第八届“泰迪杯”

全 国 数 据 挖 掘 挑 战 赛

作品名称：“智慧政务”中的文本挖掘应用

作品单位：

作品成员：

作品老师：

“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。

针对问题 1，首先使用 re 正则表达式去除数据中原有的无用文本，然后使用 jieba 库进行中文的分词操作，分词后有一些出现频率高但是无用的文本，这些使用停用词表的导入然后进行去除停用词，这样得到的文本相对干净，之后来绘制词云图，更加直观的显示出数据出现频率高的词，最后进行分类模型的构建，通过几种模型的对比，最终选择了线性支持向量机来构建分类模型，在用 F1 分数指标评价模型。

针对问题 2，首先对原始数据的预处理，同样的使用到了 re 正则表达式、jieba 库中文分词和去除停用词，这一问是无监督学习，要采用聚类算法，所以之后我们采用了 K-means 聚类算法，最后针对聚类后的标签进行了分组聚合，最终构建了两张热点问题表格。

针对问题 3，这里我们给出了解决问题的思路，首先是数据的预处理，同样的步骤这里就不在赘述，之后使用 word2vec 进行文本的相似度处理，比较答复意见与用户留言之间的相似度，显示出两者的相关性，同时也反映出两者的可解释性，并且，进一步进行命名实体识别，比较两者是否一致，这样就可以反应出答复意见的完整性。

另外，通过留言问题的分类和第二个的聚类问题，可以使政府人员节省很多时间，并且节省了不必要的重复性操作，提高了政府办事效率。

关键词：留言划分；热点整理；自然语言处理；文本挖掘

ABSTRACT

In recent years, with WeChat, such as weibo, mayor mailbox, sun hotline network asked ZhengPing stage gradually become the government understand the importance of public opinion, gathering intelligence, condensed bull channels, all kinds of public opinion related text data quantity rising, leave a message to past mainly rely on artificial to divide and hot spots of relevant departments work has brought great challenge.At the same time, with the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of intelligent government affairs system based on natural language processing technology has become a new trend of social governance innovation and development, which plays a great role in promoting the management level and efficiency of the government.

For question 1, the first to use re regular expressions to remove some useless text data central plains, and then use the jieba libraries for Chinese word segmentation operation, participle after high frequency but useless text, use the stop list of import and then remove the stop, the resulting text relatively clean, after to draw word cloud, more intuitive to show the data of high frequency words, finally to construct a classification model, through the comparison of several kinds of model, finally chose the linear support vector machine (SVM) to build the classification model, with F1 score index evaluation model.

For question 2, the first for the pretreatment of the raw data, the use of the same to the re regular expressions, jieba library in Chinese word segmentation and remove the stop, this question is unsupervised learning, to adopt the clustering algorithm, so after we adopted K means clustering algorithm, finally based on clustering the label after the group aggregation, finally build the two hot issues form.

For question 3, here we are given to solve the problem, the first is the pretreatment of data, the same steps would not be in here, after processing, using word2vec text similarity comparison reply opinion and similarity between user a message, shows the relevance of the two, but also reflects the interpretability of the two, and further to named entity recognition, compare the two are consistent, so that it can reflect the opinions of reply integrity.

In addition, through the classification of message questions and the second clustering problem, the government personnel can save a lot of time, and save unnecessary repetitive operations, and improve the efficiency of the government.

Key words： message division; hot spot arrangement; natural language processing; text mining

目 录

1 文本挖掘目标

1.1 问题背景

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，对于提升政府的管理水平和施政效率具有极大的推动作用。

附件给出了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。请利用自然语言处理和文本挖掘的方法解决下面的问题。

1.2 挖掘目标

本次文本挖掘目标是来自“智慧政务”中群众问政留言记录，及相关部门对部分群众留言的答复意见的信息数据，利用 jieba 中文分词，正则表达式去除干扰字符，去除停用词等进行数据预处理，之后利用词云可视化，词频向量化，TF-IDF 算法，朴素贝叶斯模型，F1-Score 评估模型以及 K-means 聚类的方法，达到所需解决问题的目标。文本挖据的目标如下：

(1) 群众留言分类。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（参考附件 1 提供的内容分类三级标签体系）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。请根据附件 2 给出的数据，建立关于留言内容的一级标签分类模型。

(2) 热点问题挖掘。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。请根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，定义合理的热度评价指标，并给出评价结果，按表 1 的格式给出排名前 5 的热点问题，并保存为文件“热点问题表.xls”。按 表 2 的格式给出相应热点问题对应的留言信息，并保存为“热点问题留言明细表.xls”。

(3) 答复意见的评价。根据答复的相关性、完整性、可解释性等方面对相关部门留言的答复意见的质量进行评价，并尝试实现。

2 群众留言分类

2.1 数据预处理

对数据进行以下预处理，将数据预处理代码封装成一个函数，供后面的操作进行调用。

2.1.1 分类标签

首先导入原始数据，然后把数据的标签提取出来。

```
    # 导入数据
    data = pd.read_excel(dataPath)
    # 分类标签，并放在列表labels中
    labels = data['一级标签'].drop_duplicates().tolist()
```

2.1.2 去除干扰字符、分词

将数据中的脏数据，对此次数据分析没有意义，没有影响或者有负面影响的数据进行去除操作，此项目需要去除一些干扰字符，包括空格，数字，大小写字符。“结巴”中文分词：做最好的 Python 中文分词组件。过程代码以及处理结果如下图所示：

```
# 使用正则表达式去除干扰字符
data_re = data_after['message'].apply(lambda x : re.sub('\s|[0-9a-zA-Z]', '', x))

# 使用jieba进行分词
data_cut = data_re.apply(lambda x : jieba.lcut(x))

```

分词是文本信息处理的基础环节，是将一个单词序列切分成一个一个单词的过程。准确的分词可以极大的提高计算机对文本信息的是被和理解能力。相反，不准确的分词将会产生大量的噪声，严重干扰计算机的识别理解能力，并对这些信息的后续处理工作产生较大的影响。

汉语的基本单位是字，由字可以组成词，由词可以组成句子，进而由一些句子组成段、节、章、篇。可见，如果需要处理一篇中文语料，从中正确的识别出词是一件非常基础且重要的工作。中文分词是以词作为基本单元，使用计算机自动对中文文本进行词语的切分，即使词之间有空格，这样方便计算机识别出各语句的重点内容。此次分词使用 python 的 jieba 库进行。

jieba 库支持三种分词模式：精确模式，试图将句子最精确地切开，适合文本分析；全模式，把句子中所有的可以成词的词语都扫描出来，速度非常快，但是不能解决歧义。；搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

2.1.3 去除停用词

停用词（Stop Words），词典译为“电脑检索中的虚字、非检索用字”。在 SEO 搜索引擎中，为节省存储空间和提高搜索效率，搜索引擎在索引页面或处理搜索请求时会自动忽略某些字或词，这些字或词即被称为停用词。

停用词一定程度上相当于过滤词（Filter Words），区别是过滤词的范围更大一些，包含情色、政治等敏感信息的关键词都会被视做过滤词加以处理，停用词本身则没有这个限制。通常意义上，停用词大致可分为如下两类。

一类是使用十分广泛，甚至是过于频繁的一些单词。比如英文的“i”、“is”、“what”，中文的“我”、“就”等， 另一类是文本中出现频率很高，但实际意义又不大的词。这一类词主要包括了语气助词、副词、介词、连词等，通常自身并无明确意义，只有将其放入一个完整的句子中才有一定作用的词语。常见的有“的”、“在”、“和”、“接着”等。

创建停用词文件，并将停用词导入，使用 read_csv 函数对文件进行读取，使用 lambda 表达式，去除停用词。

```
# 将停用词导入
stopWords = pd.read_csv(stopPath, header=None, sep='hhhhhhhhhhhh')
stopWords = list(stopWords.iloc[:, 0])
# 去除停用词
data_stop_words = data_cut.apply(lambda x : [i for i in x if i not in stopWords])
labels = data_after.loc[data_stop_words.index, '一级标签']
data_new = data_stop_words.apply(lambda x : ' '.join(x))
```

2.2 词云可视化

进行词云绘制可以分为以下几个部分。

（1） 调用前面封装好的函数，对数据进行数据预处理。

（2） 定义统计词频的函数。

（3） 绘制词云。

词云轮廓图

词云中文字的字体设置

调用前面封装好的数据预处理函数和词频统计函数，对数据进行数据预处理和构建词频字典，最后用 WordCloud 库中的 wordcloud 函数和 matplotlib 库结合进行词云的绘制，其中词云的形状和字体使用的是以上的文件，为了便于绘制 7 张，所以把绘图放在了整体的 for 循环中，所使用的代码如下所示：

```
# 词云可视化
for k in range(len(labels)):
    word_fre = {}    # 词频字典
    for i in data[data['一级标签'] == labels[k]]['message'].tolist():
        for j in i:
            if j not in word_fre.keys():
                word_fre[j] = 1
            else:
                word_fre[j] += 1

    plt.rcParams['font.sans-serif'] = 'SimHei'
    pic = plt.imread('./Img/ciyun.jpg')  # 词云图片
    wc = WordCloud(background_color='white', max_words=300, font_path='./Font/simkai.ttf', mask=pic)    # 绘制词云
    wc2 = wc.fit_words(word_fre)
    plt.figure(figsize=(16, 9))
    plt.imshow(wc2)
    plt.axis('off')
    plt.title(labels[k] + '词云', size=20)    # 标题
    plt.savefig('./Img/' + labels[k] + '词云图.png')    # 图片保存
    plt.show()    # 图片显示

```

词云最终绘制效果显示如下（这里只示例一张图片，全部图片请看作品附件中 Img 文件夹）：

2.3 分类模型构建并评估

2.3.1 数据划分和词频向量化

使用之前封装好的函数进行数据的加载，然后使用 scikit-learn 库 model_selection 模块下的 train_test_split 函数划分训练集和测试集，并指定测试集规模占总数据的 20%。

之后进行词频向量化处理，使用的是 sklearn.feature_extraction.text 下的 CountVectorizer 类。

```
# 加载数据
data_new, labels = data_process_one('./Data/附件2.xlsx', './Data/stopword.txt')
# 划分训练集和测试集
data_tr, data_te, labels_tr, labels_te = train_test_split(data_new, labels, test_size=0.2, random_state=1)
# 词频向量化
countVectorizer = CountVectorizer()
data_tr = countVectorizer.fit_transform(data_tr)
data_te = CountVectorizer(vocabulary=countVectorizer.vocabulary_).fit_transform(data_te)

```

2.3.2 TF-IDF 算法

TF-IDF（term frequency－inverse document frequency）是一种用于信息检索与数据挖掘的常用加权技术。TF 是词频(Term Frequency)，IDF 是逆文本频率指数(Inverse Document Frequency)。

TFIDF 的主要思想是：如果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 逆向文件频率(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。

但是实际上，如果一个词条在一个类的文档中频繁出现，则说明该词条能够很好代表这个类的文本的特征，这样的词条应该给它们赋予较高的权重，并选来作为该类文本的特征词以区别与其它类文档。这就是 IDF 的不足之处. 在一份给定的文件里，词频（term frequency，TF）指的是某一个给定的词语在该文件中出现的频率。这个数字是对词数(term count)的归一化，以防止它偏向长的文件。（同一个词语在长文件里可能会比短文件有更高的词数，而不管该词语重要与否。）对于在某一特定文件里的词语来说，它的重要性可表示为：

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

以上式子中分子是该词在文件中的出现次数，而分母则是在文件中所有字词的出现次数之和。

逆向文件频率（inverse document frequency，IDF）是一个词语普遍重要性的度量。某一特定词语的 IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取以 10 为底的对数得到：

$$\mathrm{idf}_i = \lg \frac{|D|}{|\{j : t_i \in d_j\}|}$$

其中 |D|：语料库中的文件总数

包含词语的文件数目（即的文件数目）如果该词语不在语料库中，就会导致分母为零，因此一般情况下使用$1 + |\{d \in D : t \in d\}|$作为分母。然后再计算 TF 与 IDF 的乘积

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$$

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的 TF-IDF。因此，TF-IDF 倾向于过滤掉常见的词语，保留重要的词语。

此模型可将 TF-IDF 模型纳入其中，一方面解释其合理性，另一方面也发现了其不完善之处。另外，此模型还可以解释 PageRank 的意义，以及 PageRank 权重和 TF-IDF 权重之间为什么是乘积关系。[2]

本作品中这里直接使用 TF-IDF 模型进行构建，并最终转成 numpy 的数组类型，示例代码如下：

```
# TF-IDF
X_tr = TfidfTransformer().fit_transform(data_tr.toarray()).toarray()
X_te = TfidfTransformer().fit_transform(data_te.toarray()).toarray()

```

2.3.3 线性支持向量机

支持向量机（Support Vector Machine, SVM）是一类按监督学习（supervised learning）方式对数据进行二元分类的广义线性分类器(generalized linear classifier)，其决策边界是对学习样本求解的最大边距超平面（maximum-margin hyperplane）[3-5]。SVM 使用铰链损失函数（hinge loss）计算经验风险（empirical risk）并在求解系统中加入了正则化项以优化结构风险(structural risk)，是一个具有稀疏性和稳健性的分类器[4]。SVM 可以通过核方法(kernel method)进行非线性分类，是常见的核学习(kernel learning)方法之一[6]。SVM 被提出于 1964 年，在二十世纪 90 年代后得到快速发展并衍生出一系列改进和扩展算法，在人像识别、文本分类等模式识别（pattern recognition）问题中有得到应用[7-8]。

1. 硬边距（hard margin）

给定输入数据和学习目标：$\boldsymbol{X}=\{\boldsymbol{X}_1,\ldots,\boldsymbol{X}_N\}$, $\boldsymbol{y}=\{y_1,\ldots,y_N\}$ 硬边界 SVM 是在线性可分问题中求解最大边距超平面（maximum-margin hyperplane）的算法，约束条件是样本点到决策边界的距离大于等于 1。硬边界 SVM 可以转化为一个等价的二次凸优化（quadratic convex optimization）问题进行求解。

$$\begin{aligned}&\max_{\boldsymbol{w},b}\quad \frac{2}{\|\boldsymbol{w}\|}\\&\text{s.t.}\quad y_i\left(\boldsymbol{w}^\top\boldsymbol{X}_i+b\right)\geq 1\end{aligned}\quad\Longleftrightarrow\quad\begin{aligned}&\min_{\boldsymbol{w},b}\quad \frac{1}{2}\|\boldsymbol{w}\|^2\\&\text{s.t.}\quad y_i\left(\boldsymbol{w}^\top\boldsymbol{X}_i+b\right)\geq 1\end{aligned}$$

由上式得到的决策边界可以对任意样本进行分类：$\operatorname{sign}\left[y_i\left(\boldsymbol{w}^\top\boldsymbol{X}_i+b\right)\right]$。注意到虽然超平面法向量$\boldsymbol{w}$是唯一优化目标，但学习数据和超平面的截距通过约束条件影响了该优化问题的求解[4]。硬边距 SVM 是正则化系数取 0 时的软边距 SVM，其对偶问题和求解参见软边距 SVM，这里不额外列出。

2. 软边距（soft margin）

在线性不可分问题中使用硬边距 SVM 将产生分类误差，因此可在最大化边距的基础上引入损失函数构造新的优化问题。SVM 使用铰链损失函数，沿用硬边界 SVM 的优化问题形式，软边距 SVM 的优化问题有如下表示：

$$\begin{aligned}&\min_{\boldsymbol{w},b}\quad \frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{N}L_i,\quad L_i=\max\left[0,1-y_i\left(\boldsymbol{w}^\top\boldsymbol{X}_i+b\right)\right]\\&\text{s.t.}\quad y_i\left(\boldsymbol{w}^\top\boldsymbol{X}_i+b\right)\geq 1-L_i,\quad L_i\geq 0\end{aligned}$$

上式表明可知，软边距 SVM 是一个 L_2 正则化分类器，式中 L_i 表示铰链损失函数。使用松弛变量：$\zeta \geq 0$ 处理铰链损失函数的分段取值后，上式可化为：

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i\left(\boldsymbol{w}^\top \boldsymbol{X}_i + b\right) \geq 1-\xi_i, \quad \xi_i \geq 0 \end{aligned}$$

求解上述软边距 SVM 通常利用其优化问题的对偶性（duality），这里给出推导：

定义软边距 SVM 的优化问题为原问题（primal problem），通过拉格朗日乘子（Lagrange multiplier）：

$$\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_N\},\ \boldsymbol{\mu} = \{\mu_1, \ldots, \mu_N\}$$

可得到其拉格朗日函数 [4][9]

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i\left[1-\xi_i - y_i\left(\boldsymbol{w}^\top \boldsymbol{X}_i + b\right)\right] - \sum_{i=1}^{N}\mu_i\xi_i$$

令拉格朗日函数对优化目标 $\boldsymbol{w}, b, \boldsymbol{\xi}$ 的偏导数为 0，可得到一系列包含拉格朗日乘子的表达式 [4][9]：

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N}\alpha_i y_i \boldsymbol{X}_i, \quad \frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N}\alpha_i y_i = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} = 0 \Rightarrow C = \alpha_i + \mu_i$$

将其带入拉格朗日函数后可得原问题的对偶问题（dual problem）[4] [9]：

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\alpha_i y_i (\boldsymbol{X}_i)^\top (\boldsymbol{X}_j) y_j \alpha_j\right] \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \end{aligned}$$

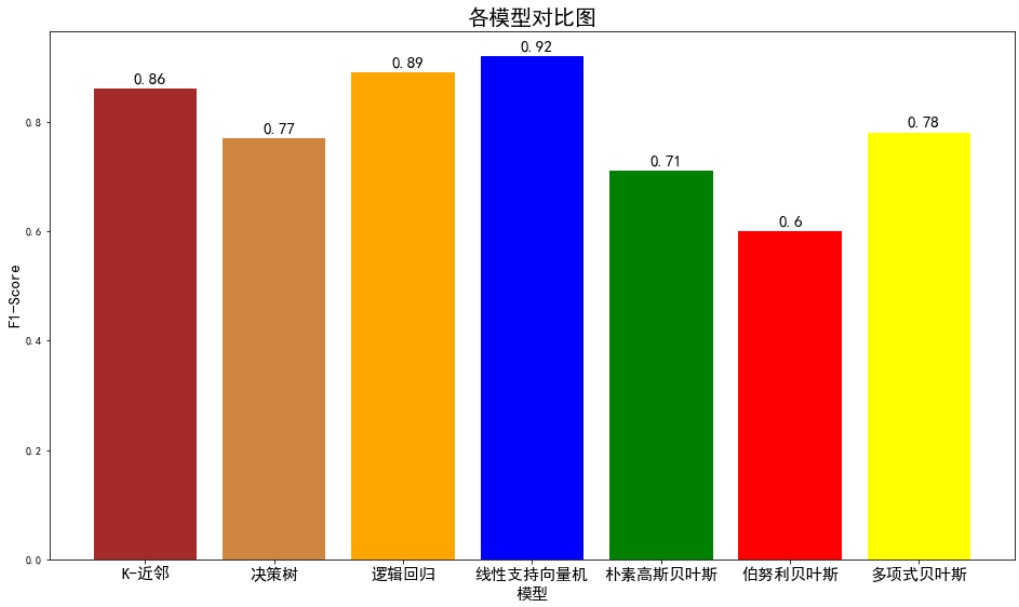
对偶问题的约束条件中包含不等关系，因此其存在局部最优的条件是拉格朗日乘子满足 Karush-Kuhn-Tucker 条件(Karush-Kuhn-Tucker condition,KKT)[4]：

$$\begin{cases} \alpha_i \geq 0, \quad \mu_i \geq 0 \\ \xi_i \geq 0, \quad \mu_i\xi_i = 0 \\ y_i\left(\boldsymbol{w}^\top \boldsymbol{X}_i + b\right) - 1 + L_i \geq 0 \\ \alpha_i\left[y_i\left(\boldsymbol{w}^\top \boldsymbol{X}_i + b\right) - 1 + L_i\right] = 0 \end{cases}$$

由上述 KKT 条件可知，对任意样本 $(\boldsymbol{X}_i, y_i)$，总有 $\alpha_i = 0$ 或 $y_i\left(\boldsymbol{w}^\top \boldsymbol{X}_i + b\right) = 1-\xi_i$，对前者，该样本不会对决策边界 $\boldsymbol{w}^\top \boldsymbol{X}_i + b = 0$ 产生影响，对后者，该样本满足 $y_i\left(\boldsymbol{w}^\top \boldsymbol{X}_i + b\right) = 1-\xi_i$ 意味其处于间隔边界上（$\alpha_i < C$）、间隔内部（$\alpha_i = C$）或被错误分类（$\alpha_i > C$），即该样本是支持

向量。由此可见，软边距 SVM 决策边界的确定仅与支持向量有关，使用铰链损失函数使得 SVM 具有稀疏性[4]。

本作品在使用该模型前进行了几个模型的对比，最终该模型表现最好。



使用 scikit-learn 库 svm 模块下的 LinearSVC 类进行模型的构建，示例代码如下：

```
# 建立模型 线性支持向量机
model = LinearSVC()
model.fit(X_tr, labels_tr)
```

2.3.4 F1-Score 模型评估

精确率(Precision)和召回率(Recall)评估指标,理想情况下做到两个指标都高当然最好，但一般情况下，Precision 高，Recall 就低，Recall 高，Precision 就低。所以在实际中常常需要根据具体情况做出取舍，例如一般的搜索情况，在保证召回率的条件下，尽量提升精确率。而像癌症检测、地震检测、金融欺诈等，则在保证精确率的条件下，尽量提升召回率。

引出了一个新的指标 F-score,综合考虑 Precision 和 Recall 的调和值

$$F-Score=(1+\beta^2)\cdot\frac{Precision\cdot Recall}{\beta^2\cdot Precision+Recall}$$

•当 β =1 时，称为 F1-score 或者 F1-Measure，这时，精确率和召回率都很重要，权重相同。

•当有些情况下，我们认为精确率更重要些，那就调整 β 的值小于 1，

•如果我们认为召回率更重要些，那就调整 β 的值大于 1。

F1 指标（F1-score）:F1-score 表示的是 precision 和 recall 的调和平均评估指标。

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

本作品直接调用

```
# 预测
pred = model.predict(X_te)
# 模型评估
print(classification_report(labels_te, pred))

```

2.4 结果展示

第一问的结果如图：

	precision	recall	f1-score	support
交通运输	0.92	0.82	0.86	109
劳动和社会保障	0.93	0.97	0.95	407
卫生计生	0.95	0.89	0.92	161
商贸旅游	0.92	0.87	0.90	237
城乡建设	0.89	0.94	0.91	399
教育文体	0.96	0.93	0.94	332
环境保护	0.93	0.96	0.94	197
avg / total	0.93	0.93	0.92	1842

3 热点问题挖掘

3.1 数据预处理

为实现数据预处理，主要步骤如下：

（1）将备注字段转为字符串类型。

（2）使用正则表达式去除干扰字符。

（3）使用 jieba 进行分词。

（4）将停用词导入，去除停用词。

由于这步的数据预处理与上面的类似，这里就不在赘述。

```
# 导入相关的库
import re
import jieba
import jieba.analyse
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
```

数据预处理

```
# 导入数据
data = pd.read_excel('./Data/附件3.xlsx')
# 将留言主题和留言详情合并到message中
data_after = data['留言主题'] + data['留言详情']
# 使用正则表达式去除干扰字符
data_re = data_after.apply(lambda x : re.sub('\s|[0-9a-zA-Z]', '', x))
# 使用jieba进行分词
data_cut = data_re.apply(lambda x : jieba.lcut(x))
# 将停用词导入
stopWords = pd.read_csv('./Data/stopword.txt', header=None, sep='hhhhhhhhhhhhh')
stopWords = list(stopWords.iloc[:, 0])
# 去除停用词
data_stop_words = data_cut.apply(lambda x : [i for i in x if i not in stopWords])
data_new = data_stop_words.apply(lambda x: ' '.join(x))
```

3.2 模型构建

同样的，这里使用 TF-IDF 模型，然后使用快速聚类算法 K-means 进行聚类。

```
# TF-IDF
tfidfVectorizer = TfidfVectorizer(max_df=0.8, max_features=200000,
                                  min_df=0.2, stop_words='english',
                                  use_idf=True, ngram_range=(1,3))
tfidfMatrix = tfidfVectorizer.fit_transform(data_new)
# K-means聚类
n = 1000
model = KMeans(n_clusters=n)
model.fit(tfidfMatrix)

```

3.3 热点问题明细表

先将模型聚类好的标签转成一个列表，这个列表就是我们想要得到的问题 ID 列，所以直接将这个列表作为数据插入到原始的数据中，这样就构建好了热点问题留言明细表。该布的示例代码如下所示：

```
# 聚类标签
labels = (model.labels_ + 1).tolist()
# 将标签插入到原始数据中
data.insert(0, '问题ID', labels)
# 将数据用‘问题ID’列排序
data_after = data.sort_values(by='问题ID')
# 保存热点问题留言明细表
data_after.to_excel('./Data/热点问题留言明细表.xls', index=False)
```

3.4 热点问题表

3.4.1 问题 ID 列、问题描述列

这一步要构建热点问题表，首先我们通过聚类好的标签，也就是问题 ID 列进行分组，该表格的热度指数列我们设置为点赞数与反对数的差（这个值可能会是负数，这是由于反对数比点赞数多），这样就可以显示出不同热点问题的热度，目前的热度指数列还是数字，为之后修改做好铺垫。

问题描述列，我们将分好组的每一组进行拼接，形成一个字符串，每一个标签对应一个字符串，然后对这些字符串进行去除无用字符，进行分词处理，然后去除停用词，最后我们得到的是一个个的列表，最后将这些列表对应的拼接成字符串，每个词语之间是用空格来分隔，之后在用 jieba 库的 analyse.extract_tags()函数进行提取 6 个出现频次最高的词语，作为该表格的问题描述列。

该步的示例代码如下所示：

```
# 将'留言时间'列的类型转为datetime
data_after['留言时间'] = pd.to_datetime(data_after['留言时间'])
# 将‘问题ID’列分组
dataGroup = data_after.groupby(by='问题ID')
# 指定热度指数
data_hot_index = dataGroup[['点赞数', '反对数']].agg(np.sum)
# 创建热点问题表
data_hot = pd.DataFrame()
# 热点问题表'热度指数'列
data_hot['热度指数'] = data_hot_index['点赞数'] - data_hot_index['反对数']
# 热点问题表'问题ID'列
data_hot.insert(0, '问题ID', range(1, n+1))
# 热点问题表'问题描述'列
data_describe = dataGroup['留言主题'].apply(lambda x : ''.join(x))
data_describe = data_describe.apply(lambda x : re.sub('\s|[0-9a-zA-Z]', '', x)) # 去除英文字母和数字
data_describe = data_describe.apply(lambda x : jieba.lcut(x)) # 分词
data_describe = data_describe.apply(lambda x : [i for i in x if i not in stopWords]) # 去除停用词
data_describe = data_describe.apply(lambda x : ' '.join(x)) # 将列表转成字符串
data_describe = data_describe.apply(lambda x : ' '.join(jieba.analyse.extract_tags(x, topK=6, withWeight=False)))
data_describe = pd.DataFrame(data_describe)
data_describe.columns = ['问题描述']
data_hot = pd.concat([data_hot, data_describe], axis=1)
```

3.4.2 时间范围

该表格的时间范围列要通过 pandas 库的时间序列类型来进行操作，在上一步的代码中，我们先将原始数据的留言时间列都转成了 datetime 类型，之后将分组后的留言时间列选出最晚的时间和最早的时间，这里我们使用 numpy 库的 max()函数和 min()函数进行选择，现在得到了两列时间数据，然后在将这两列转成题目所需要的格式，之后把这两列数据拼接起来形成一个新的时间列，并加入到热点问题表中。

该步的示例代码如下所示：

```
# 找到每组最大和最小时间
date_max = dataGroup['留言时间'].apply(lambda x : np.max(x))
date_min = dataGroup['留言时间'].apply(lambda x : np.min(x))
# 将每组的最大时间和最小时间格式转成想要的格式
date_max = date_max.apply(lambda x : str(x))
date_max = date_max.apply(lambda x : x[:-9])
date_max = date_max.apply(lambda x : re.sub('-', '/', x))
date_min = date_min.apply(lambda x : str(x))
date_min = date_min.apply(lambda x : x[:-9])
date_min = date_min.apply(lambda x : re.sub('-', '/', x))
# 将date列插入到热点问题表'时间范围'列
date = date_min + ' 至 ' + date_max
data_hot.insert(2, '时间范围', date.tolist())

```

3.4.3 热度指数列

本作品先将数据用热度指数进行降序排序，热度最高的问题就在最上面。

```
# 用热度指数排名
data_hot = data_hot.sort_values(by='热度指数', ascending=False)
```

之后定义了一个用来划分数据范围的函数。该函数的传入参数为数据的列表形式和范围划分的个数。首先计算出数据的最大值和最小值，放在一个范围的列表中，因为是留言问题，上面使用的点赞数与反对数的差，所以这里的第一次划分范围我们使用的 0，即得到的新列表，之后我们在对小范围进行划分，0 的左侧固定划分为 2 个，右侧划分为指定范围划分个数与 2 的差，并且划分的标准是使用中位数，这样划分更合理，该定义函数如下所示：

```
# 定义划分范围函数，参数list_为数据的列表形式，n为范围划分个数
def divide(list_, n):
    min = np.min(list_)
    max = np.max(list_)
    middle = [min, 0, max]

    n -= 2
    for i in range(1, n+1):
        if i <= 1:
            minlist1 = [i for i in list_ if i < middle[1]]
            middle.insert(1, np.median(minlist1))
        else:
            minlist2 = [i for i in list_ if i > middle[-2]]
            middle.insert(-1, np.median(minlist2))

    return middle
```

对于此题，我们一共将数据进行划分 10 类，并且使用星来代表热度的等级，指定该列为热度星级列，并将该列加到热度指数列的后方，使得数字更加直观。最后我们新加入了一个热度排名列，然后将表格保存。该部分示例代码如下：

```python
# 将数据转成列表
hot_list = data_hot['热度指数'].tolist()
# 划分范围
di = divide(hot_list, 10)
# 把热度指数用星星表示，并放在列表中
hot_str = []
for i in range(len(di)-1, -1, -1):
    for j in hot_list:
        if di[i-1] < j <= di[i]:
            hot_str.append('★' * i)
hot_str.append('★')
# 将星级列表加入热点问题表中
data_hot.insert(2, '热度星级', hot_str)
# 热点问题表'热度排名'列
data_hot.insert(0, '热度排名', range(1, data_hot.shape[0] + 1))
# 保存热点问题表
data_hot.to_excel('./Data/热点问题表.xls', encoding='utf-8', index=None)
```

3.5 结果展示

第二问的结果为两个表格，如下所示：

热点问题表.xls

热度排名	问题ID	热度指数	热度星级	时间范围	问题描述
1	712	2096	★★★★★★★★★★	2019/01/21 至 2019/08/19	小区 别墅 群私 搭乱建 市鑫 天鑫城
2	1	1887	★★★★★★★★★★	2017/06/08 至 2020/01/08	西地省 咨询 建议 地铁 扰民 投诉
3	126	1757	★★★★★★★★★	2019/01/08 至 2019/06/06	入学 低质 区青园 市旭辉 金毛 精装
4	32	856	★★★★★★★★★	2019/01/19 至 2019/12/22	暮云 六路 街道 希望 扰民 新村
5	779	735	★★★★★★★★★	2019/03/01 至 2019/12/15	镇政府 公证 青山 改道 国道 拆迁

热点问题留言明细表.xls

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	192257	A00031475	举报A3区星雨驾校退费的霸王条款及退费中产生偷税漏税的行为	2019/12/16 23:00:48	尊敬的胡书记，您好！我于201	0	2
2	188006	A000102948	A3区一米阳光婚纱艺术摄影是否合法纳税了?	2019/2/28 11:25:05	座落在A市A3区联丰路米兰春天	0	0
2	234873	A00034753	A市702公交车难等	2019/7/8 18:49:01	中建五局到A市理工段，30多分	0	0
2	234707	A00059799	西地省有没有可能实行每周"2.5天小长假"的休假模式?	2019/1/24 9:10:26	最近看到新闻说河北有印发文件	1	0
2	234633	A909194	无视消费者权益的A市伊景园滨河苑车位捆绑销售行为	2019-08-20 12:34:20	伊景园滨河苑项目商品房，广告	0	0
2	234506	A00028641	A5区古曲南路广益中学门口脏乱差	2019/4/8 17:30:19	之前反应的问题一点都没有改变	0	0
2	234468	A00074236	A市贺隆体育馆是否有启用的计划安排?	2019/1/14 20:08:00	贺隆体育馆总是闲置在那里也不	0	0
2	234406	A000102981	现在A6区原高塘岭镇范围内的四轮电动车遍地开花	2019/12/5 8:49:00	现在A6区原高塘岭镇范围内城区	0	0
2	234360	A000102307	咨询A市调整用电基数问题	2019/3/6 11:38:16	目前，常住人口是6人，电	0	1
2	234320	A000106592	不要让A市因为58车贷案件而臭名远扬	2019/7/8 17:16:57	胡书记：您好反映关于西地省A	0	0
2	234309	A00042290	A8县灰汤镇雅乐居C10栋1109房阳台悬空梁已断	2019/8/10 13:26:50	A8县市灰汤镇雅乐居C10栋110	0	0
2	234294	A00094433	判决书已生效，多次去到A2区法院都无法立案	2019/12/17 11:02:55	胡市长，你好，西地省A市A2区	0	0
2	234004	A00031618	A6区银杉路两旁街道破烂不堪	2019/3/15 7:36:08	地处A市A6区主干道银杉路两旁	0	6
2	233897	A00072330	A市医保局对肾衰竭患者的医保政策调整不合理	2019/1/7 11:50:28	众所周知，国家对大病医疗保障	0	0
2	233717	A00061821	为什么西地省农机购置补贴结算时间差距很大?	2019/6/14 13:15:01	西地省农机局在C市九华工业园	0	0
2	233630	A000107064	咨询退伍兵转业异地安置入户A市的问题	2019/12/16 20:38:02	您好！我是一名今年将转业回乡	0	0
2	233502	A00041172	A市青竹湖瞰园开发商强买强卖，拒不退还购房首付款	2019/12/24 23:12:35	本人是一名普通购房者，与今年	0	0

4 答复意见的评价

根据对相关部门留言答复的相关性、完整性、可解释性等方面，我们认为应该先对留言进行预处理，使用jieba库等进行分析处理，然后去除停用词，之后对词进行向量化处理，对相关部门留言的答复也进行这样的预处理操作。

使用word2vec进行文本的相似度处理，看看答复意见与留言的相似度是否够高，这样就可以看出留言答复的相关性，同时也可以反映出留言答复的可解释性，但是这样还是不够的，还是比较的局限，所以，进一步要进行命名实体识别，看看留言答复是否与留言一致，可以反应出答复意见的完整性。

总结

本次文本挖掘中，中文文本不像英文文本那样有空格隔开，所以我们用到了 jieba 分词库来完成分词，在进行数据预处理时，由于非文本内容相对少些，我们直接用 Python 的正则表达式(re)进行删除。先使用 jieba 进行分词，然后导入停用词表并删除停用词，去除掉那些没用的文本的内容后，我们进行真正的文本预处理，之后将数据预处理代码封装成一个函数，供后面的词云绘制和分类操作进行调用。但是在问题二中要进行聚类并且之后要构建表格，所以第二问的数据预处理就没有封装成函数。

在文本挖掘预处理之 TF-IDF 中，使用 scikit-learn 的 TfidfVectorizer 类来进行 TF-IDF 特征处理，使用线性支持向量机模型构建分类模型，并且进行了 F1-Score 模型评估。分析了热点问题并构建了热点问题的两个表格，对答复意见进行相应的评价，根据答复意见的质量给出相应的评价方案。后期我们需要对文本挖掘进行深度的学习与探讨。

参考文献

[1] TF-IDF.站长之家[引用日期 2013-06-10]
[2] TF-IDF 模型的概率解释.博客园[引用日期 2013-06-05]
[3] Vapnik, V.. Statistical learning theory. 1998 (Vol. 3). New York, NY：Wiley，1998：Chapter 10-11, pp.401-492
[4] 周志华. 机器学习. 北京：清华大学出版社，2016：pp.121-139, 298-300
[5] 李航. 统计学习方法. 北京：清华大学出版社，2012：第七章，pp.95-135
[6] Hsieh, W.W.. Machine learning methods in the environmental sciences: Neural networks and kernels：Cambridge university press，2009：Chapter 7, pp.157-169
[7] Qin, J. and He, Z.S., 2005, August. A SVM face recognition method based on Gabor-featured key points. In Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on (Vol. 8, pp. 5144-5149). IEEE.
[8] Sun, A., Lim, E.P. and Ng, W.K., 2002, November. Web classification using support vector machine. In Proceedings of the 4th international workshop on Web information and data management (pp. 96-99). ACM.
[9]Friedman, J., Hastie, T. and Tibshirani, R.. The elements of statistical learning (Vol. 1, No. 10). New York, NY：Springer，2001：Chapter 12, pp.417-438