C 题：“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。

关键字： 文本挖掘 聚类分析 文本分类

目录

第一章 引言

1.1 研究背景

进入 21 世纪，互联网发展可谓是日新月异。越深越多的人蓼与到互联网甲以便利的信息发布方式和渠道使得互联网中的信息数量极速增长。大量文本以及富文本信息无疑丰富了人们的生活。但是同时，大量文本信息又给用户在寻找和关注重点热点信息时带来了困扰。搜索引擎的出现解决了人们寻找某些固定意图信息的需求。另一方面，人们在获取固定信息的同时，也希望关注一些领域中的热门信息，以及这些热门信息的来龙去脉和事件未来走向等泛化的信息。为了满足人们这种需求，网络中也出现了相应的服务和应用，例如网络订阅 RSS 服务，当然这需要用户订阅一些固定信息源来获得最新信息；新闻资讯方面信息搜索也一定程度上解决了上述需求，但还是不能使用户从全局上来把握热门信息演化过程。当人们希望关注某个方面的热点信息及其在时间线上的发展趋势时，往往会显得比较困难。

另一方面，随着电子政务的进一步发展，政府部门内部及政府部门之间产生了大量政务信息。经过前两年电子政务基础资源的大规模建设，海量政务信息资源挖掘和电子政务知识管理等深层次应用正逐步进入电子政务舞台，对电子政务实施数据挖掘将成为政府信息化的一个新的研究方向。

1.2 研究意义

近年来,由于新媒体的快速发展,微博、微信、网易等已成为广为人知的社交媒体平台。“智慧政务”的文本挖掘可以将分散的信息整合汇集且提取有效信息从而协助政府办案,利于政府对一些突发事件的处理。在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系对留言进行分类，以便后续将群众留言分派至相应的职能部门 处理。目前，大部分电子政务系统还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。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，并尝试实现。及时发现热点问题有助于相关 部门进行有针对性地处理，提升服务效率。

第二章群众留言分类

2.1 留言分类主要内容

在处理网络问政平台的群众留言时，工作人员首先按照一定的划分体系对留言进行分类，以便后续将群众留言分派至相应的职能部门 处理。目前，大部分电子政务系统还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。

*在 21 世纪 ，计算机非常重要的特征是信息化、数字化和网络化 ，计算机网络经过 40 年的发展和完善 ，已经广泛应用于各个领域。群众留言已成为人们信息交流和交换的一种重要方式 ，它可以实现网站与客户之间及不同客户之间的交流与沟通。而怎么利用机器语言更好的对群众留言进行分类*问题属于文本分类中的多分类，一般用word2vec，通过Python安装Pandas、Scikit-learn、XGBoost、TextBlob、Keras 实现。

2.2 留言分类技术步骤分析

第一步是准备数据集，包括加载数据集和执行基本预处理，然后把数据集分为训练集和验证集。群众留言纷繁众多，首先加载数据集附件，创建一个dataframe，列名为 text 和 label，将数据集分为训练集和验证集，label 编码为目标变量。

```
#导入数据集预处理、特征工程和模型训练所需的库
from sklearn import model_selection, preprocessing, linear_model, naive_bayes, metrics, svm
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn import decomposition, ensemble

import pandas, xgboost, numpy, textblob, string
from keras.preprocessing import text, sequence
from keras import layers, models, optimizers
```

```
#将数据集分为训练集和验证集
train_x, valid_x, train_y, valid_y = model_selection.train_test_split(trainDF['text'], trainDF['label'])

# label编码为目标变量
encoder = preprocessing.LabelEncoder()
train_y = encoder.fit_transform(train_y)
valid_y = encoder.fit_transform(valid_y)
```

第二部分是特征工程，在这一步，原始数据将被转换为特征向量，另外也会根据现有的数据创建新的特征。为了从数据集中选出重要的特征，我们以 TF-IDF 向量作为特征，然后加载预先训练好的词嵌入模型、创建一个分词对象、将文本文档转换为分词序列并填充它们、创建分词和各自嵌入的映射

TF-IDF 是一个统计方法，用来评估某个词语对于一个文件集或文档库中的其中一份文件的重要程度。词频 TF 计算了一个单词在文档中出现的次数，它认为一个单词的重要性和它在文档中出现的次数呈正比。逆向文档频率 IDF，是指一个单词在文档中的区分度。它认为一个单词出现在的文档数越少，就越能通过这个单词把该文档和其他文档区分开。IDF 越大就代表该单词的区分度越大。所以 TF-IDF 实际上是词频 TF 和逆向文档频率 IDF 的乘积。这样我们倾向于找到 TF 和 IDF 取值都高的单词作为区分，即这个单词在一个文档中出现的次数多，同时又很少出现在其他文档中。这样的单词适合用于分类。TF-IDF 分数由两部

分组成：第一部分是计算标准的词语频率（TF），第二部分是逆文档频率（IDF）。其中计算语料库中文档总数除以含有该词语的文档数量，然后再取对数就是逆文档频率。

TF(t)=（该词语在文档出现的次数）/（文档中词语的总数）；↓
IDF(t)= log_e（文档总数/出现该词语的文档总数）；↓
TF-IDF 向量可以由不同级别的分词产生（单个词语，词性，多个词（n-grams））；↓
词语级别 TF-IDF：矩阵代表了每个词语在不同文档中的 TF-IDF 分数；↓
N-gram 级别 TF-IDF: N-grams 是多个词语在一起的组合，这个矩阵代表了 N-grams 的 TF-IDF 分数；↓
词性级别 TF-IDF：矩阵代表了语料中多个词性的 TF-IDF 分数；↵

```
#词语级tf-idf
tfidf_vect = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}', max_features=5000)
tfidf_vect.fit(trainDF['text'])
xtrain_tfidf = tfidf_vect.transform(train_x)
xvalid_tfidf = tfidf_vect.transform(valid_x)

# ngram 级tf-idf
tfidf_vect_ngram = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}', ngram_range=(2,3), max_features=5000)
tfidf_vect_ngram.fit(trainDF['text'])
xtrain_tfidf_ngram = tfidf_vect_ngram.transform(train_x)
xvalid_tfidf_ngram = tfidf_vect_ngram.transform(valid_x)

#词性级tf-idf
tfidf_vect_ngram_chars = TfidfVectorizer(analyzer='char', token_pattern=r'\w{1,}', ngram_range=(2,3), max_features=5000)
tfidf_vect_ngram_chars.fit(trainDF['text'])
xtrain_tfidf_ngram_chars = tfidf_vect_ngram_chars.transform(train_x)
xvalid_tfidf_ngram_chars = tfidf_vect_ngram_chars.transform(valid_x)
```

```
#加载预先训练好的词嵌入向量
embeddings_index = {}
for i, line in enumerate(open('data/wiki-news-300d-1M.vec')):
values = line.split()
embeddings_index[values[0]] = numpy.asarray(values[1:], dtype='float32')

#创建一个分词器
token = text.Tokenizer()
token.fit_on_texts(trainDF['text'])
word_index = token.word_index
```

```
#将文本转换为分词序列，并填充它们保证得到相同长度的向量
train_seq_x = sequence.pad_sequences(token.texts_to_sequences(train_x), maxlen=70)
valid_seq_x = sequence.pad_sequences(token.texts_to_sequences(valid_x), maxlen=70)

#创建分词嵌入映射
embedding_matrix = numpy.zeros((len(word_index) + 1, 300))
for word, i in word_index.items():
embedding_vector = embeddings_index.get(word)
if embedding_vector is not None:
embedding_matrix[i] = embedding_vector
```

虽然上述框架可以应用于多个文本分类问题，但是为了达到更高的准确率，让网页上的不同留言更加精确的分类，可以在总体框架中进行一些改进。例如下面是一些改进文本分类模型和该框架性能的技巧：

1.清洗留言文本：文本清洗有助于减少文本数据中出现的噪声，包括停用词、标点符号、后缀变化等。

2.组合文本特征向量的文本/NLP 特征：特征工程阶段，我们把生成的文本特征向量组合在一起可能会提高文本分类器的准确率。

3 接着创建基于文本的特征比如词性标注的频率分布、名词数量、动词数量、形容词数量、副词数量、代词数量来解决文本语义带来的文本交叉。

```
trainDF['word_density'] = trainDF['char_count'] / (trainDF['word_count']+1)
trainDF['punctuation_count'] = trainDF['text'].apply(lambda x: len("".join(_ for _ in x if _ in
string.punctuation)))
trainDF['title_word_count'] = trainDF['text'].apply(lambda x: len([wrd for wrd in x.split() if wrd.istitle()]))
trainDF['upper_case_word_count'] = trainDF['text'].apply(lambda x: len([wrd for wrd in x.split() if
wrd.isupper()]))
pos_family = {
'noun' : ['NN','NNS','NNP','NNPS'],
'pron' : ['PRP','PRP$','WP','WP$'],
'verb' : ['VB','VBD','VBG','VBN','VBP','VBZ'],
'adj' : ['JJ','JJR','JJS'],
'adv' : ['RB','RBR','RBS','WRB']
```

第三部分是利用之前创建的特征训练一个分类器。朴素贝叶斯分类最适合的场景就是文本分类、情感分析和垃圾邮件识别。其中情感分析和垃圾邮件识别都是通过文本来进行判断。所以朴素贝叶斯也常用于自然语言处理 NLP 的工具。多项式朴素贝叶斯：特征变量是离散变量，符合多项分布，在文档分类中特征变量体现在一个单词出现的次数，或者是单词的 TF-IDF 值等。

朴素贝叶斯群众留言分类流程和步骤朴素贝叶斯分类算法利用贝叶斯定理的优势,在群众留言分类中有广泛应用,是文本分类最为精确的技术。在智能文本分类技术中 ,通过贝叶斯分类器的“自我学习”智能技术 ,能有效保护信息的正常通信,过滤垃圾信息的骚扰。 朴素贝叶斯分类分为以下 3 个阶段。

第 1 阶段:准备工作阶段。收集大量正常留言和垃圾留言作为样本 ,确定特征属性 ,并对每个特征属 性进行适当划分 ,然后提取留言样本中主题和信体中的字符串 ,建立对应的数据库分类 ,输出特征属性和训练样本。

第 2 阶段:分类器训练阶段。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,创建贝叶斯概率库统计出每个字串在网络留言中出现的概率以及在正常留言中出现的概率,然后根据公式计算出留言中含某字串则为垃圾留言的概率。

第 3 阶段:应用阶段。使用训练好的 Bayes 分类器对分类项进行分类 ,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。通过对留言样本的训练,Bayes 分类器可以自动获取垃圾留言的特征,并根据特征的变化,对网络留言进行有效分类。

2.3 评价方法

通常使用 F-Score 对分类方法进行评价：

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i},$$

其中 P_i 为第 i 类的查准率， R_i 为第 i 类的查全率。

一般来说准确率和召回率呈负相关，一个高，一个就低，如果两个都低，一定是有问题的。 一般来说，精确度和召回率之间是矛盾的，这里引入 F1-Score 作为综合指标，就是为了平衡准确率和召回率的影响，较为全面地评价一个分类器。F1 是精确率和召回率的调和平均。

2.4 本章小结

理论上，朴素贝叶斯模型与其他分类方法相比具有最小的误差率。但是实际上并非总是如此，这是因为朴素贝叶斯模型假设属性之间相互独立，这个假设在实际应用中往往是不成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好。不过朴素朴素贝叶斯分类是贝叶斯分类中最简单，也是常见的一种分类方法。而我们所想要实现的留言过滤分类其实是一种分类行为，是通过对于概率的判断，来对样本进行一个归类的过程。所以用朴素贝叶斯分类还是可以实现在处理网络问政平台的群众留言时按照一定的划分体系对留言进行分类的来提高工作人员的工作效率的。

第二章 热点问题挖掘

3.1 热点问题挖掘的主要内容

某一时段内群众集中反映的某一问题可称为热点问题，如“XXX 小区多位业主多次反映 入夏以来小区楼下烧烤店深夜经营导致噪音和油烟扰民”。及时发现热点问题，有助于相关 部门进行有针对性地处理，提升服务效率。热点问题挖掘和分析主要包括：首先，需要我们从互联网中获取信息。在获取了原始网页信息之后,我们需要进行网页分析,提取出对热点信息分析有帮助的关键信息。如反映某一问题的标题，日期，正文，发布人等。而在对原始的网页信息进行提取之后，我们需要对这些文本的信息进行进一步的处理。这些主要包括：

留言投稿内容的分类。出于留言内容涉及到方方面面，如不同领域，政治，经济，教育，娱乐等，或不同话题。内容分类不仅能够帮助用户更好地浏览和查

找需要的内容，还有如下优点：能够将海量的用户留言数据有效地组织起来，从而发现一些问题。面对互联网上不断涌现的信息，它们之间的内容实际上存在一定程度上的重叠。比如一段时间集中爆发的问题，多人反映同一问题。如果能够对留言信息按主题分类，给内容相似的留言赋予相同的类别标签，人们就可以按照主题来阅读，从而在主题上把握热点问题网络中一些问题之间存在明显的不同，这些信息用分类的方法可以很容易识别出来，如果不加区别的直接将它们放在一起聚类的话，可能会形成噪音数据而影响聚类的结果，而且将这些“距离很远”的数据考虑进来聚类，还会影响聚类的效率。

对同一个类别的信息进行聚类。所谓聚类就是指将该类别下相同或者相似的留言内容聚集在一起，将众多的留言中识别相似的留言，这样更加方便浏览信息，掌握问题的热度等。最后是对热点信息的可视化展现，在进行留言信息的分析之后，我们会展现信息的大类别以及每个类别下的主题，以及这些主题在时间轴上留言条数的变化，从而展示出问题热度的变化。帮助我们掌握热点问题，及时发现热点问题，有助于相关部门进行有针对性地处理，提升服务效率。

3.2 热点问题分析相关技术

3.2.1 文本处理技术

对留言问题来说，另外一个比较重要的部分就是网页分词。网页分词技术是进行网页信息的语义分析的关键所在。中文分词技术目前是一门发展的相对成熟的技术，按照方法的不同，可以大致分为三类：

基于字符串匹配的分词方法，基于理解的分词方法和基于统计的分词方法。所谓基于字符串匹配的方法，它是按照一定的策略将待分析的汉字串与一个“充分大的"机器词典中的词条进行配，若在词典中找到某个字符串，则匹配成功（识别出一个词)。而基于理解的方法是指通过让计算机模拟人对句子的理解，达到识别词的效果。其基本思想就是在分词的同时进行句法、语义分析，利用句法信息和语义信息来处理歧义现象。而基于统计的方法则是通过字的协同出现作为分词的依据。因为词是稳定的字的组合，因此在上下文中，相邻的字同时出现的次数越多，就越有可能构成一个词。因此字与字相邻共现的频率或概率能够较好的反映成词的可信度。可以对语料中相邻共现的各个字的组合的频度进行统计，计算它们的互现信息。定义两个字的互现信息，计算两个汉字 X、Y 的相邻共现概率。便可以认为此字组可能构成了一个词。

3.2.2 文本分类

数据分类技术是根据观测到的数据的一些基本特征，经过一定的学习方法，建立起对目标值的预测函数，从而对新的未知的数据实例进行归类。简单的来说，数据分类是一个两步的过程。第一步，建立一个模型，该模型用来描述预定的数据类集。它是通过预定的数据类集的特征来进行构建的。这个预定的数据类集也叫训练集。第二步，是使用第一步得到的模型进行分类，将得到的模型分类函数作用在新数据实例的特征上，就能够将新的数据划分到相应的预定义的数据类别中。

3.2.3 文本聚类

聚类（Clustering）分析的目标和分类比较类似，即将一个数据集合按照样本之间的相互关系划分程不同的类别。而它们之间最大的不同则在于聚类的类别不是预先知道的。我们可能既不知道需要划分为多少类，也不知道聚集出来的类别的具体含义。聚类分析要求在将数据对象划分成的类中，在同一类中的数据对象具有较高的相似度，而不同类之间的差别则尽可能大。所以在聚类分析中，需要面临的几个主要问题是：数据划分的方式，数据个体之间距离的定义方式以及数据类别标签如何产生。

数据个体之间的距离的定义也是一个十分重要的问题，它是用来衡量在聚类过程中个体之间的远近关系，从而用来判断两个个体是否应该在同一个类别中。个体之间的距离表示也有很多方法，主要使用的有欧氏距离等。最后，在聚类的过程完成之后，对于有些聚类的应用，例如文本本聚类和聚类之后的检索。则需要用一个合适的短语或者标签来描述聚出来的类别的基本信息，也就是聚类标签的生成。一般来说，聚类标签的生成利用自然语言处理的技术，从类别的个体中通过句法和语法的分析得到最能描述类别信息的短语合成列别标签。目前来说，并没有一个十分成熟和通用的方法来完成这项任务。很多时候，需要根据具体的情况，选择合适的标签产生的方法。

3.2.4 数据可视化

数据可视化主要旨在借助于图形化手段，清晰有效地传达与沟通信息。但这并不意味着，数据可视化就一定要因为实现其功能用途而令人感到枯燥乏味，或

者是为了看上去绚丽多彩而显得极端复杂。为了有效地传达思想概念，美学形式与功能需要齐头并进，通过直观地传达关键的方面与特征，从而实现相当稀疏而又复杂的数据集的深入洞察。然而，设计人员往往并不能很好地把握设计与功能之间的平衡，从而创造出华而不实的数据可视化形式，无法达到其主要目的，也就是传达与沟通信息。数据可视化与信息图形、信息可视化、科学可视化以及统计图形密切相关。当前，在研究、教学和开发领域，数据可视化乃是一个极为活跃而又关键的方面。“数据可视化”这条术语实现了成熟的科学可视化领域与较年轻的信息可视化领域的统一。

3.3 对数据进行分析

运用 Rstudio 软件进行数据处理

```
#读取数据
liuyan<-read.csv("C:\\Users\\asus\\Desktop\\12.csv")
d<-dist(liuyan,method = "euclidean",diag=F);d
```

```
library(NLP)
library(tm)
#读取分词后的文件，注意用的是 readLines,因为是把每一条评论当成一个文档

mydoc<-readLines(file.choose())

edit(mydoc) #查看文件，看看是否有乱码的现象

#生成语料库

mydoc.vec<-VectorSource(mydoc)

mydoc.corpus<-Corpus(mydoc.vec)
```

```
inspect(mydoc.corpus) #查看语料库中的文档
#导出语料库，方法：writeCorpus(x, path = ".", filenames = NULL)

writeCorpus(mydoc.corpus, path = "D:/text_fenci/",filenames = paste(seq_along(mydoc.corpus), ".txt", sep = ""))

#引入停用词表

#读取时出现了以下错误：EOF within quoted string  所以加上了 quote=""

data_stw<-read.table(file=file.choose(),colClasses="character",quote = "")

cnword<-c(NULL)

for(i in 1:dim(data_stw)[1]){
    cnword<-c(cnword,data_stw[i,1])
}
#出现以下错误：In gsub(sprintf("(*UCP)\\b(%s)\\b", paste(sort(words, decreasing = TRUE),可能是停用此表中出现了非 UTF-8 的元素，所以先去掉未知的编码字符
stopwords<-cnword[Encoding(cnword)!="unknown"]
mydoc.corpus<-tm_map(mydoc.corpus,removeWords,stopwords)#去掉停用词
mydoc.corpus<-tm_map(mydoc.corpus,removeNumbers)  #去掉数字
mydoc.corpus<-tm_map(mydoc.corpus,stripWhitespace) #删除空白
#建立 TDM 矩阵
    #removePunctuation 表示去除标点
    #minDocFreq=5 表示只有在文档中至少出现 5 次的词才会出现在 TDM 的行中
    #wordLengths = c(1, Inf)表示字的长度至少从 1 开始。
    #默认的加权方式是 TF，即词频，这里采用 Tf-Idf，该方法用于评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度：
control<-list(removePunctuation=T,minDocFreq=5,wordLengths = c(1,
```

```
Inf),weighting = weightTfIdf)
#建立 TermDocumentMatrix 矩阵
mydoc.tdm<-TermDocumentMatrix(mydoc.corpus,control)
length(mydoc.tdm$dimnames$Terms) #查看原先有多少个词

#词太多了需要降维
tdm_removed<-removeSparseTerms(mydoc.tdm, 0.95) #去掉低于 95%的的稀疏条数，可以根据自己的文本，自行调整

length(tdm_removed$dimnames$Terms) #再次查看还有多少个词
findFreqTerms(mydoc.tdm,3) #查看高频词

findAssocs(mydoc.tdm,"态度",0.5) #查找与“态度”相关系数大于 0.5 的词

#将数据转换成数据框结构
mydata <- as.data.frame(inspect(tdm_removed))
#开始聚类分析-层次聚类
mydata.scale<-scale(mydata) #数据标准化和中心和变换

d<-dist(mydata.scale,method="euclidean") #计算矩阵距离来的函数 tm 包

fit <- hclust(d, method="ward.D")   #层次聚类算法

plot(fit)

#kmeans 聚类

k<-3

kmeansRes <- kmeans(mydata,k) #k 是聚类数
```

mode(kmeansRes) #kmeansRes 的内容

names(kmeansRes)

kmeansRes$cluster #聚类结果

热度排名	问题ID	时间范围	地点/人群	问题描述	热度指数
1	1	2019/08/18 至 2019/09/04	A 市 A5 区魅力之城小区	小区临街餐饮店油烟噪音扰民	1
2	2	2017/06/08 至 2019/11/22	A 市经济学院学生	学校强制学生去定点企业实习	2
.....	...	...	...	..	...

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
1	360104	A012417	A 市魅力之城 商铺无排烟管道，小区内到处油烟味	2019/08/18 14:44:00	A 市魅力之城小区自打交房入住后， 底层商铺无排烟管道，经营餐馆导致 大量油烟排入小区内，每天到凌晨还 在营业……	0	0
1	360105	A120356	A5 区魅力之城小区一楼被搞成商业门面，噪音扰	2019/08/26 08:33:03	我们是魅力之城小区居民，小区朝北 大门两侧的楼栋下面一楼，本来应是	1	0

			民严重		架空层，现搞成商业门面，嗓声严重 扰民，有很大的油烟味往楼上窜，没 办法居住……		
1	360106	A235367	A 市魅力之城 小区底层商铺营业到凌晨，各种噪音好痛苦	2019/08/26 01:50:38	2019 年 5 月起，小区楼下商铺越发嚣 张，不仅营业到凌晨不休息，各种烧 烤、喝酒的噪音严重影响了小区居民 休息……	0	0
…	…	…	…	…	…	…	…
1	360109	A0080252	魅力之城小区底层门店深夜经营，各种噪音扰民	2019/09/04 21:00:18	您好：我是魅力之城小区的业主，小 区临街的一楼是商铺，尤其是餐馆夜 宵摊等，每到凌晨都还在营业，每到 晚上睡觉耳边都充斥着吆喝……	0	0
2	360110	A110021	A 市经济学院 寒假过年期间组织学生去工厂工作	2019/11/22 14:42:14	西地省 A 市经济学院寒假过年期间组 织学生去工厂工作，过年本该是家人 团聚的时光，很多家长一年回来一次， 也就过年和自己孩子见一次面，可是 这样搞……	0	0
2	360111	A1204455	A 市经济学院 组织学生外出打工合理吗？	2019/11/5 10:31:38	学校组织我们学生在外边打工,在东 莞做流水线工作，还要倒白夜班。本 来都在学校好好上课，十月底突然说 组织到外省打工……	1	0
…	…	…	…	…	…	…	…
2	36011	A0182	A 市经济学	2017/06/	系里要求我们在实习前分别去	9	0

	4	491	院 变相强制实习	08 17:31:20	指定的 不同公司实训，我这的公司的工作内 容和老师之前介绍以及我们专业几乎 不对口，不做满 6 个月不给实训分， 不能毕业……		

3.4 本章小结

通过上述的聚类方法实现，我们对最后的聚类结果进行了大致的分析。其中，对于留言问题的集合，聚类算法能够很好地对这些留言进行区分。但是对于没有明显聚类特征的留言集合，聚类算法也是无能为力。聚类算法研究的数据对象是有明显聚类的，并且存在少量噪音的数据集合。从留言分布可以看出，把特定地点或人群的数据归并，即把相似的留言归为同一问题，因此在聚类的时候，就能够不受到干扰。从最后的结果来看，在先进行网页分类之后，聚类的准确度得到了很大的提高和改善。通过聚类分析以及数据文本挖掘，我们及时发现热点问题，有助于相关 部门进行有针对性地处理，提升服务效率。

第四章　答复意见的评价

4.1 相关性

从答复的相关性来说，评价答复意见质量。说到相关性，那么很大程度上会联想到 R 语言的相关性分析。但是很明显，这道题如果能做相关性分析和相关性检验，应该提供的是数值型数据。于是我们就需要换一个思路。这里我们应该要考虑是否答非所问。即答复意见与被答复问题是否对应。

所以我们可以用 R 语言提取一些关键词出来。已知：关于提取关键词的方法，除了 TF-IDF 算法，比较有名的还有 TextRank 算法。它是基于 PageRank 衍生出来的自然语言处理算法，是一种基于图论的排序算法，以文本的相似度作为边的权重，迭代计算每个文本的 TextRank 值，最后把排名高的文本抽取出来，作为这段文本的关键词或者文本摘要。之所以提到关键词和文本摘要，两者其实宗旨是一样的，就是自动化提取文本的重要表征文字。

如果分词是以词组作为切分，那么得到的是关键词。以词作为切分的时候，

构成词与词之间是否连接的，是词之间是否相邻。相邻关系可以分为 n 元，不过在中文中，认为 2 元关系已经非常足够了（比如一句话是：“我/有/一只/小/毛驴/我/从来/也/不/骑”，那么设置二元会让“一只”和“毛驴”发生关联，这就足够了）。下面，将用 R 语言的 textrank 包来实现关键词的提取和文本摘要。（以下方法仅提取一列，题目应当提取两列来做比较。另一列提取方法类似。）

先用 R 语言安装必备的包

```
install.packages("pacman")
library(pacman)
p_load(tidyverse,tidytext,textrank,rio,jiebaR)
```

然后导入数据

```
import("./文件名称") ->  列名称
列名称     #输出列
```

因为要做关键词和关键句的提取，因此我们要进行分词和分句。分词还是利用 jiebaR。不过这次，我们希望能够在得到词频的同时，得到每个词的词性，然后只把名词提取出来。

```
列名称%>%
mutate(id = 1:n()) ->  列名称  #给文档编号
worker(type = "tag") -> wk #构造一个分词器，需要得到词性
列名称  %>%
mutate(words = map(列名称,tagging,jieba = wk)) %>% #给文档进行逐个分词
mutate(word_tag = map(words,enframe,name = "tag",value = "word")) %>%
select(id,word_tag) -> hire_words
```

然后，我们分组进行关键词提取

...挑选规则是：词频必须大于 1，在此基础上，n 元越高越好。

等到我们把问题和意见的关键字提取出来后，我们再进行对比。此时就可以看出大致的相似性。

还可以想到的角度：度量数据的相似性。

- **-相似性Similarity**
  - 数值测量两个数据对象类似程度
  - 目标越相似时值越大
  - 通常介于 [0,1]
- 2个或多个状态, e.g., red, yellow, blue, green (二元属性的推广)
- Method 1: 简单匹配
  - *m*: p个变量中匹配的个数, *p*: 全部变量的个数

$$d(i,j)=\frac{p-m}{p}$$

- Method 2:使用一系列的二进制属性
  - 为M个名义状态的每一个产生一个新的二进制/二元属性

思考能否对关键词编码进行相关性分析。

4.2 完整性

从答复的完整性来说，评价答复意见质量。完整性应该是说明是否满足某种标准，是否有一套完整的体系，是否有标准的开头和结尾。

首先应该可以看看是否有缺失值。有无异常数据或重复数据。

4.3 可解释性

从答复的可解释性来说，评价答复意见质量。可解释性是指答复意见内容中的相关解释，有无理论支撑。解释中有没有引经据典。如果不能解决问题，有没有相应的理论、法律法规、现实情况来证明真的不能解决。以下也是要用 R 语言找出一些名人名言，法律法规等。

这里我们用 R 语言来找关键句。如果要得到文本的关键句子，还是要对每句话进行分词，得到每句话的基本词要素。根据句子之间是否包含相同的词语，我们可以得到句子的相似度矩阵，然后再根据相似度矩阵来得到最关键的句子（也就是与其他句子关联性最强的那个句子）。当句子比较多的时候，这个计算量是非常大的。文本摘要就是从文档中提出我们认为最关键的句子。我们会用

textrank 包的 textrank_sentences 函数，这要求我们有一个分句的数据框，还有一个分词的数据框（不过这次需要去重复，也就是说分词表中每个文档不能有重复的词）。非常重要的一点是，这次分词必须以句子为单位进行划分。

实践证明，TextRank 算法是一个比较耗时的算法，因为它依赖于图计算，需要构成相似度矩阵。当数据量变大的时候，运行时间会呈“几何级”增长。但是对于中小型的文本来说，这个方法还是非常不错的。但是中小型的文本，还需要摘要么？尽管如此，这还是一个非常直观的算法，如果 TF-IDF 在一些时候不好用的话，这是一个非常好的候补选项。

4.4 本章小结

从相关性、完整性、可解释性出发，可能会用不同的理念去评价答复意见的质量。我们要有发散思维。答复意见的质量还可以看是否有一些多余词句，有无直奔主题，有无存在废话。

第五章 总结

C 题要完成 3 个大问题：群众留言分类、热点问题挖掘、答复意见的评价。

群众留言分类一般用 word2vec，通过 Python 安装 Pandas、Scikit-learn、XGBoost、TextBlob、Keras 实现。第一步是准备数据集，第二部分是特征工程，第三部分是利用之前创建的特征训练一个分类器。依据题目所给的 F-Score 分类法对所得模型进行评价。过程中运用 R 语言编写代码，以朴素贝叶斯模型进行支撑。我们需要了解公式的用法。朴素贝叶斯的主要优点有：（1）朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率。（2）对小规模的数据表现很好，能个处理多分类任务，适合增量式训练，尤其是数据量超出内存时，我们可以一批批的去增量训练。（3）对缺失数据不太敏感，算法也比较简单，常用于文本分类。缺陷已在 2.4 总结内叙述。

热点问题挖掘对集中的留言进行归类。要求定义合理的指标，还要给出两张完美的 Excel 表格。主要运用文本分析、文本分类、文本聚类、数据可视化。这里有一个重要的方法——分词。牵涉到数据清洗、文本特征选择、文本相似度、模型验证，调优。也是运用 R 语言来写代码。

答复意见的评价主要角度有三——相关性、完整性、可解释性。代码同样用R语言编写。相关性使用了TextRank算法，也有联想到度量数据的相似性，并给出计算公式。完整性指标体系个人各不相同。可解释性用了textrank包的textrank_sentences函数。其核心想法是提取关键句。与相关性的提取关键词有异曲同工之妙。不过要考虑数据量的大小对计算时间的影响。算法选择的不同可能工作量不同，有可能加大工作量。

如今是大数据时代，如何让可用数据透明化是我们的需求。这里的透明化是指数据可以高效率被利用查看。就像这个“智慧政务”一样，我们可以很快地筛选整理出自己最需要得到的信息。目的一定是简化我们的工作量，让数据更优秀的为我们服务。

参考文献

【1】CSDN -- https://blog.csdn.net/weixin_41931965/article/details/83831921

【2】王伟、许鑫，基于聚类的网络舆情热点发现及分析

【3】许鑫、章成志，互联网舆情分析及应用研究 情报科学

【4】刘林涛、陈志刚、赵明，网络热点新闻事件挖掘和跟踪

【5】博客——https://www.sohu.com/a/302022548_466874

【6】中国大学MOOC——数据挖掘课程
https://www.icourse163.org/learn/CUFE-1207262801?tid=1207607201#/learn/content?type=detail&id=1215382718&cid=1246251936

【7】马小龙，网络留言分类中贝叶斯复合算法的应用研究 佛山科学技术学院学报 (自然科学版)

【8】
https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/