所选题目：“智慧政务”中的文本挖掘应用

综合评定成绩：_____

评委评语：

评委签名：

“智慧政务”中的文本挖掘

摘要：近年来，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。通过数据云计算建立起一种基于自然语言处理技术的智慧政务系统，来代替以往的人工工作已是一种热潮。不仅节约人力，工作效率更是直线而上。利用自然语言处理和文本挖掘对群众留言进行分类；挖掘热点话题的存在；针对相关部门对留言的答复意见，从多角度对答复意见的质量给出一套评价方案。

针对第(1)问，本文通过 jieba 提取数据使用聚类算法，首先按照一定的划分体系对留言进行分类。根据给出的数据，建立关于留言内容的一级标签分类模型。

针对第(2)问，通过 jieba 根据以给数据，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，达到热点话题的提取目的。并对于提取出的相应热点问题对应的留言信息进行附件留存。

针对第(3)问，建立模型，针对相关部门对留言的答复意见，通过 BP 神经网络，以所答复问题、以及问题答复意见作为输入因子。，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可行的高质量评价方案。

关键字：聚类算法、jieba、BP 神经网络、话题分类、热点话题提取

Text Mining In "Smart Government"

Abstract: in recent years, the online political platform has gradually become an important channel for the government to understand public opinion, gather people's wisdom and gather people's spirit. It is an upsurge to build an intelligent government system based on natural language processing technology through data cloud computing to replace the previous manual work. Not only saving manpower, but also working efficiency. Using natural language processing and text mining to classify the public comments; mining the existence of hot topics; according to the response of relevant departments to the comments, this paper gives a set of evaluation scheme for the quality of the response from multiple perspectives.

For question (1), this paper uses the clustering algorithm to extract data through Jieba, and first classifies the messages according to a certain classification system. According to the given data, the first level label classification model of message content is established.

For question (2), according to the data given by Jieba, we classify the messages that reflect the problems of a specific place or population in a certain period of time, define a reasonable heat evaluation index, and give the evaluation results to achieve the purpose of hot topic extraction. And the message information corresponding to the extracted hot issues is retained in the attachment.

For question (3), a model is established. For the reply of the relevant departments to the message, BP neural network is used as the input factors of the reply questions and the reply opinions. From the perspective of relevance, integrity, and interpretability of the response, a set of feasible high-quality evaluation scheme is proposed for the quality of the response.

Keywords: clustering algorithm, Jieba, BP neural network, topic classification, hot topic extraction

目录

1.挖掘目标

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，通过数据云计算建立起一种基于自然语言处理技术的智慧政务系统，来高效高质量代替以往的人工工作。

本次挖掘目标，通过网络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，对已得数据进行划分体系，建立关于留言内容的一级标签分类模型。提取出群众所给出意见的热点话题，同时根据相关部门所给出对留言的答复意见，从多角度对答复意见的质量给出一套评价方案。具体挖掘目标如下：

1) 在处理网络问政平台的群众留言时，对留言进行分类，建立关于留言内容的一级标签分类模型。

2) 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，并且合理的定义热度评价指标，给出评价结果，挖掘出热点话题，进行数据附件留存。

3) 根据相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等多角度对答复意见的质量给出一套高效高质量评价方案。

2.分析方法与过程

2.1 总体流程

本文对“智慧政务”中的文本挖掘模型进行深入分析和研究，采用 jieba、聚类智能算法方法来进行群众话题的分类，以及群众热点话题的提取，并采取 BP 神经网络法建立一个高效高质量评价方案的模型，具体建模的步骤如下：

步骤一：针对问题进行过程分析、相关因素分析、确立整体建模思路。

步骤二：根据前面的相关因素分析，从原始数据中集中选取建模数据。

步骤三：采用合理的方法对抽取的建模数据进行预处理。

步骤四：针对题目要求，进行相关数据的一系列分类以及热点话题的提取，建立相关数学模型进行问题求解，并对模型缺陷进行优化改进。

步骤五：针对实验结果进行相关分析。

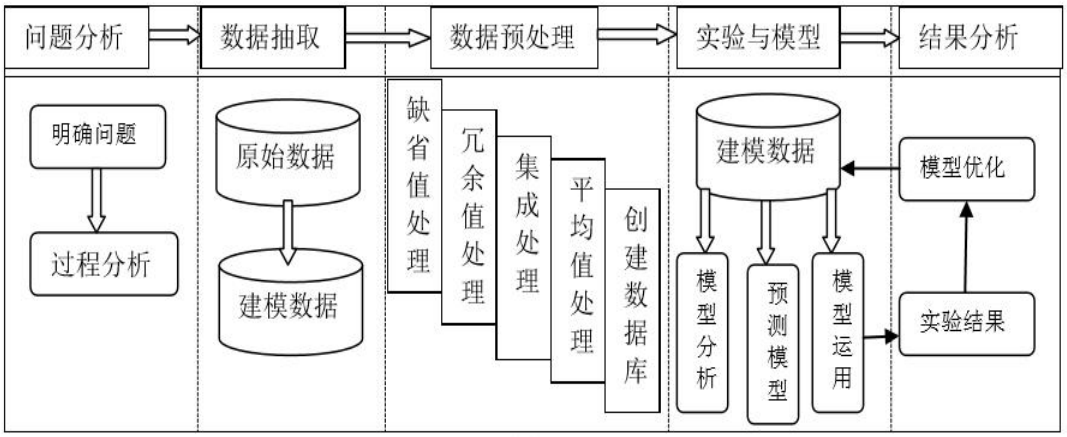

图 1

2.2 具体步骤

2.2.1 问题分析

(1)明确问题

本题属于政府部门对民众留言的分类处理问题，目的在于有效的对群众留言分类，以及对于某一时间段某一地点或某一人群热点话题的提取，并从多角度出发给出一套高效高质量评价方案。实现对于群众留言处理的最大效率化，最高质量化。

(2)过程分析

对“智慧政务”的群众话题高效化，主要分为三个过程：

第一步：多维度的系列划分群众话题；

第二步：从多因素分析，通过 jieba、聚类算法提取出群众的热点话题；

第三步：从评论多角度出发，通过 BP 神经网络法建立起一套高效高质量评价方案。

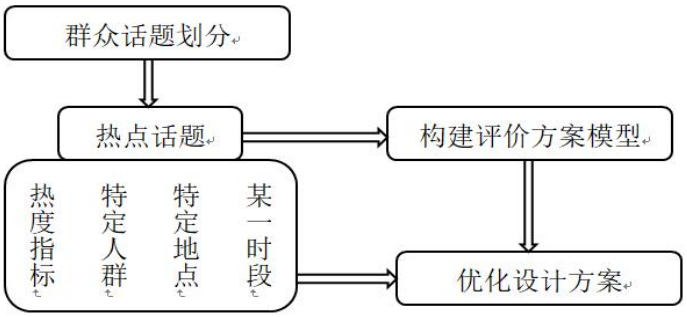

图 2

2.2.2 数据抽取

群众留言数据进行所需因子提取。列如：留言类型，问题地点，留言时间等。用所提取的因子数据，建立最佳的话题模型。

2.2.3 数据预处理

(1)数据清洗、文本去重

在数据的的储存和提取过程中，由于技术和某些客观的原因，造成了相同短信文本内容缺失等情况，因此需要对文本数据进行去重，去重即仅保留重复文本中的一条记录。(去除敏感字符、去除空格、去除“\n”“ \t”)

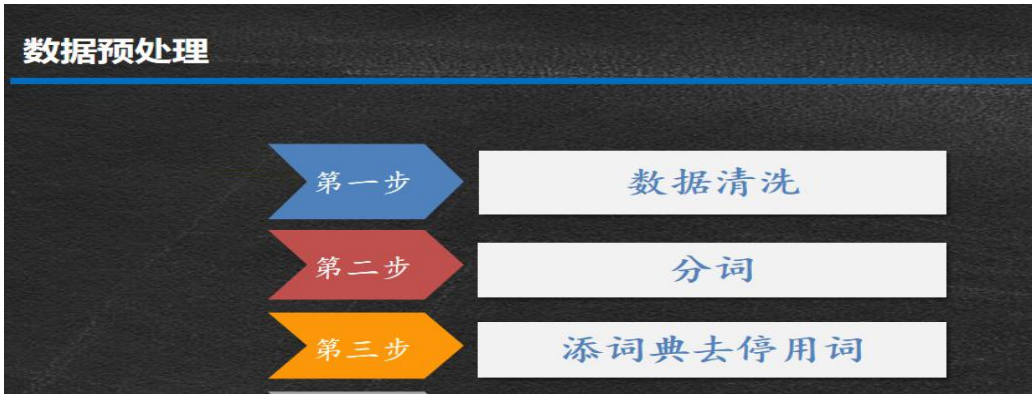

图 3

```
import pandas as pd
import re
import jieba

#def data_process(file=None):
data=pd.read_excel(r'D:\Cti\Cti\附件2.xlsx')
print(data.shape)
def data_process():
    # 数据预处理
    data_message=data[['留言详情','一级标签']]    #提取留言详情和一级标签
    data_message=data.iloc[:,[4,5]]
    print(data_message)

    a=data_message[data_message['一级标签']=='城乡建设']
    b=data_message[data_message['一级标签']=='环境保护']
    c=data_message[data_message['一级标签']=='交通运输']
    d=data_message[data_message['一级标签']=='教育文体']
    e=data_message[data_message['一级标签']=='劳动和社会保障']
    f=data_message[data_message['一级标签']=='商贸旅游']
    g=data_message[data_message['一级标签']=='卫生计生']
    data_new=pd.concat([a,b,c,d,e,f,g],axis=0)

    data_message=data_message.apply(lambda x:re.sub('x','',x))  #去除敏感字符
    data_message=data_message.dropna().apply(lambda x:re.sub('\n','',x))    #去除\n
    data_message=data_message.dropna().apply(lambda x:re.sub('\t','',x))    #去除\t
    data_message=data_message.dropna().apply(lambda x:re.sub(' ','',x))     #去除空格
    data_message=data_message['留言详情'].drop_duplicates()          #去重
    print(data_message)
```

图 4

(2)中文分词

中文分词：指以词作为基本单元，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。

(3)去除停用词

停用词:中文表达中最常用的功能性词语是限定词,如“的”、“一个”、“这”、“那”等。这些词语的使用较大的作用仅仅是协助一些文本的名词描述和概念表达，并没有太多的实际含义。而大多数时候停用词都是非自动生产、人工筛选录入的,因为需要根据不同的主题人为地判断和选择合适的停用词语。

```
#分词
data_messagelcut=data_message.apply(jieba._lcut)    #分词
stopword=pd.read_csv('D:\jd\iddata\stopword.txt',sep='haha')   #导入停用词表
data_afterstop=data_messagelcut.apply(lambda x:[i for i in x if i not in list(stopword.iloc[:,0])])
adata=data_afterstop.apply(lambda x:''.join(x))  #用空格连接词语
```

图 5

(4)部分处理结果

```
0                    [市, 西湖, 建筑, 集团, 占, 道, 施工, 安全隐患]
1                    [市, 在水一方, 大厦, 人为, 烂尾, 多年, 安全隐患]
2           [A3, 区, 杜鹃, 文苑, 小区, 外, 非法, 汽车, 检测站, 开业]
3      [民工, A6, 区明发, 国际, 工地, 受伤, 工, 地方, 拒绝, 支付, 医疗费]
4                          [K8, 县, 丁字街, 商户, 乱, 摆摊]
                              ...
490          [B9, 市, 卫生局, 药品, 监督局, 乱收费, 请, 邓局, 明查]
491                  [K8, 县, 山门, 镇, 医院, 乱收费, 现象]
492                [K7, 县, 医院, 医务人员, 职称, 晋升, 好难]
493                                [医务, 工作者]
494                [G8, 县乡镇, 通, 医务人员, 节假日, 何方]
```

图 6

2.2.4 数据分类

(1)TF-IDF：词频逆文档权重方案，如果某个词或者短语在一篇文章中出现频率高，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。

TF(词频):指的是某一个给定词语在文件中出现的频率。

TF = 该词在文件中出现的次数/在文件中所有字词的次数之和

IDF(逆向文本频率):是一个词语普遍重要性的度量。

IDF = log(总文件数目/包含该词语的文件的数目)。

然后计算 TF 和 IDF 的乘积，值越大说明该词越重要。

(2)朴素贝叶斯

贝叶斯公式：

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

原始的朴素贝叶斯只能处理离散数据，当 $x_1,\cdots,x_n$ 是连续变量时，我们可以使用高斯朴素贝叶斯（Gaussian Naive Bayes）完成分类任务。

当处理连续数据时，一种经典的假设是：与每个类相关的连续变量的分布是基于高斯分布的，故高斯贝叶斯的公式如下：

$$P(x_i=v|y_k)=\frac{1}{\sqrt{2\pi\sigma_{y_k}^2}}\exp(-\frac{(v-\mu_{y_k})}{2\sigma_{y_k}^2})$$

2.2.4 数据热点提取

(1)第二问，依旧要对文本数据附件 3 进行文本数据清洗，文本去重，由于技术和某些客观的原因，造成了相同短信文本内容缺失等情况，因此需要对文本数据进行去重，去重即仅保留重复文本中的一条记录。(去除敏感字符、去除空格、去除“\n”“ \t”)

```
data = pd.read_excel('附件3.xlsx')        #导入附件3.xlsx表
y=data['留言详情'].value_counts() #提取附件中的一级分类的各个类别的数量
#print(y)
plt.rcParams['font.sans-serif']='SimHei'
plt.bar(range(len(y)), y)   #一级分类类别数量条形图
plt.xticks(range(len(y)), y.index)
#plt.show()
data_ao=data['留言主题']#提取数据
data_ao = data_ao.dropna().apply(lambda x: re.sub('？', '', x))  #利用正则表达式将？替换为空
data_ao = data_ao.dropna().apply(lambda x: re.sub('！', '', x))  #利用正则表达式将！替换为空

data_ao=data_ao.drop_duplicates()#去重复数据
data_ao=data_ao.apply(lambda x:re.sub('&[a-zA-Z]+;','',x))#去除表情包
#print(data_ao)
```

(2)中文分词

中文分词：指以词作为基本单元，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。

(3)去除停用词

停用词：中文表达中最常用的功能性词语是限定词，如“的”、“一个”、“这”、“那”等。这些词语的使用较大的作用仅仅是协助一些文本的名词描述和概念表达，并没有太多的实际含义。而大多数时候停用词都是非自动生产、人工筛选录入的，因为需要根据不同的主题人为地判断和选择合适的停用词语。

```
data_cut=data_ao.apply(jieba.cut)#分词
stop_word=pd.read_csv('stopword.txt',sep='haha') #读数据
stop_words = list(stop_word.iloc[:, 0]) + [' ']
data_after_stop = data_cut.apply(lambda x: [i for i in x if i not in stop_words])#去停用词
print(data_after_stop)
```

(4)部分预处理结果

```
10              [魅力, 城, 小区, 临街, 门面, 油烟, 直排, 扰民]
11           [A5, 区, 劳动, 东路, 魅力, 城, 小区, 油烟, 扰民]
12       [市, 经济, 学院, 寒假, 过年, 期间, 组织, 学生, 工厂, 工作]
13       [L, 市, 物业, 服务, 收费, 标准, 应, 居民, 经济, 承受能力]
14                      [市, 江山, 帝景, 新房, 安全隐患]
15        [市, 魅力, 城, 小区, 底层, 商铺, 营业, 凌晨, 噪音, 痛苦]
16          [12123, 申请, 驾驶证, 期满, 换证, 星期, 无人, 受理]
17  [A5, 区, 魅力, 城, 小区, 一楼, 搞, 成, 商业, 门面, 噪音, 扰民]
18                   [分层, 单独, 补交, 超, 面积, 地款]
19             [市, 魅力, 城, 商铺, 排烟, 管道, 区内, 油烟味]
20     [万科, 魅力, 城, 小区, 底层, 门店, 深夜, 经营, 噪音, 扰民]
21                     [市, 实行, 独生子女, 护理, 假]
22          [J4, 县, 供销, 合作社, 岗, 失业, 职工, 追缴, 社保]
23                   [市能, 提高, 医疗, 门诊, 报销, 范畴]
24                   [市, 经济, 学院, 强制, 学生, 实习]
25    [A5, 区, 劳动, 东路, 魅力, 城, 小区, 底层, 餐馆, 油烟, 扰民]
26               [市, 经济, 学院, 强制, 学生, 外出, 实习]
27               [市, 经济, 学院, 组织, 学生, 外出, 打工]
```

(5)提取留言详情中的地点和人群

通过 nltk 库对已经预处理好了的留言详情提取地点人名。

```
# tokenize sentences
sentences = parse_document(text)
tokenized_sentences = [nltk.word_tokenize(sentence) for sentence in sentences]
# tag sentences and use nltk's Named Entity Chunker
tagged_sentences = [nltk.pos_tag(sentence) for sentence in tokenized_sentences]
ne_chunked_sents = [nltk.ne_chunk(tagged) for tagged in tagged_sentences]
# extract all named entities
named_entities = []
for ne_tagged_sentence in ne_chunked_sents:
    for tagged_tree in ne_tagged_sentence:
        # extract only chunks having NE labels
        if hasattr(tagged_tree, 'label'):
            entity_name = ' '.join(c[0] for c in tagged_tree.leaves()) #get NE name
            entity_type = tagged_tree.label() # get NE category
            named_entities.append((entity_name, entity_type))
            # get unique named entities
            named_entities = list(set(named_entities))

# store named entities in a data frame
entity_frame = pd.DataFrame(named_entities, columns=['Entity Name', 'Entity Type'])
# display results
print(entity_frame)
```

(6) 通过 TF-IDF 来定义评价指标

(1)TF-IDF：词频逆文档权重方案，如果某个词或者短语在一篇文章中出现频率高，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。

TF(词频):指的是某一个给定词语在文件中出现的频率。

TF = 该词在文件中出现的次数/在文件中所有字词的次数之和

IDF(逆向文本频率):是一个词语普遍重要性的度量。

IDF = log(总文件数目/包含该词语的文件的数目)。

然后计算 TF 和 IDF 的乘积，值越大说明该词越重要。

(7)保存文本数据

按表 1 的格式将前五名的热点问题保存问文件“热点问题表.xlsx”,按表 2 的格式保存为“热点问题留言明细表”

2.2.5 答复建议评价

（1）第三问，提取数据留言详情，和留言答复，通过数据 dataFrame 化，并进行数组化。

（2）对数据进行分词，并去除停用词。

（3）np.vectorizer 向量化函数，调用函数进行分词和停用词的去除

（4）使用 TF-idf 词袋模型，对特征进行向量化数值映射

（5）通过 from sklearn.metrics.pairwise import consine_similarity，对留言详情和留言答复做相关性矩阵

（6）进行答复建议评价

4.参考文献

[1]jieba 简介

https://blog.csdn.net/qq_37098526/article/details/88877798，2019 年 3 月 28 日

[2] python 实现文本

类,https://blog.csdn.net/laobai1015/article/details/80415080,2018-05-23

[3] python 中文文

类, https://blog.csdn.net/qq_26591517/article/details/79111318,2018-01-19

[4]TF-IDF 权重策

略,https://blog.csdn.net/evan_qb/article/details/78131864,2019 年 9 月 29 日

[5]python 基于 Kmeans 算法实现文本聚类的简单练

习,https://blog.csdn.net/weixin_41276745/article/details/79611259,2018-03-19

[1]神经网络——最易懂最清晰的一篇文

章,https://blog.csdn.net/illikang/article/details/82019945,2018 年 8 月 24 日