“智慧政务”中的文本挖掘应用

摘要

目录

一、问题重述 1
二、模型假设 2
三、数据预处理 2
3.1 数据处理分词并除去停用词 2
四、问题一的分析与求解 4
五、问题二的分析与求解 4
六、问题三的分析与求解 4
七、总结 5
八、附录 5

一、问题重述

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。

智慧政务是移动互联网公共服务平台的客户端，其主要功能是针对地区，搭建政府信息公开新平台，供市民查询政府公告及便民信息，了解办事流程，追踪办件状态，随时随地便享‘智慧政务’全方位覆盖市民日常生活的各个领域，是未来发展新方向。

智慧政务有利于帮助提高政府在行政、服务和管理方面效率，同时实施电子政务可积极推动政府优化办公流程和机构的精简等工作。政府的信息网络覆盖面宽，能够为社会公众提供更快捷、更优质的多元化服务。解决题目问题有利于提

高政府运作效率，降低运作成本，具有现实意义。

问题给出四个附件数据，附件一表示分类标签，包括一级分类、二级分类、三级分类，附件二给用户的留言数据数据，包括留言时间、留言详情等，附件三也是用户留言数据，与数据二不同的是附件三没有分类标签，有点赞数量，附件四是工作人员处理留言详情并回复的数据。

任务一：按照一定的划分体系，并参考附件 1 提供的内容分类三级标签体系对留言进行分类，分类过程中注意附件一标签是层层递进关系。最后使用 F-Score 这个计算公式对分类进行评价。

任务二：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。

任务三：附件 4 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。

二、模型假设

假设信息具有真实性。

三、数据预处理

数据大体上都是不完整，不一致的脏数据，无法直接进行数据挖掘，或挖掘结果不能使用。为了提高数据挖掘的质量产生了数据预处理技术。由于本次项目是用户留言数据，这些数据会存在文本语义带来的词语交叉、数据不平衡、长文本无意义表达太多，需要对本次数据进行以下步骤：数据清洗、分词和词性、除去停用词、文本特征选择、文本表示、文本相似度。

3.1 数据处理并将数据函数封装

针对数据处理，先删除重复值、缺失值，再运用 Python 中的 jieba 库， 对数据进行分词、去除停用词，处理过程中的代码如下：

```
import  pandas  as  pd
import  re
import  jieba
def  data_process(file=&apos;附件 2.xlsx&apos;):
  data=pd.read_excel(file)
  data_dup=data[&apos;留言详情&apos;].drop_duplicates()
  data_cut=data_dup.apply(lambda    x:  jieba.lcut(x))
  stopwords=pd.read_csv(&apos;stoplist.txt&apos;,sep=&apos;hhhhh&apos;,header=None)
  data_stop=data_cut.apply(lambda  x:[i  for  i  in  x  if  i  not  in  stopwords])
```

```
  labels=data.loc[data_stop.index,&apos;留言编号&apos;]
  adata=data_stop.apply(lambda  x:&apos;  &apos;.join(x))

  return  adata,  data_stop,  labels

data_process()
```
得到的就是一个较为干净的文本类数据，函数封装后更方便下一次的调用。

3.2 数据处理并将数据函数封装

```
import  pandas  as  pd
import  re
import  jieba
def  data_process(file=&apos;附件 3.xlsx&apos;):
  data=pd.read_excel(file)
  data_dup=data[&apos;留言详情&apos;].drop_duplicates()
  data_cut=data_dup.apply(lambda    x:  jieba.lcut(x))
  stopwords=pd.read_csv(&apos;stoplist.txt&apos;,sep=&apos;hhhhh&apos;,header=N
one)
  data_stop=data_cut.apply(lambda  x:[i  for  i  in  x  if  i  not  in  stopwor
ds])
  labels=data.loc[data_stop.index,&apos;留言主题&apos;]
  adata=data_stop.apply(lambda  x:&apos;  &apos;.join(x))

  return  adata,  data_stop,  labels

data_process()
```

3.3 数据处理并将数据函数封装

```
import  pandas  as  pd
import  re
import  jieba
def  data_process(file=&apos;附件 4.xlsx&apos;):
  data=pd.read_excel(file)
  data_dup=data[&apos;留言详情&apos;].drop_duplicates()
  data_cut=data_dup.apply(lambda    x:  jieba.lcut(x))
  stopwords=pd.read_csv(&apos;stoplist.txt&apos;,sep=&apos;hhhhh&apos;,header=N
one)
  data_stop=data_cut.apply(lambda  x:[i  for  i  in  x  if  i  not  in  stopwor
ds])
  labels=data.loc[data_stop.index,&apos;答复意见&apos;]
  adata=data_stop.apply(lambda  x:&apos;  &apos;.join(x))
```

```
  return  adata,  data_stop,  labels

data_process()
```

四、问题一的分析与求解

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。对于群众留言分类实际上就是一个文本分类（多分类）问题，针对问题一首先建立一个关于留言内容多分类模型，并建立一个评价模型对分类模型进行评价。存在的难点就是多分类问题带来的难度，需要转为为多个二分类。

求解：1.导入处理好的数据。

2.划分训练集和测试机。

3.将文本数据向量化表达。

4.获取训练样本的 tf_idf 权值向量。

5.获取测试样本的 tf_idf 权值向量。

6.模型训练及测评。通过 F-score 中的精确率：F=2Pi*Ri/Pi+Ri 测试该模型的准度。

五、问题二的分析与求解

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。问题二主要是热点问题挖掘，热点问题是某段一时段内群众集中反映的某一问题成为热点问题,主要难点地点人群识别、相似的计算复杂。问题重点主要分为以下三点：

1、 问题识别

从众多留言中识别出相似的留言

2、 问题归类

把特定地点或人群的数据归并，相似留言归为同一类

3、 热度评价

根据热度评价指标的定义计算方法，对指标排名之后得出对应表

求解：1.导入预处理数据。

2.通过去除停用词将留言主题的长句变为短句，提取关键字，进行词频统计。

3.提取出现词频多的事件，按出现次数设置五个等级，对应时间地点排序，得出热点事件。

4.导出热点事件对应的留言。

六、问题三的分析与求解

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。问题中主要从相关性、完整性可解释性三个角度进行分析，针对这三个角度，再解决过程中会存在将相关性、完整性、可解释性描述量化表现，还需要定制一个指标来评价这套方案。

求解：1.导入预先处理好的数据。

2.分别提取留言主题和答复意见中的关键词。

3.建立文本相似度

七、总结

随着互联网时代的到来，人民反应生活状况更加方便的同时，增加了政府在行政、服务和管理方面等方面的工作难度，而自然语言处理却可以帮助这些工作者们分担工作重量。建立分类模型可以使留言自动分派到相对应的问题，可以直接交给对这一问题做处理的部门去处理，方便人们归类。找出热点问题能更好的体现民情民意，处理当前困扰群众最多的问题。给群众回复留言方便处理一些简单常见的问题，大大减少了工作分量。

八、附录