一种基于自然语言处理技术的文本挖掘和文本分析模型

摘要

随着互联网时代的到来，移动通讯设备的普及，现在大家喜欢通过各种网络问政平台来了解国家大事，并且在上面发表自己的建议。随着网络问政的人数不断增加，对于这一类的信息处理强度也随之越来越大。

近年来，自然语言处理 (NLP) 在人工智能领域飞速发展，对此，我们想要通过构建基于自然语言处理技术的文本挖掘和文本分析模型来解决以上问题。本文可以根据任务分为以下三个部分：

在任务一中，为了建立一级标签，我们对所给附件中的数据进行预处理，然后，进行 $TF-IDF$ 关键词匹配。接下来，我们结合朴素贝叶斯分类器，初步建立线性支持向量机模型。基于初步建立的模型，我们对其进行模型训练。最后，我们采用 $F-Score$ 对模型进行检验，得出模型可用的结论，成功解决群众留言问题。

在任务二中，我们对相似问题进行归类，通过 $kmeans$ 聚类的方法对其矩阵数据按照附件一的三级标签个数进行聚类，将聚类后每一簇留言主题对应的留言详情的文本数据合并在一起，对文本数据进行提取摘要。随后做文本相似度分析，相似度越高则证明该问题被提及次数越多，最终对结果进行排列。

在任务三中，根据判断，我们选择从留言时间与答复时间的平均所需时间、整体的留言详情和答复意见的文本相似度、留言详情的关键词与答复意见的关键词之间的文本相似度三个方面进行分析。首先我们运用了 $python$ 中的 $pandas$ 函数进行时间差的分析并对其进行数据可视化。紧接着，我们运用 $TextRank$ 算法进行关键词的提取，随后运用余弦相似度算法进行文本相似度的计算，最后得出答复意见中没有很好的结合群众的留言，在其角度提供意见的结论。

对于此次建立的模型，我们会不断进行优化和改良，希望能够帮助有关部门更好的处理有关的文本数据，方便了解民意，做出更好的决策。

关键词： 自然语言处理 (NLP)　$F-Score$　$TextRank$ 算法　余弦相似度算法

A text mining and text analysis model based on natural language processing technology

With the advent of the Internet era and the popularity of mobile communication devices, people now like to ask political platforms through various networks to learn about national affairs and express their own Suggestions on them.With the increasing of the number of people asking about politics on the Internet, the intensity of this kind of information processing is also increasing.

In recent years, natural language processing (NLP) has been developing rapidly in the field of artificial intelligence. Therefore, we want to build a text mining and text analysis model based on natural language processing technology to solve the above problems.This paper can be divided into the following three parts according to the task:

In task 1, we preprocessed the data in the attachment given, and then established a level 1 label.Then, $TF-IDF$ keyword matching was carried out, and the linear support vector machine model was preliminarily established by combining with naive bayesian classifier.Then, model training was carried out on the obtained model. Finally, $F-score$ was used to test the model, and the conclusion that the model was available was obtained.We successfully solved the problem of people's comments.

In task 2, we classified similar problems, clustered its matrix data according to the number of three-level labels in annex I through $kmeans$ clustering method, combined the text data of message details corresponding to the message topics of each cluster after clustering, and extracted a summary of the text data.Then we conduct similarity analysis on the text. The higher the similarity is, the more the problem is mentioned. Finally, we arrange the results.

In task 3, the paper analyzes from three aspects: the average required time between message time and response time, the text similarity between overall message details and response comments, and the text similarity between the keywords of message details and response comments.First, we used the $pandasfunction$ in $python$ to analyze the time difference and visualize its data.Then, we used $TextRank$ algorithm to extract the keywords, and then used cosine similarity algorithm to calculate the text similarity. Finally, we came to the conclusion that there was no good combination of the comments of the masses in the replies, and the opinions were provided from their perspectives.

As for the model established this time, we will continue to optimize and improve it, hoping to help relevant departments better process relevant text data, facilitate understanding of public opinions and make better decisions.

Keywords: Natural language processing (NLP) $F-Score$ $TextRank$ Algorithm Cosine similarity algorithm

目录

一、 引言

随着互联网的高速发展以及智能设备的普及，近年来，微信、微博、市长邮箱、阳光热线等网络问政平台成为了政府了解民意、汇聚民智、凝聚民气的重要渠道，各类相关文本信息数据量不断攀升，为了方便信息的寻找和处理，我们需要进行群众留言的分类、总结群众所关心的热点问题、对群众的留言和意见进行回复等工作。若还像以前一样依靠人工来进行留言的划分和热点整理，将会是一个极大的挑战。随着大数据、云计算、人工智能等技术的发展，基于自然语言处理技术的智慧政务系统来解决了上述现有的问题，对提升政府的管理水平和施政效率具有极大的推动作用。

基于对自然语言处理技术的理解以及上述背景，本文中我们将通过附件中所给出的自互联网公开来源的群众问政留言记录，以及相关布恩岁部分群众的答复意见，构建一种基于自然语言处理技术的文本挖掘和文本分析模型解决群众留言分类、热点问题挖掘和答复意见的评价问题。

二、 实验环境

2.1 实验环境

在我们的模型验证过程中，我们以 Windows 为主要操作系统、以 Python 为开发环境于内存为 128G 的容量、8T 的固态硬盘的实验环境下完成模型的构建。

三、 任务一

3.1 数据分析与预处理

我们通过在网上收集相关的资料（百度停用词表，四川大学停用词表，哈工大停用词表等四个停用词表），随后将其进行组合形成新的停用词表后进行数据的预处理。

3.1.1 数据分析

首先，我们对附件 2 中的数据进行了观察，发现若我们需要建立一级标签，我们只需要建立’ 留言主题’,’ 留言详情’,’ 一级标签’ 的关系即可。随后我们读取其三个指标的数据并建立 dataframe，计算得到数据总量，随机选取 10 行查看结果（如图 1）

紧接着，我们使用 python 检查三个指标数据是否存在空值（如图 2）:

为了方便后面的操作，我们将对应的指标用相对应的英文进行表示（如图 3）。

数据总量：9210 .

Out[2]:

	留言主题	留言详情	一级标签
2057	就火车噪音扰民问题进一言，望采纳	常言道"火车叫声传十里，声声震耳吓死人"...	环境保护
5213	请问A市社保部人员失业保险申请需准备什么资料？	A市社保部人员不作为，作为一个残疾人，行...	劳动和社会保障
8248	考导游证在A市哪里报名？	我想考导游证；请问在A市什么地方报名？谢...	商贸旅游
5242	A4区临时环卫工退休工资为何一月才640元？	市长，A4区临时环卫工退休工资一月才64...	劳动和社会保障
6309	投诉K1区第一公共交通公司辞退员工，损害职工利益	本人是K市K1区第一公共交通公司辞退员工，本人在该公司从事一...	劳动和社会保障
7100	请关注今年进工商局系统的正营职转业干部的编制身份	尊敬的胡厅长：　您好！我是根据国家转业政策于2011年度转业的...	劳动和社会保障
5639	C市粒粒珍楚莲有限公司莫名辞退我，不与我签订劳动合同	我从2013.3月在现在C市粒粒珍楚莲有限公司（河口）这家公...	劳动和社会保障
7278	B7县红砖协会、砂石协会涉嫌价格垄断	垄断势力与黑势力悄悄地走进西地省B7县老...	商贸旅游
3061	D6县客运一体化何时实施	交通部2016年下文，要在全国推行城乡客运一...	交通运输
1883	I4县鸬鹚渡镇违建何时能拆除	I4县鸬鹚渡镇国土、城建办二部门不作为、渎职...	城乡建设

图 1　处理后随机选取 10 行结果

```
print("在 留言主题 列中总共有 %d 个空值." % df['留言主题'].isnull().sum())
print("在 留言详情 列中总共有 %d 个空值." % df['留言详情'].isnull().sum())
print("在 一级标签 列中总共有 %d 个空值." % df['一级标签'].isnull().sum())
```

```
在 留言主题 列中总共有 0 个空值.
在 留言详情 列中总共有 0 个空值.
在 一级标签 列中总共有 0 个空值.
```

图 2　是否存在空值结果

```
#为了后面更好的操作，我们将对应的列名换成英文
df.rename(columns={'留言主题': 'review_head', '留言详情' : 'review_body', '一级标签' : 'first_label'}, inplace = True)
df.sample(5)
```

	review_head	review_body	first_label
1554	请你关住一下M7市五里牌公租房工程	尊敬的领导：　请你关住一下M7市五里牌公租...	城乡建设
8922	A市妇幼保健院对待胎死腹中事故置之不理	本人自2016年2月15日确定怀孕，2016年3月3日起在A...	卫生计生
3760	E市民办代课教师的补偿何日到位	你好。民办代课教师的情况6月份进行了填表调查，E市的没有动静，有没有补偿？...	教育文体
6355	我兄退休前死亡个人缴费子女能继承吗？	尊敬的贺厅长：　您好！我是C3县易俗河镇转城户人员，现住花...	劳动和社会保障
5011	G5县澧阳平原申遗有进展吗？	做为家乡人，看了看到杭州良渚文化遗址申请世界遗址成功，我就想到我...	教育文体

图 3　用英文替换后结果

3.1.2 数据可视化

为了更好的了解级标签的数量，我们统计了每个标签的数据并使用直方图进行了数据可视化分析。（如图 4、5）

	first_label	count
0	城乡建设	2009
1	劳动和社会保障	1969
2	教育文体	1589
3	商贸旅游	1215
4	环境保护	938
5	卫生计生	877
6	交通运输	613

图 4　一级标签数量统计

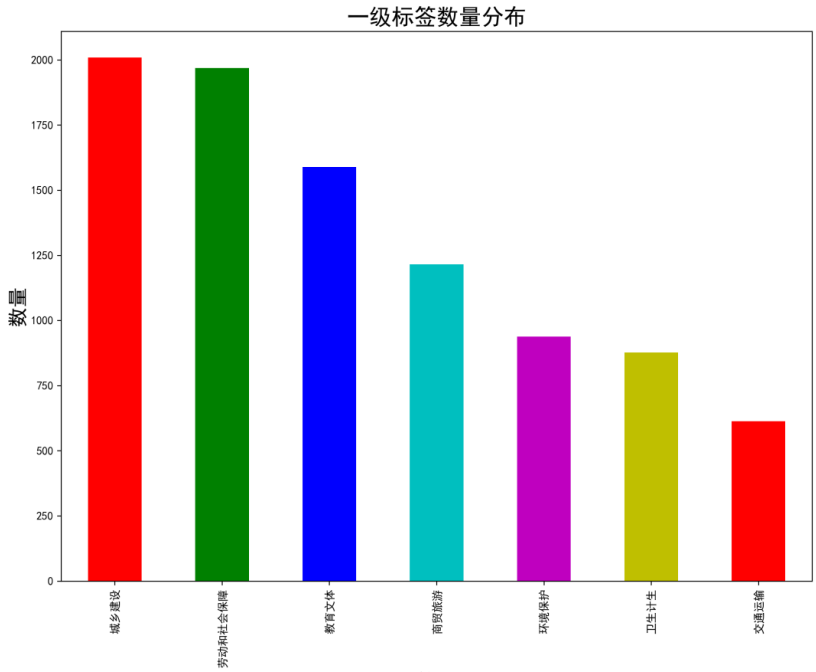

图 5　一级标签数量分布直方图

随后，我们给文本标签加上数字化的标签，并设计分词函数对 $review-head$，$review-body$ 的文本数据进行分词并去除除字母, 数字，汉字以外的所有符号，将经过文本初步处理后的 $review-head$,$review-body$ 合并文本为 $clean-review$，最后再对 clean-review 进一步进行去停用词处理得 $cut-review$。（如图 6）

	review_head	review_body	first_label	label_id	clean_review_body	clean_review_head	clean_review	cut_review
0	A市西湖建筑集团占道施工有安全隐患	\n\n\n\n\n\n\n\n\n\nA3区大道西行便道，未管所路口至加油站路段，...	城乡建设	0	A3区大道西行便道未管所路口至加油站路段人行道包括路灯杆被圈西湖建筑集团燕子山安置房项目施工...	A市西湖建筑集团占道施工有安全隐患	A市西湖建筑集团占道施工有安全隐患A3区大道西行便道未管所路口至加油站路段人行道包括路灯杆被...	A 市 西湖 建筑 集团 占道 施工 安全 隐患 A3 区 大道 西行 便道 未管 所路口 加...
1	A市在水一方大厦人为烂尾多年，安全隐患严重	\n\n\n\n\n\n\n\n\n\n位于书院路主干道的在水一方大厦一楼至四楼人为...	城乡建设	0	位于书院路主干道的在水一方大厦一楼至四楼人为拆除水电等设施后烂尾多年用护栏围着不但占用人行道...	A市在水一方大厦人为烂尾多年安全隐患严重	A市在水一方大厦人为烂尾多年安全隐患严重位于书院路主干道的在水一方大厦一楼至四楼人为拆除水电...	A 市 在水一方 大厦 人为 烂尾 多年 安全 隐患 严重 位于 书院 路 主干道 在水一方 大厦 ...
2	投诉A市A1区苑物业违规收停车费	\n\n\n\n\n\n\n\n\n\n尊敬的领导：A1区苑小区位于A1区火炬路，小...	城乡建设	0	尊敬的领导A1区苑小区位于A1区火炬路小区物业A市程明物业管理有限公司未经小区业主同意利用业...	投诉A市A1区苑物业违规收停车费	投诉A市A1区苑物业违规收停车费尊敬的领导A1区苑小区位于A1区火炬路小区物业A市程明物业管...	投诉 A 市 A1 区 苑 物业 违规 收 停车费 尊敬 领导 A1 区 苑 小区 位于 A1 ...
3	A1区蔡锷南路A2区华庭楼顶水箱长年不洗	\n\n\n\n\n\n\n\n\n\nA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设	0	A1区A2区华庭小区高层为二次供水楼顶水箱长年不洗现在自来水龙头的水严重霉味大家都知道水是我...	A1区蔡锷南路A2区华庭楼顶水箱长年不洗	A1区蔡锷南路A2区华庭楼顶水箱长年不洗A1区A2区华庭小区高层为二次供水楼顶水箱长年不洗现...	A1 区 蔡锷 南路 A2 区 华庭 楼顶 水箱 长年 不洗 A1 区 A2 区 华庭 小区 高...
4	A1区A2区华庭自来水好大一股霉味	\n\n\n\n\n\n\n\n\n\nA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设	0	A1区A2区华庭小区高层为二次供水楼顶水箱长年不洗现在自来水龙头的水严重霉味大家都知道水是我...	A1区A2区华庭自来水好大一股霉味	A1区A2区华庭自来水好大一股霉味A1区A2区华庭小区高层为二次供水楼顶水箱长年不洗现在自来...	A1 区 A2 区 华庭 自来水 好大 一股 霉味 A1 区 A2 区 华庭 小区 高层 二次 ...

图 6　文本最终处理结果展示

通过词云图对 7 个标签，每类标签的 100 个高频词进行可视化，再而通过 $TfidfVectorizer$ 计算词的特征向量并对其进行卡方检验 chi2，得到每类标签最相关的一对词和两对词的可视化（如图 7）

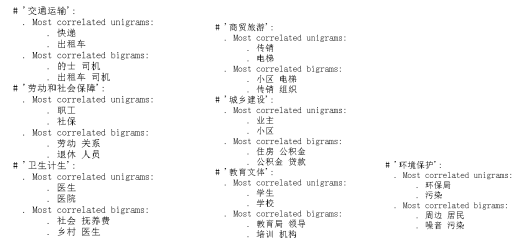

图 7　最相关的一对词和两对词

图 8　可视化

3.2 关键词匹配

3.2.1 TF-IDF

TF-IDF 用以评估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。其主要思想是如果某个词或短语在一篇文章中的出现频率高，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。字词的重

要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

对于我们所处理的留言中某个词语的重要程度，可以标记为词频 (TF) 和逆向文件词频 (IDF) 的乘积，即

$$TFIDF = TF * IDF \tag{1}$$

这里我们采用 $sklearn.feature-extraction.text.TfidfVectorizer$ 方法来抽取文本的 $TF-IDF$ 的特征值。随后使用了参数 $ngram-range$=$(1,2)$, 这表示我们除了抽取留言中的每个词语外, 还要抽取每个词相邻的词并组成一个“词语对”, 如: 词 1, 词 2, 词 3, 词 4,(词 1, 词 2),(词 2, 词 3),(词 3, 词 4)。这样就扩展了我们特征集的数量, 提高我们分类文本的准确度。为了便于比较，设定参数 $nrom$ 使得特征值控制在 $(1,2)$ 的范围内。

基于上述公式可以计算出留言中各个词语的 $TD-IDF$ 特征值，再通过卡方检验方法来检验数据的拟合度和关联度，从而可以找出每个分类留言中关联度最大的两个词语和两个词语对。

3.3 分类器的选择

通过 $sklearn$ 包调入 $train-test-split$,$CountVectorizer$,$TfidfTransformer$ 和朴素贝叶斯分类器, 通过 $train-test-split$ 将按我们设置 $1:4$ 的测试数据和训练数据对数据进行分割, 利用 $CountVectorizer$ 文本特征提取方法, 对训练文本的词频矩阵做进一步的筛选特征, 保留具有代表性的特征 (可提高计算效率), 对词频向量 $fit-transform$ 匹配数据, 找到转换数据的规则, 然后根据找到的规则转换数据, 再使用 $TfidfTransformer$ 里的 $fit-transform$, 将经过转换规则的词频矩阵转为 $TF-IDF$ 向量, 最后将转换好的 $X-train-tfidf$ 和 $y-train$ 带入分类器, 匹配数据并训练可得初次分类目的。

3.3.1 朴素贝叶斯分类器

朴素贝叶斯分类器最适合用于基于词频的高维数据分类器，最典型的应用如垃圾邮件分类器等。其中心思想是对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，比较各个类别出现的概率大小，如果该类别的出现概率最大就认为此待分类项属于哪个类别。待分类的数据对象的所属类别可记为:

$$y_k = argmax(P(y_k|x)), y_k \in y \tag{2}$$

所得 y_k 即为 x 所属类别。上式表示，已知待分类数据对象的情况下，分别计算 x 属于 y_1、y_2、⋯、y_k 的概率，选取其中概率的最大值，此时所对应的 y_k，即为 x 所属类别。我们将留言转换为词频向量，随后将词频向量转换为 $TF-IDF$ 向量，最后利用 $TF-IDF$ 向量对朴素贝叶斯分类器进行训练。当分类器训练完成后，我们定义了一个 $Mypredict$ 函数对一些留言分类进行预测。

3.4 模型的选择

我们同时选择上述的多个模型，代入之前所得出的 $TF-IDF$ 的 $(features)$ 特征向量和 $label-id$ 到这多个模型中，随后经过训练，使用箱线图可视化训练的准确度，最后，我们选择准确度最高的做为分类模型。

随后，我们训练线性支持向量机模型,(这里我们选择的测试数据与训练数据比例是 1 : 2), 通过上述所选择的模型，我们可以得到 $y-pred$ 和测试集的 $y-test$, 生成混淆矩阵。随后为了方便分析，我们将混淆矩阵进行数据可视化，得到的便是每类标签分类成功与分类错误的数据表。

最后通过 $sklearn.metrics$ 调用 $classification-report$，利用 $F1-Score$ 评价模型，计算的最后各个标签分类的平均准确值为 92%。

3.4.1 线性支持向量机模型

线性支持向量机模型是一类按监督学习方式对数据进行二元分类的广义线性分类器，其决策界是对学习样本求解的最大边距超平面。在分类问题中给定输入数据和学习目标：x=x_1,x_2...,x_n,y=y_1,y_2,⋯,y_n

其中输入数据的每个样本都包含多个特征并由此构成特征空间。其中学习目标为二元变量 $y \in$ (-1,1) 分别表示为负类和正类。若输入数据所在的特征空间存在作为决策边界的超平面将学习目标按正类和负类分开, 并使两个平行超平面的距离 $d \geq 1$。

其中决策边界可以记为:

$$\omega * x_i + b = 0 \tag{3}$$

样本到决策边界的距离可以记为:

$$y_i(\omega * x_i + b) \geq 0 \tag{4}$$

参数 ω,b 分别为超平面的法向量和截距。满足该条件的决策边界实际上构造了 2 个平行的超平面作为间隔边界以判别样本的分类:(其中 [1] 为正类、[2] 为负类)

$$\begin{cases} [1] : y_i(\omega * x_i + b) \geq 1 \\ [2] : y_i(\omega * x_i + b) \leq -1 \end{cases} \tag{5}$$

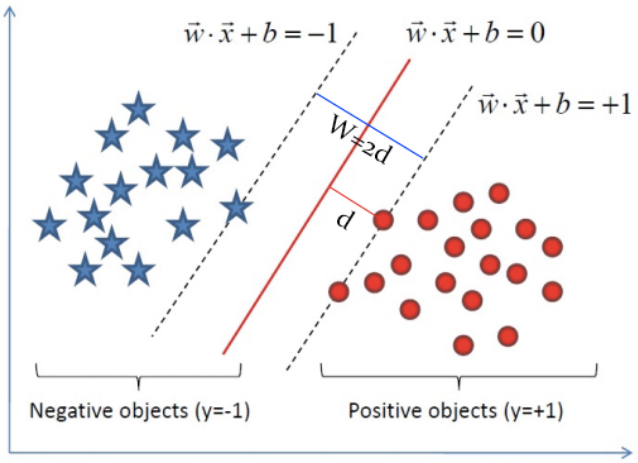

图 9　线性支持向量机示意图

3.4.2 模型训练

基于上述线性支持向量机模型的思想，我们输入留言数据和分类标签进行模型训练，并对模型训练结果的准确率进行评估。结果显示该线性支持向量机模型准确率为86.15%。

然后我们针对该模型作出混淆矩阵，并将其进行可视化后显示预测标签和实际标签之间的差异。

图 10　混淆矩阵预测结果

混淆矩阵的主对角线表示预测正确的数量，除对角线外其余都是预测错误的数量。从上面的混淆矩阵可以看出“城乡建设”的预测错误数量较多。

3.5 模型检验

3.5.1 F-Score

对于多分类模型，在训练数据不平衡的时候，模型的准确率既不能反映出每一个分类的准确性，也不能反映出模型的实际预测精度。因此，我们借助 F-Score 对模型进行评价，从而帮助解决上述问题。

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i + R_i} \tag{6}$$

其中 P_i 为第 i 类的查准率，R_i 为第 i 类的查全率。利用此公式我们可以计算出各类的 $F1 - Score$ 分数。

accuracy 0.9157894736842105

	precision	recall	f1-score	support
城乡建设	0.85	0.96	0.90	663
环境保护	0.96	0.94	0.95	310
交通运输	0.97	0.76	0.85	202
教育文体	0.95	0.94	0.95	525
劳动和社会保障	0.92	0.95	0.93	650
商贸旅游	0.91	0.85	0.88	401
卫生计生	0.96	0.88	0.92	289
accuracy			0.92	3040
macro avg	0.93	0.90	0.91	3040
weighted avg	0.92	0.92	0.92	3040

图 11　F1 分数汇总

从上图的 $F1$ 分数看，模型整体 $F1 - Score$ 分数为 0.92，环境保护类和教育文体类的 $F1$ 分数最高，而交通运输类的 $F1$ 分数最低。其原因可能是该分类的训练数据较少，使得模型学习不充分，导致预测失误较多。但从总体来看，模型的准确率较高且各分类 $F1$ 分数都较高，故得出该模型可用的结论。

四、 任务二

对相似问题进行归类，首先通过留言主题进行分析，将留言主题的文本数据做好分词去停用等预处理后，对其数据进行计算得到词频矩阵，通过 $kmeans$ 聚类的方法对其矩阵数据按照附件一的三级标签个数进行聚类，将聚类后每一簇留言主题对应的

留言详情的文本数据合并在一起，对文本数据进行提取摘要；将每一留言详情与全部的留言详情做文本相似度分析，相似度越高证明该问题所被提及的次数越多，得到每一个留言详情的相似度，便可与反对数和赞成数进行设计热点指数的计算，并对结果进行排列.

五、 任务三

为了从答复的相关性、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，我们选择从留言时间与答复时间的平均所需时间、整体的留言详情和答复意见的文本相似度、留言详情的关键词与答复意见的关键词之间的文本相似度三个方面进行分析。

5.1 对留言时间与答复时间进行分析

首先，我们处理数据中的日期文件，读取附件 4 的数据，将留言时间, 答复时间建立为一个 dataframe，为了更好符号化标注，所以将其文本列名改为对应英文，查看前 5 行检查结果。

	message_time	reply_time
0	2019/4/25 9:32:09	2019/5/10 14:56:53
1	2019/4/24 16:03:40	2019/5/9 9:49:10
2	2019/4/24 15:40:04	2019/5/9 9:49:14
3	2019/4/24 15:07:30	2019/5/9 9:49:42
4	2019/4/23 17:03:19	2019/5/9 9:51:30

图 12　前 5 行检查结果

随后，我们通过 $pandas$ 导入 $to-datetime$ 函数，对其进行时间差的计算。(图 13)

计算 $distance-time$ 的平均值后，将 $distance-time$ 的数据导出至时间差.xlsx 中，在 Excel 中对 $distance-time$ 的数据进行取整处理为 d$istancetime$，通过标准化时间，使用箱线图可视化时间差的大概分布情况.(图 14)

由下图可知，所需时间大部分集中在箱线图上半部分，而且箱线图也出现也不同程度的离群值，也说明了答复所需时间的不确定性，和答复所需时间总体趋势偏长，不能够及时给群众最好的答复，所以答复所需时间的缩短是提高答复意见的首要问题。

	message_time	reply_time	distance_time
0	2019/4/25 9:32:09	2019/5/10 14:56:53	15 days 05:24:44
1	2019/4/24 16:03:40	2019/5/9 9:49:10	14 days 17:45:30
2	2019/4/24 15:40:04	2019/5/9 9:49:14	14 days 18:09:10
3	2019/4/24 15:07:30	2019/5/9 9:49:42	14 days 18:42:12
4	2019/4/23 17:03:19	2019/5/9 9:51:30	15 days 16:48:11
5	2019-04-08 08:37:20	2019/5/9 10:02:08	31 days 01:24:48
6	2019/3/29 11:53:23	2019/5/9 10:18:58	40 days 22:25:35
7	2018/12/31 22:21:59	2019/1/29 10:53:00	28 days 12:31:01
8	2018/12/31 9:55:00	2019/1/16 15:29:43	16 days 05:34:43
9	2018/12/31 9:45:59	2019/1/16 15:31:05	16 days 05:45:06

```
df['distance_time'] = pd.DataFrame(pd.to_datetime(df['reply_time']) - pd.to_datetime(df['message_time']))
```

图 13　pandas 代码以及运行结果

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid',{'font.sans-serif':['simhei','Arial']})
plt.style.use('ggplot')
fig = plt.figure(figsize = (10,16))
data_time = (data_time - np.mean(data_time,axis=0)) / np.std(data_time,axis=0)#标准化时间，以避免箱线图过于扁平
sns.boxplot(data = data_time ,y = 'distancetime')
plt.title('答复时间差箱形图')
plt.ylabel('distancetime')
plt.savefig(r'C:/Users/钟城manman/Desktop/TeddyCup/demo/时间差直方图.png', dpi=100)
```

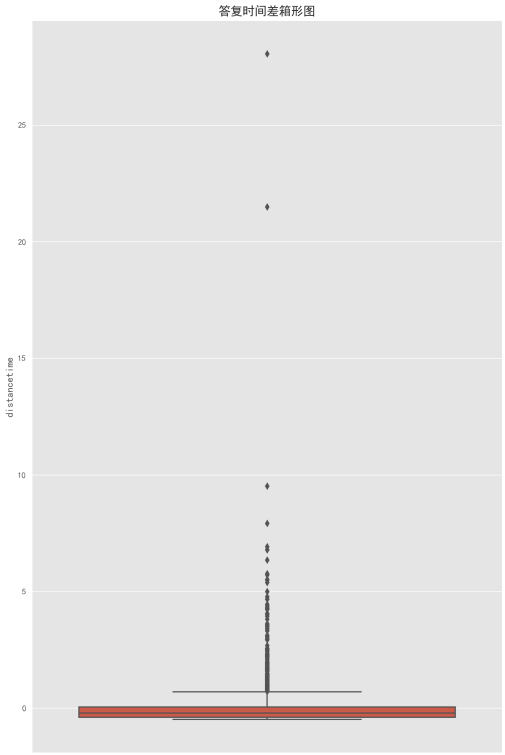

图 14　箱线图可视化

5.2 对整体的留言详情和答复意见的文本相似度进行分析

要了解答复意见对于对于留言详情的效果和完整性，必须要从整体的留言和整体的答复进行把握，从整体上分析答复中对于留言中的词语是否涉及到、一些文本的表述方面是否符合解释留言的内容，所以我们通过计算整体的留言详情和整的答复意见的文本相似度来对其完整性和可解释性进行分析。

5.2.1 关键词提取 -$TextRank$ 算法

$TextRank$ 算法是一种用于文本的基于图的排序算法。$TextRank$ 一般模型可以表示为一个有向有权图 G =(V, E), 由点集合 V 和边集合 E 组成,E 是 V*V 的子集。图中任两点 V_i,V_j 之间边的权重为 w_{ji} , 对于一个给定的点 V_i,In(V_i) 为指向该点的点集合, Out(V_i) 为点 V_i 指向的点集合。点 V_i 的得分定义如下:

$$WS(V_i) = (1-d) + d * \sum_{V_i \in (V_i)} \frac{w_{ji}}{\sum_{V_k \in OUT(V_j)} w_{jk}} WS(V_j) \tag{7}$$

其中,d 为阻尼系数, 取值范围为 0 到 1, 代表从图中某一特定点指向其他任意点的概率, 一般取值为 0.85。使用 $TextRank$ 算法计算图中各点的得分时, 需要给图中的点指定任意的初值, 并递归计算直到收敛, 即图中任意一点的误差率小于给定的极限值时就可以达到收敛, 一般该极限值取 0.0001。

基于 TextRank 算法提取关键词，我们把给定文本 T 按照完整句子进行分割，即 T=[S_1,S_2,$\cdots S_m$]。对于每个句子 $S_i \in T$, 进行分词和词性标注处理，并过滤掉停用词，只保留指定词性的单词，如名词、动词、形容词，即 S_i=[$t_{i,1}$,$t_{i,2}$,$\cdots t_{i,n}$], 其中 $t_{i,j}$ 保留后的候选关键词。根据保留后的候选关键词构建候选关键词图 G=(V, E)，其中 V 为节点集，然后采用共现关系构造任两点之间的边，两个节点之间存在边仅当它们对应的词汇在长度为 K 的窗口中共现 K 表示窗口大小，即最多共现 K 个单词。再根据上面公式，迭代传播各节点的权重，直至收敛。对节点权重进行倒序排序，从而得到最重要的 T 个单词，作为候选关键词。将得到的 T 个候选关键词在原始文本中进行标记，若形成相邻词组，则组合成多词关键词。最后，我们分别把留言详情的关键词和答复意见的关键词写入文本文件中用作下一步处理。

5.2.2 相似度计算 -余弦相似度算法

余弦相似度算法是通过计算两个向量的夹角余弦值来评估他们的相似度。给定两个属性向量，A 和 B，其余弦相似性 θ 由点积和向量长度给出，则余弦相似度 θ 可以表示为：

$$cos(\theta) = \frac{\sum_{i=1}^{n}(A_i * B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} * \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{8}$$

其中 A、B 分别代表为向量 A，B 的各分量，即两个文本的词频向量。对于两个文本的相似性是用两个文本的词频向量的夹角的余弦值来度量它们之间的相似性。相似性的范围 $[-1, 1]$, 若余弦值越接近 -1，则表示它们之间的相似度越低。若余弦值越接近 1，则表示它们之间的相似度越高。

基于余弦相似度算法原理，我们首先对文本作分词处理并分别储存在不同列表里。再对列表进行编码并计算词频，从而得到两个文本的词频向量，最后根据上述式子，进行余弦相似性计算。

5.2.3 分析过程及结论

首先对于留言详情和答复建议的文本预处理，将其导入数据，并建立留言详情和答复建议的 $dataframe$，在对其文本数据进行分词处理，将处理好的文本导出为留言详情.txt 和答复意见.txt, 用做后面的文档之间的文本相似性.(图 15)

```
df['留言'].to_csv('C:/Users/钟城manman/Desktop/TeddyCup/demo/data/留言详情.txt', sep='\n',encoding='utf-8',index=False, header=Nor
```

```
df['答复'].to_csv('C:/Users/钟城manman/Desktop/TeddyCup/demo/data/答复意见.txt', sep='\n',encoding='utf-8',index=False, header=Nor
```

	留言详情	答复意见	留言	答复
0	\n\r\r\r\n\r\r\r\n2019年4月以来，位于A市A2区桂花坪街道...	现将网友在平台《问政西地省》栏目向胡华衡书记留言反映"A2区景馨花苑物业管理有问题"的调查核...	2019年4月以来位于A市A2区桂花坪街道的A2区公安分局宿舍区景馨华苑出现了一番乱象(该小区...	现将网友在平台问政西地省栏目向胡华衡书记留言反映A2区景馨花苑物业管理有问题的调查核实情况向...
1	\n\r\r\r\n\r\r\r\n潇楚南路从2018年开始修，到现在都快一年了...	网友"A00023583"：您好！针对您反映A3区潇楚南路洋湖段怎么还没修好的问题,A3区洋...	潇楚南路从2018年开始修到现在都快一年了路挖得稀烂用围栏围起一直不怎么动工有时候今天来台挖...	网友A00023583您好针对您反映A3区潇楚南路洋湖段怎么还没修好的问题A3区洋湖街道高度...
2	\n\r\r\r\n\r\r\r\n地处省会A市民营幼儿园众多，小孩是祖国的未来...	市民同志：你好！您反映的"请加快提高民营幼儿园教师的待遇"的来信已收悉。现回复如下：为了改善...	地处省会A市民营幼儿园众多小孩是祖国的未来但民营幼儿园教师一直都是最负荷工作且收入又是所有行...	市民同志你好您反映的请加快提高民营幼儿园教师的待遇的来信已收悉现回复如下为了改善和提高民办幼...
3	\n\r\r\r\n\r\r\r\n尊敬的书记：您好！我研究生毕业后根据人才新政...	网友"A000110735"：您好！您在平台《问政西地省》上的留言已收悉，市住建局及时将您反...	尊敬的书记您好我研究生毕业后根据人才新政落户A市想买套公寓请问购买公寓能否享受研究生3万元的...	网友A000110735您好您在平台问政西地省上的留言已收悉市住建局及时将您反映的问题交由市...
4	\n\r\r\r\n\r\r\r\n建议将"臼竹坡路口"更名为"马坡岭小学"，原...	网友"A0009233"，您好，您的留言已收悉，现将具体内容答复如下：关于来信人建议"臼竹坡...	建议将臼竹坡路口更名为马坡岭小学原马坡岭小学取消保留马坡岭	网友A0009233您好您的留言已收悉现将具体内容答复如下关于来信人建议臼竹坡路口更名为马坡...

图 15　运行结果以及代码

随后，打开已经保存的两个文本，使用使用 jieba 分词与自定义的停用词表，对两个文档文本再一步进行处理，将处理后的文本数据放入 word-set 中。随后使用字典将 word-set 里的所有次进行编号，构建词袋模型，并根据词袋模型统计词在两篇文档中出现的次数，形成词频向量。(图 16)

通过引入多维空间余弦函数的公式，求得余弦相似度即可得两个文本的整体相似度为 0.752。通过分析我们可以得出，词频向量是同维度的，距离与相似度呈现反比关系。(图 17)

```
import jieba
import math
import re
#读入两个txt文件存入s1,s2字符串中
s1 = open('C:/Users/钟城manman/Desktop/TeddyCup/demo/data/留言详情.txt','r',encoding='utf-8').read()
s2 = open('C:/Users/钟城manman/Desktop/TeddyCup/demo/data/答复意见.txt','r',encoding='utf-8').read()
```

```
#利用jieba分词与停用词表，将词分好并保存到向量中
stopwords=[]
fstop=open('C:/Users/钟城manman/Desktop/TeddyCup/demo/data/stopwords.txt','r',encoding='utf-8-sig')
for eachWord in fstop:
    eachWord = re.sub("\n", "", eachWord)
    stopwords.append(eachWord)
fstop.close()
s1_cut = [i for i in jieba.cut(s1, cut_all=True) if (i not in stopwords) and i!='']
s2_cut = [i for i in jieba.cut(s2, cut_all=True) if (i not in stopwords) and i!='']
word_set = set(s1_cut).union(set(s2_cut))
```

```
#用字典保存两篇文章中出现的所有词并编上号
word_dict = dict()
i = 0
for word in word_set:
    word_dict[word] = i
    i += 1


#根据词袋模型统计词在每篇文档中出现的次数，形成向量
s1_cut_code = [0]*len(word_dict)

for word in s1_cut:
    s1_cut_code[word_dict[word]]+=1

s2_cut_code = [0]*len(word_dict)
for word in s2_cut:
    s2_cut_code[word_dict[word]]+=1

```

图 16　运行代码

```
# 计算余弦相似度
sum = 0
sq1 = 0
sq2 = 0
for i in range(len(s1_cut_code)):
    sum += s1_cut_code[i] * s2_cut_code[i]
    sq1 += pow(s1_cut_code[i], 2)
    sq2 += pow(s2_cut_code[i], 2)

try:
    result = round(float(sum) / (math.sqrt(sq1) * math.sqrt(sq2)), 3)
except ZeroDivisionError:
    result = 0.0
print("\n余弦相似度为：%f"%result)
```

```
余弦相似度为：0.752000
```

图 17　余弦相似度代码以及运行结果

通过计算两个总体的相似度，直观分析可知在整体答复意见里对于部分的留言回复的词语提及，对于其的完整性和更加准确的回复是不够的，从整体的角度上，答复意见中没有很好的结合群众的留言，在其角度提供意见。

5.3 对留言详情的关键词与答复意见的关键词进行分析

首先我们将附件 4.xlsx 转换为附件 4.csv，然后我们开始对留言主题，留言详情及答复意见进行定义函数 $getKeywords-textrank$，进行拼接标题和摘要，去除停用词，调用 $jieba.analyse.textrank$ 对本本的关键词和词性进行筛选，然后建立 $dataframe$ 把进行筛选的词性 ids，主题 $titles$，关键词 key 进行存储并返回（$topk$ 是关键词的设置）

```
def getKeywords_textrank(data,topK):
    idList,titleList,abstractList = data['id'],data['留言主题'],data['留言详情']
    ids, titles, keys = [], [], []
    for index in range(len(idList)):
        text = '%s。%s' % (titleList[index], abstractList[index]) # 拼接标题和摘要
        jieba.analyse.set_stop_words("C:/Users/75989/Desktop/stopwords.txt") # 加载自定义停用词表
        keywords = jieba.analyse.textrank(text, topK=topK, allowPOS=('n','nz','v','vd','vn','l','a','d'))
        word_split = " ".join(keywords)
        print (word_split)
        keys.append(word_split.encode("utf-8"))
        ids.append(idList[index])
        titles.append(titleList[index])

    result = pd.DataFrame({"id": ids, "title": titles, "key": keys}, columns=['id', 'title', 'key'])
    return result
```

图 18　关键词预处理代码

然后，通过读取附件 4.csv，调用我们设计好的函数 $getKeywords-textrank$，对每一留言详情提取 10 个关键词、每一答复建议 10 个关键词，并展示其结果。由于答复与建议的操作是一样的，下面展出其中一种。

小区 投票 物业公司 业主 方式 采用 收费 业委会 制定 水电
有时候 很大 生意 店面 带来 部门 稀烂 老百姓 监管 围栏
工作 教师 压力 没交 国家 提高 民办 加快 水平 增加
公寓 人才 享受 新政 研究生 您好 购房 补贴 购买 毕业
小学 建议 名称 变更 公交站点 取消 保留 更名
战术 干净 最脏 水资源 职能部门 泥泞不堪 吸尘 洒水车 机器 泥巴
电梯 安装 期待 小区 政策 爱护 关心 群众 买房 部门
教育 社区 医疗 希望 居民 在外 打拼 特别 小孩 小孩子
业主 部门 相关 开工 质量 渗水 停工 督促 解决 交付
建设 路段 绿化带 绿化 地方 带来 小区 成本 草地 森林
养殖 用于 温室 农业 用地 大托 租用 大棚 闲置 有限公司
工程 人防 安置 请问 您好 咨询 地下室
小区 渔业 行人 人行天桥 修建 业主 出行 上下班 地下通道 持续
小区 投票 物业公司 业主 方式 采用 收费 业委会 制定 水电
有时候 很大 生意 店面 带来 部门 稀烂 老百姓 监管 围栏

图 19　每个用户的留言详情 10 个关键词

业主 业委会 业主大会 停车 工作 意见 情况 网友 答复 感谢您
施工 道路 排水 渠道 组织 项目 换填 大道 调查 高度重视
待遇 民办 教师 依法 学前教育 保障 提高 保险 教职工 监管
购房 补贴 公寓 管理中心 房屋交易 全日制 收悉 平台 购买 人才
变更 市民 小学 来信 建议 公交站点 留言 不宜 收悉 支持
街道 学士 工作 监督 关心 回复 马路 较差 很差 路段
电梯 小区 住宅 建局 增设 盼望 工作 监督 关心 组织
服务 街道 建设 工作 社区 居民 教育局 小区 社区卫生 方案
单位 建设 质量 项目 有限公司 施工单位 整改 责任 督查 停工
建设 绿化带 小区 标准 顺路 绿化 片区 两厢 权属 规划设计
租金 支付 公司 村民 同意 中心 耕地 大托村 用地 小组
人防 工程 人防办 建设 安置 批复 地下室 审批表 调查核实 理解业主 业委会 业主大会 停车 工作 意见 情况 网友 答复 感谢您
施工 道路 排水 渠道 组织 项目 换填 大道 调查 高度重视
待遇 民办 教师 依法 学前教育 保障 提高 保险 教职工 监管
购房 补贴 公寓 管理中心 房屋交易 全日制 收悉 平台 购买 人才

图 20　每个用户的答复意见 10 个关键词

在保存其结果的时候，我们发现其中有一部分乱码。对此，我们将其直接复制至留言详情关键词.txt 和答复建议关键词.txt，接着读取这两个文本的数据，对这两个文本进行相似度分析 (与上文步骤相同，计算余弦相似度)，对这两个文本计算的相似度为 92.6%

相较于整体的相似度，细节上的关键词回复的准确度和对于一些答复的考量是明显较好的，其关键词的相似度达到 92.6%，但是整体内容的把握上还是需要改进.

```
#读入两个txt文件存入s1,s2字符串中
s1 = open('C:/Users/75989/Desktop/留言详情关键词.txt','r',encoding='utf-8').read()
s2 = open('C:/Users/75989/Desktop/答复意见关键词.txt','r',encoding='utf-8').read()
```

```
# 计算余弦相似度
sum = 0
sq1 = 0
sq2 = 0
for i in range(len(s1_cut_code)):
    sum += s1_cut_code[i] * s2_cut_code[i]
    sq1 += pow(s1_cut_code[i], 2)
    sq2 += pow(s2_cut_code[i], 2)

try:
    result = round(float(sum) / (math.sqrt(sq1) * math.sqrt(sq2)), 3)
except ZeroDivisionError:
    result = 0.0
print("\n余弦相似度为：%f"%result)
```

```
余弦相似度为：0.926000

余弦相似度为：0.926000
```

图 21　运行代码

综上对答复所需时间，整体和细节文本的相似度的比较来看，留言的质量在时间方面和整体内容把握方面不足，但是在细节关键词上面已经达到了很好的效果，提高留言的质量应该从多方面同时促进，在保证答复速度的同时既能满足用户的需要。

六、 模型的优点以及缺点

6.1 优点

- 本题的模型对于文本数据预处理结合和了多方停用词和 jieba 传统分词工具，既能简单理解操作又可以较为高效的对文本数据完成预处理。
- 问题三对于文本关键词的相似度分析，可较为精确分析一一对应的留言和答复之间的关系。
- 题一中所用到的贝叶斯分词器，当数据大量增加时，也可以保持很好的精确度对于文本的分类，已达到分类目的。

6.2 缺点

- 分类模型较为传统单一，对于分类的精确度没有突破性的提高。
- 传统的预处理方法导致模型实现初期需要花费较长时间。
- 分类模型未能很好的运用到更加精确和广泛的分类任务。

七、 总结

在本文中，我们通过对数据进行预处理、建立标签、运用余弦相似度算法、$TextRank$ 算法等方法解决问题，但在任务二的处理时，由于有限的时间，我们对对于文本的分析不够透彻，极有可能误解了 $kmeans$ 聚类的适用范围，在使用词频矩阵数据进行聚类时出现错误等，该方法无法实现，故任务二没有实现解决。

参考文献

[1] 使用余弦相似度算法计算文本相似度,https://www.cnblogs.com/airnew/p/9563703.html,2018-08-31

[2] 使用python和sklearn的中文文本多分类实战开发, https://blog.csdn.net/weixin_42608414/article/details/88046380

[3] 使用python和sklearn的文本多标签分类实战开发, https://blog.csdn.net/weixin_42608414/article/details/88100879

[4] scikit-learn 机器学习常用算法原理及编程实践, 机械工业出版社