通过公式计算得到的逆文档频率写入 TXT 文件（如下图），最后计算 TFIDF。

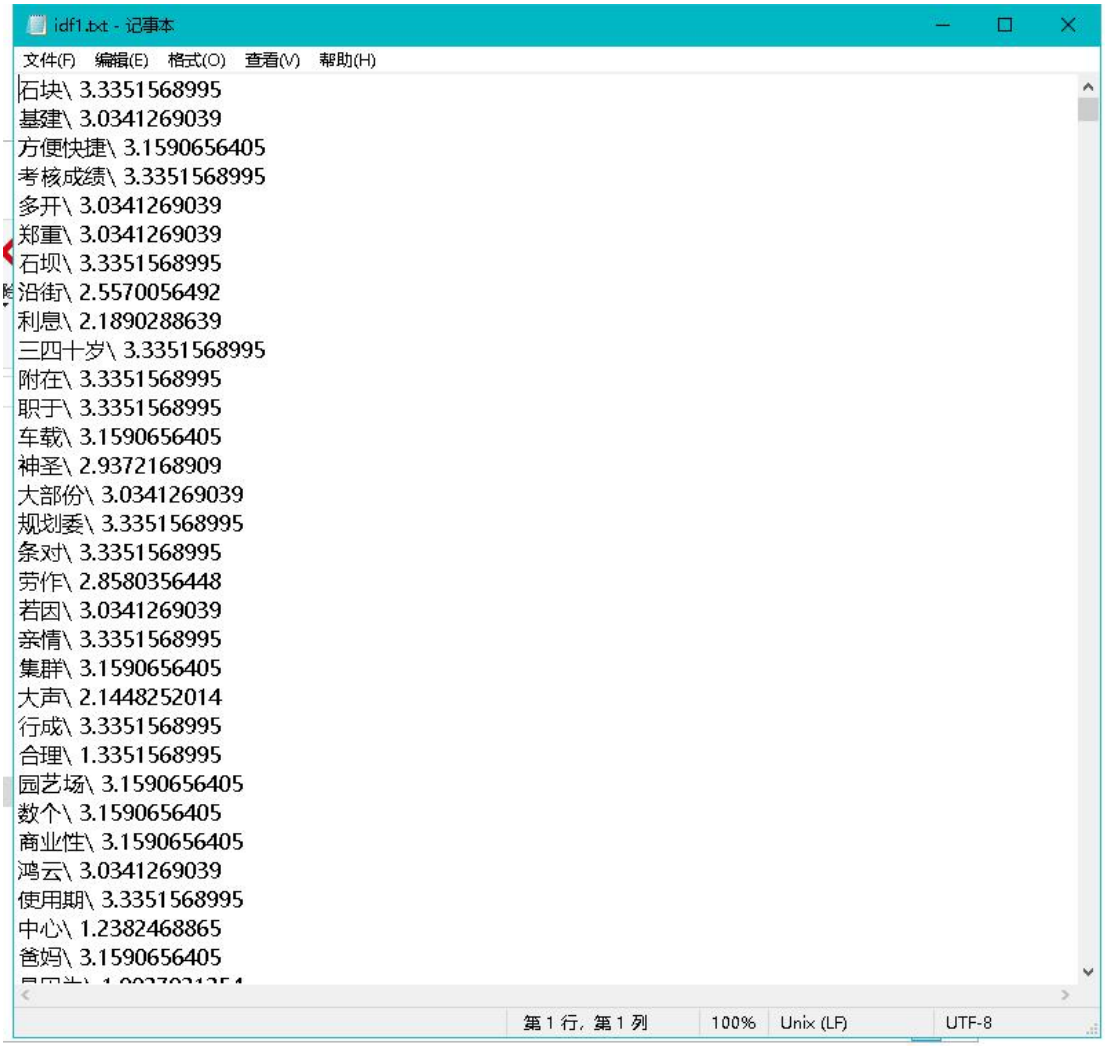


图 5-2-2-1 IDF 图

将 TFIDF 值排名前 20 的打印出来，同时有些意义不大的词成为了关键词。对于分类没有帮助，还会导致分类结果过拟合，这里我们将这些词加入停用词表。再次重复预处理和提取关键词的步骤，同时将关键词整理成语料。

最后就用到了 LDA 模型，也就是线性判别式分析模型。对每篇文档，在主题分布中抽取一个主题，对抽到的主题所对应的单词分布中随机抽取一个单词，重复上述过程直至遍历整篇文档中的每个单词。LDA 模型是词袋模型，不考虑词语上下文的联系。通过对文章的主题分布，词语的主题分布，多次迭代选取该文档主题分布中，概率最大的那个主题。由于留言详情反映的问题多种多样，涉及地区范围广，导致主题模型分类不好确定主题数。我们先尝试设置主题数为 3 ，但同一类的主题词明显涉及多个范围。如：学校， 街边， 消费，公司，人群，规划 。这些词语让人很难看出该主题到底所属哪一个政务问题。使用困惑度评估模型分类结果。

测试文本集中有 M 篇文本，对词袋模型里的任意一个单词 w，$P(w) = (z\, p(z|d) * p(w|z)$，即该词在所有主题分布值和该词所在文本的主题分布乘积。模型的 perplexity 就是 $exp^{-(\sum log(p(w))) / (N)}$，（$\sum log(p(w))$ 是对所有单词取 log（直接相乘一般都转化成指数和对数的计算形式），N 的测试集的单词数（不排重）困惑度结果也有 200+，我们决定调整主题数，得到进一步细分结果。将主题数调为 10，并将有代表意义的词在 IDF 频率文档里将该词频率增大，依此增大该词对于某一主题的权重。运行结果显示主题词展现的政务问题范围在缩小，topic_0 可以推断是火灾，爆炸等安全隐患问题。再将文档主题矩阵通过 Transform 函数打印出来。反向对应得到每个文档属于哪一个主题。将同一主题的文档语料归纳到一个列表再次用 LDA 模型分类，最后得到热点问题。

5.2.3 对时间筛选

由于给出的数据时间跨度比较大，不好分出时间段，我们决定在排出提及次数比较高的留言后，再根据时间来筛选。这里也是用 Excel 来处理的。

5.3 答复意见的评价

对于评价答复意见，我们也是根据题目所给出的相关性、完整性和可解释性三个方向来进行研究。

5.3.1 相关性

相关性分析是指对两个或多个具备相关性的变量元素进行分析，从而衡量两个变量因素的相关密切程度。在我们的这个题目里，就是要计算出留言详情和针对留言详情的答复意见的相关度，从而得到答复意见的可采纳度。

R 语言中运用 textreuse 包来做文本的相关性比较，此处因为是文本机械相似关分析，那么对分词没有那么高的要求，要求分成单个字符串的形式就可以满足要求了。所以，运用 textreuse 就可以比较方面的实现简单的机械分词，只是把文档的内容去掉噪音+分开成字符串。

将留言详情和答复意见均运用机械分词 tokenize_words 函数可以实现，同时能把一些词提取出来，并去掉了标点，而且速度较快。利用如下代码机械分词：

a < - tokenize_words(paste(liuyan))
b < - tokenize_words(paste(dafu))接下来通过 minhash 算法利用 Jaccard 计算距离的方式，计算两个文本之间的相关度：

1. Hamming Distance（海明距离）

（1）Hamming Distance 可以用来度量两个串（通常是二进制串）的距离，其定义为这两个二进制串对应的位有几个不一样，那么海明距离就是几，值越小越相似。例如 x=1010，y=1011，那么 x 和 y 的海明距离就是 1。又如 x=1000，y=1111，那么 x 和 y 的海明距离就是 3。

2.Jaccard Coefficient（Jaccard 系数）

（2）Jaccard Coefficient 用来度量两个集合的相似度，设有两个集合和，它们之间的 Jaccard Coefficient 定义为： $s = \frac{|S1 \cap S2|}{|S1 \cup S2|}$ 且 s 值越大越相似。

例如 $S1 = \{A, B, C\}$ $S2 = \{B, D\}$ 则 $s = \frac{1}{4}$。

（3）运用此算法，其中 a 代表留言详情分完词后的集合 s1，b 代表答复意见分完词后的集合 s2，二者利用 Jaccard 计算距离公式计算出两者之间相似度为 0.8153736：

（4）textreuse 包中同样有其他的一些距离，可以来运算一下：
比如两个文件的相差程度等于 0.1846264：

其中 bag-similarity 是相似性的一种，比如有两个 bags{a,a,a,b}和{a,a,b,b,c}，它们的 bag-similarity 就是 1/3，在交集中，a 出现 2 次，b 出现一次，所以它的大小是 3。两个 bags 的并的大小为两个 bags 的大小的和，在这个例子中是 9，所以就为 1/3.这两个文件的 bag-similarity 应为 0.4412051（最大为 0.5）。 最后还可以计算一下在 ab 集合的交集/a 总数为多少，此时为 0.8154065，相关性比较即可结束，也就是相关度为 0.815。

5.3.2 完整性

完整性主要是答复意见针对留言详情的完整程度，在我们的想法里，主要是看留言详情所提出的问题，答复意见有没有全部提及到。据此问题思考的研究方向是，运用 tf-idf 提取出留言详情的关键词，即分词后对分出的词进行频数统计，频数越高的词被定为关键词，同时对答复意见进行一个关键词提取。再而对留言详情和答复意见的关键词进行相似度分析，就能得到留言详情和答复意见的完整度。我们这里得到的结果为 0.815.

5.3.3 可解释性

对于可解释性，我们目前只是有了一些思路。事实上在我们最开始尝试理解可解释性这个概念的时候，我们就发现似乎可解释性并没有一个确定且大众都默认的定义，而且关于可解释性的模型也没有很多，许多人提出，可解释性就是对模型产生信任的方式，也有些人认为可解释的模型就是可取的。我们自己的理解就是我们能理解这个文本的程度，是否能轻松的获知文本信息的感情色彩和关键信息等。我们的具体思路就是对于一个文本，我们可以从中提取一些关键词作为评价因素，将这些因素放入可解释性模型，来得到可解释性。

在提出模型之前，要借用北大、哈工大和加州大学圣巴巴拉分校在 ACL 2019 的一篇论文中联合提出的一个全新的生成性解释框架和里面的两个名词细粒度（也就是我们上文提到的关键词）和解释因子（建立结果和预测解释间的联系）。我们用到一个文本分类生成性解释框架（GEF），该框架能够在进行分类预测的同时产生预测结果解释的细粒度信息，它的具体表现就是，基于留言详情推测细粒度信息并进行精简，并使用该信息辅助预测分类结果的解释，提升整体模型表现。再加上一个最小化风险训练方法，能够生成整个预测结果的合理解释。

具体来说，就是我们运用条件式变分自编码器（CVAE）+文本分类生成性解释框架的模式，CVAE 的好处是能够生成带情感的文本，来捕捉更大的多样性。在 CVAE 上应用 GEF，来生成有不同情感倾向（积极、消极和中立）的解释，再用 BLEU 分数对生成的文本解释进行评分，就能得到最终的可解释度（文章最后附上 GEF 和 GEF+ CVAE 结构图）。

六．总结

一个智慧的政务系统，就是要用最简单的操作达到最多的效果，要拒绝大量依赖人工来处理繁杂的文本信息，我们根据一个政务系统所需要的功能和内在的逻辑，将功能的实现外化为三个方面，整篇文章就是围绕着群众留言分类、热点问题挖掘和答复意见评价这三个方面来进行自然语言处理和文本数据挖掘的。另外，处理政务，还是要有比较高的准确度，能做到精准分类，对点挖掘。

在群众留言分类中，重点就是分门别类，我们主要用到了朴素贝叶斯算法来创建分类模型，使用概率统计的知识对样本数据集进行分类，这一阶段也是机械性阶段，由程序完成。然后再来检验分类的准确率，通过将交叉表矩阵对角化求和，与交叉表求和计算出分类模型的失误率，进行修改得出预测的精度为 80%。

对于热点问题挖掘来说，最重要的是要制定好对热点问题的判决标准，说简单点，就是在某一时段内同一地区的同一主题的留言次数更多的问题。

我们处理答复意见评价时，产生过最多的分歧，因为我们对于相关性、完整性和可解释性的理解都不一样，在搜索时也遇到了更多不同的理解。后来做了一个简单的解释：相关性就是判断留言详情和答复意见是否能够得到同一个主题；

完整性就是留言详情和答复意见在分别提取关键词后的相似度，即答复是否针对留言提及的所有点。在处理相关性时，我们计算出留言详情和针对留言详情的答复意见的相关度，在过程中其实更多的利用了相似度，通过 minhash 算法利用 Jaccard 计算距离的方式，计算两个文本之间的相关度。我们理解的可解释性就是我们能理解这个文本的程度，是否能轻松的获知文本信息的感情色彩和关键信息等，主要运用条件式变分自编码器+文本分类生成性解释框架的模式。

参 考 文 献

[1] 布雷特·兰茨（Brett Lantz）著,李洪成，许金炜，李舰译.《机器学习与 R 语言（原书第 2 版）》.华章出版社

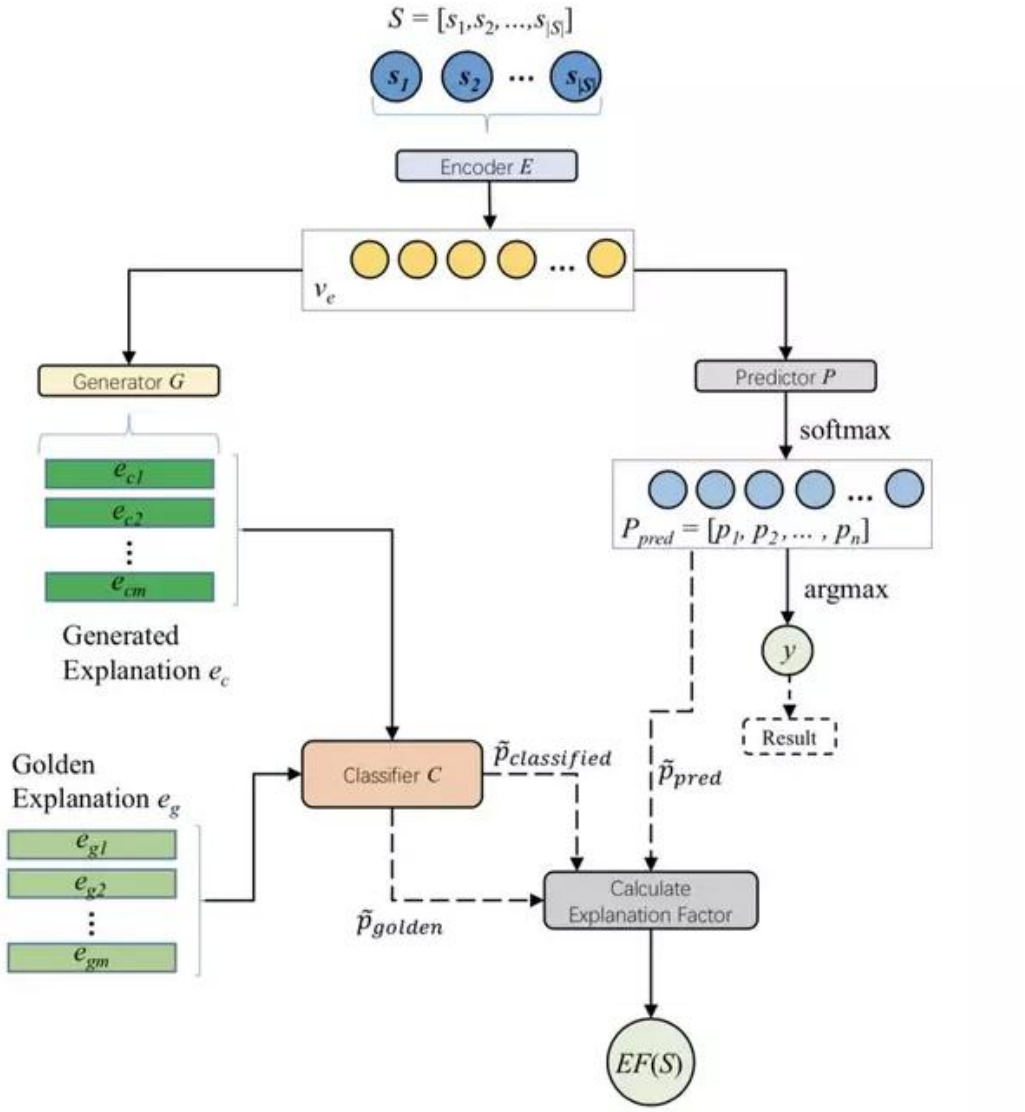
[2]《LDA-math-LDA 文本建模》

[3] Gamelife27.《基于 Python 的中英文分词基础：正则表达式和 jieba 分词器》, 2019-09-01

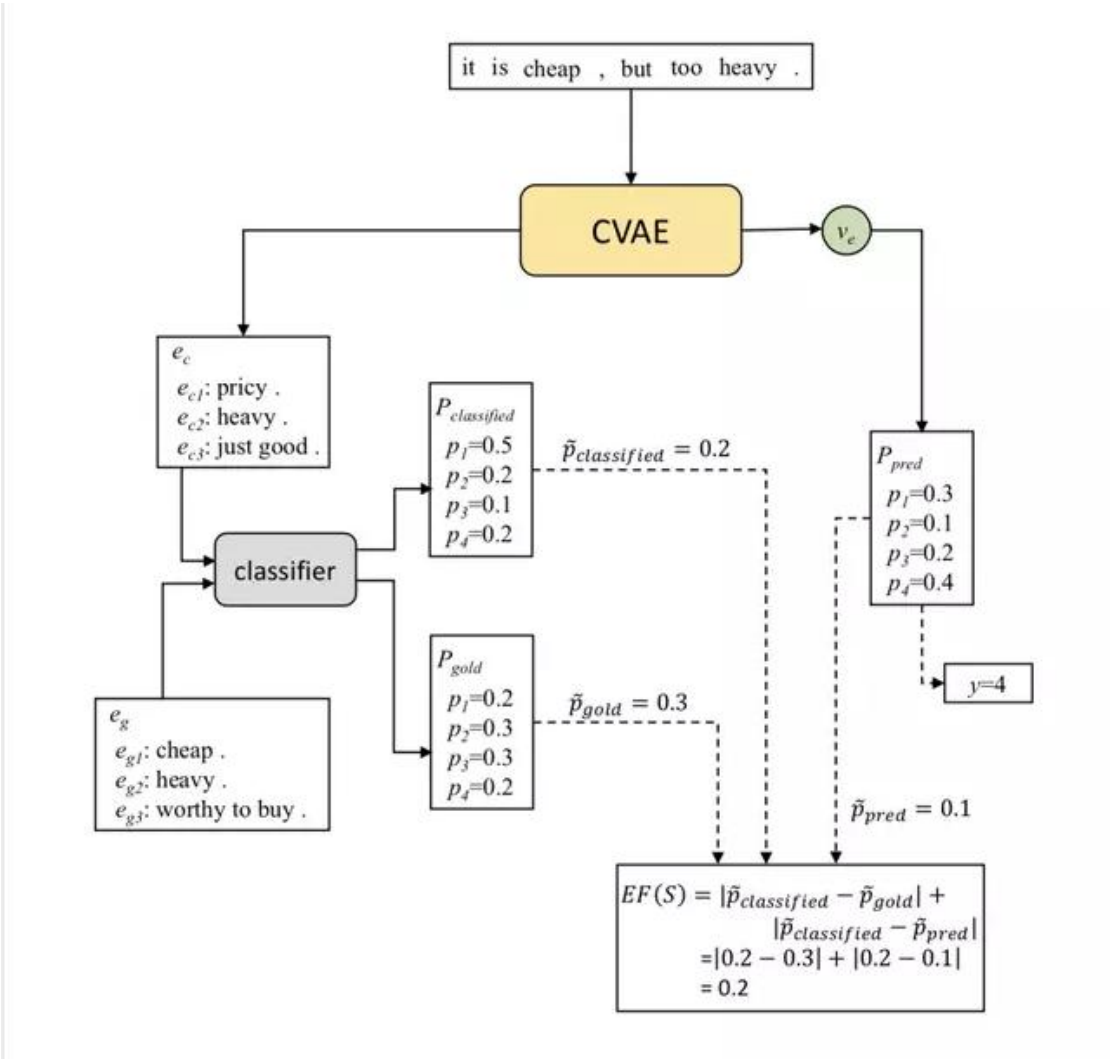
[4] 宗成庆.《统计自然语言处理》.清华大学出版社，2013-08-01

[5] shanshant.《机器学习-LDA 主题模型》，2019-04-22

附 录


$S = [s_1, s_2, \ldots, s_{|S|}]$
s_1 s_2 ... $s_{|S|}$
Encoder E
v_e
Generator G
Predictor P
softmax
e_{c1}
e_{c2}
e_{cm}
$P_{pred} = [p_1, p_2, \ldots, p_n]$
argmax
Generated Explanation e_c
y
Result
Golden Explanation e_g
Classifier C
$\tilde{p}_{classified}$
$\tilde{p}_{pred}$
e_{g1}
e_{g2}
e_{gm}
Calculate Explanation Factor
$\tilde{p}_{golden}$
$EF(S)$

GEF 结构图


it is cheap , but too heavy .
CVAE
v_c
e_c
e_{c1}: pricy .
e_{c2}: heavy .
e_{c3}: just good .
classifier
e_g
e_{g1}: cheap .
e_{g2}: heavy .
e_{g3}: worthy to buy .
$P_{classified}$
p_1=0.5
p_2=0.2
p_3=0.1
p_4=0.2
P_{gold}
p_1=0.2
p_2=0.3
p_3=0.3
p_4=0.2
P_{pred}
p_1=0.3
p_2=0.1
p_3=0.2
p_4=0.4
$\tilde{p}_{classified} = 0.2$
$\tilde{p}_{gold} = 0.3$
$\tilde{p}_{pred} = 0.1$
y=4
$EF(S) = |\tilde{p}_{classified} - \tilde{p}_{gold}| + |\tilde{p}_{classified} - \tilde{p}_{pred}|$
$= |0.2 - 0.3| + |0.2 - 0.1|$
$= 0.2$

GEF+ CVAE 结构图