第八届“泰迪杯”

全国大学生数据挖掘竞赛

论

文

作品名称：基于文本挖掘的群众留言问题分析

基于文本挖掘的群众留言问题分析

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。 而得到的这些信息，也会有助于相关部门提升工作质量。本文将基于数据挖掘技术对留言和热点的数据进行内在信息的挖掘与分析。

在本次数据挖掘过程中，我们首先利用 python、R 语言等操作软件对获得的一些数据进行数据的预处理、分词以及停用词过滤操作，最终将数据转化为可供分析的结构化数据，然后利用分类、聚类、关联分析等技术实现相应的目标。实现了对评论数据的优化，提升可建模度，希望能够解决群众对于政务方面的留言分类并给出答复意见等问题。

关键词：群众留言问题、文本分析、信息提取、文本挖掘、机器自动化

目录

针对问题 1.....4
一、挖掘目标.....4
二、分析方法与过程.....4
针对问题 2.....8
针对问题 3.....14
3.4 基于逻辑规则的机器自动评分.....17
3.5 基于文本相似度的机器自动评分.....18
参考文献.....18

针对问题 1

一、挖掘目标

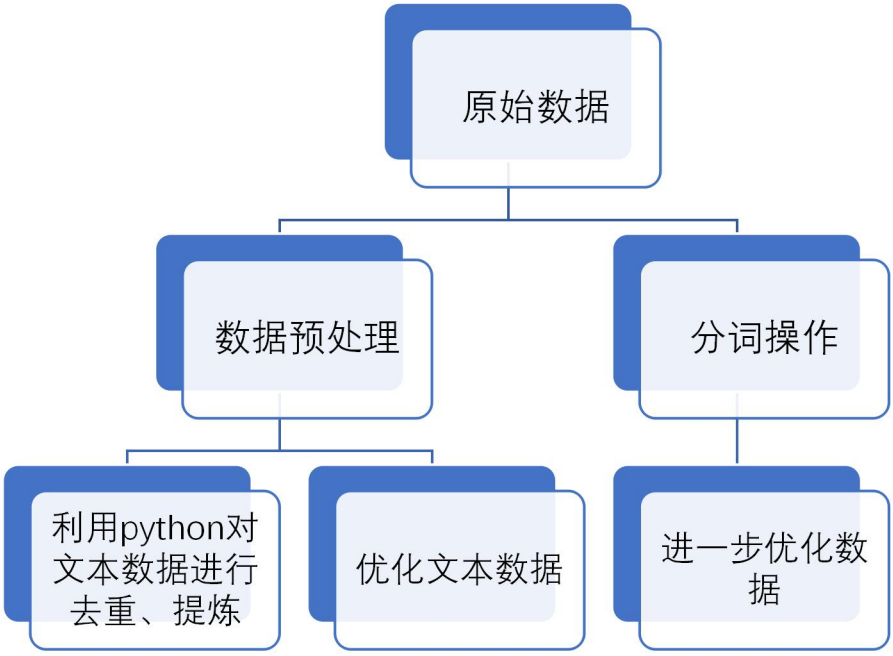
本次建模目标是利用电子政务系统发布的群众留言问题文本信息数据，在对文本数据进行基本的预处理，接着利用 python 的第三方库 jieba 中文分词工具对留言问题进行分词以及停用词过滤后，然后通过建立数学模型，实现对文本留言数据按照三级分类标签体系的分类方法进行分类，期望得到一个更加清晰的数据分类结果，有利于之后相关部门对不同留言问题给出的答复意见。

1.1 对群众留言问题处理难度

目前，由于在处理网络问政平台的群众留言时，大部分电子政务系统对群众的分类还是依靠人工根据经验处理。针对群众留言问题的分析，都是由人工根据留言内容逐一归纳，再对归纳的原因进行专项的统计及分析，进而制定专项解决或对群众有帮助的策略。留言内容繁杂、群众数量大并且人工分类受主观性影响等，致使回复留言效率低，且差错率高等问题。

二、分析方法与过程

2.1 流程图



2.2 数据预处理

2.2.1 群众留言问题的去重、去空

在第 1 小问中，给出附件 2 的数据中，出现了对留言分类问题无用的信息，例如群众留言的留言编号、留言用户、以及留言的时间。由于给出的数据不存在数据缺失

2.3 导入附件 2 群众留言的文本数据并利用 python 的第三方库提取关键词以及绘制词云图

1）、代码如下

```
import jieba
import wordcloud

f = open("C://Users//admin//Desktop//附件 2 文本.txt", "r", encoding="UTF-16LE")

t = f.read()
f.close()
ls = jieba.lcut(t)
txt = " ".join(ls)
w = wordcloud.WordCloud( \
    width = 1500, height = 1200,\
    background_color = "white",
    font_path = "msyh.ttc"  )
w.generate(txt)
w.to_file("liuyan.png")
```

2)、词云图如下

3)、优化程序，生成新的词云图

由于词云中出现了大量诸如“我们，他们，没有，可以，现在，一个，问题，这样，已经”等无用的词，于是进一步优化程序如下：

```
import jieba
import wordcloud

f = open("C://Users//admin//Desktop//附件 2 文本.txt", "r", encoding="UTF-16LE")

exclude={'我们','你们','他们','没有','现在','因而','所以','如果','那么',\
         '如此','只是','但是','就是','这是','那是','而是','而且','虽然',\
         '这些','有些','然后','已经','于是','一种','一个','一样','时候',\
    '没有','什么','这样','这种','这里','不会','一些','这个','仍然','不是',\
    '自己','知道','可以','看到','那儿','问题','一会儿','一点','现在','两个',\
        '三个',\
         }
t = f.read()
f.close()
```

```
ls = jieba.lcut(t)
txt = " ".join(ls)
w = wordcloud.WordCloud( \
    width = 1500, height = 1200,\
    background_color = "white",
    font_path = "msyh.ttc",stopwords=exclude)
w.generate(txt)
w.to_file("liuyan2.png")
```

得到新的词云图：

4)、关键词统计

```
import jieba

txt = open("C://Users//admin//Desktop//附件 2 文本.txt", "r",

encoding="UTF-16LE").read()
words  = jieba.lcut(txt)
counts = {}
for word in words:
    if len(word) == 1:
        continue
    else:
        counts[word] = counts.get(word,0) + 1
items = list(counts.items())
items.sort(key=lambda x:x[1], reverse=True)
for i in range(15):
```

```
    word, count = items[i]
    print ("{0:<10}{1:>5}".format(word, count))
```

得出结果：

```
我们        15568
没有         8334
领导         5533
一个         5467
问题         5136
公司         4333
学校         4200
工作         4027
10         4000
现在         3982
教育         3637
政府         3580
部门         3484
西地省         3435
可以         3428
```

针对问题 2

2.1 导入附件 3 文本数据

（1）词云图

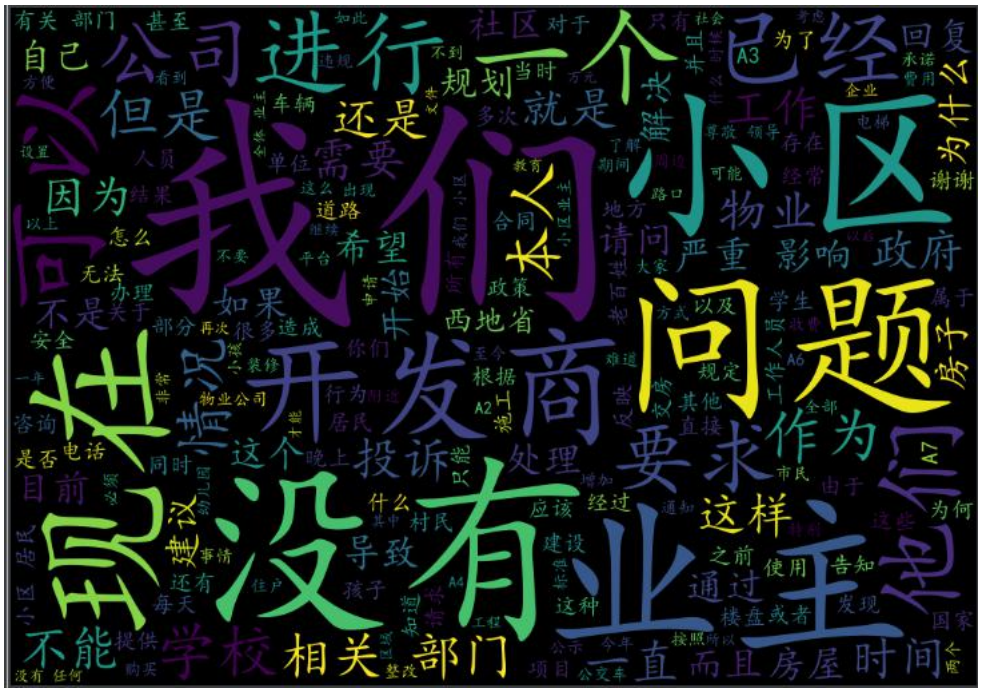
源代码

```
import  jieba
import  wordcloud
from  imageio  import  imread
excludes  =  {  }
f  =  open("附件 3.txt","r",encoding="UTF-16LE")
t  =  f.read()
f.close()
ls  =  jieba.lcut(t)
txt  =  "  ".join(ls)
w  =  wordcloud.WordCloud(\
        width  =  1000,  h)eight  =  700,\
        font_path  =  "simkai.ttf"
```

)

```
w.generate(txt)
w.to_file("grwordcloudm.png")
```

词云图



（2）关键词统计

关键词

```
import  jieba
txt  =  open("附件 3.txt",  "r",  encoding='UTF-16LE').read()
words    =  jieba.lcut(txt)
counts  =  {}
for  word  in  words:
        if  len(word)  ==  1:
                continue
        else:
                counts[word]  =  counts.get(word,0)  +  1
items  =  list(counts.items())
items.sort(key=lambda  x:x[1],  reverse=True)
for  i  in  range(15):
```

```
        word,  count  =  items[i]
        print  ("{0:<10}{1:>5}".format(word,  count))
```

结果显示

```
我们          5738
2019        5313
小区          4423
业主          4234
没有          3434
问题          2845
领导          2134
一个          2054
严重          1877
10          1860
相关          1856
开发商          1845
部门          1832
政府          1732
居民          1711
```

```
> x<-read.csv(file = "C:\\Users\\asus\\Desktop\\附件3.csv")
> x
     留言编号    留言用户
1      188006 A000102948
2      188007  A00074795
3      188031  A00040066
4      188039  A00081379
5      188059  A00028571
6      188073    A909164
7      188074    A909092
8      188119  A00035029
9      188170  A88011323
10     188249  A00084085
11     188251  A00013092
12     188260  A00053484
13     188396  A00047580
14     188399  A00097934
15     188409   A0003274
16     188414  A00096844
17     188416  A00029753
18     188451  A00013004
19     188455  A00035902
20     188467  A00050188

 [ reached 'max' / getOption("max.print") -- omitted 4184 row
s ]
```

导入数据

对数据进行切分，随机分为训练集和测试集

```
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 .
..
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5
 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1
 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..:
 1 1 1 1 1 1 1 1 1 1 ...
```

对整体纯度提升的的最低指标，低于这个指标就进行剪枝。

```
> tc<-rpart.control(minbucket = 5,maxdepth = 10,xval = 5,cp=0
.005)
```

对训练集建立模型

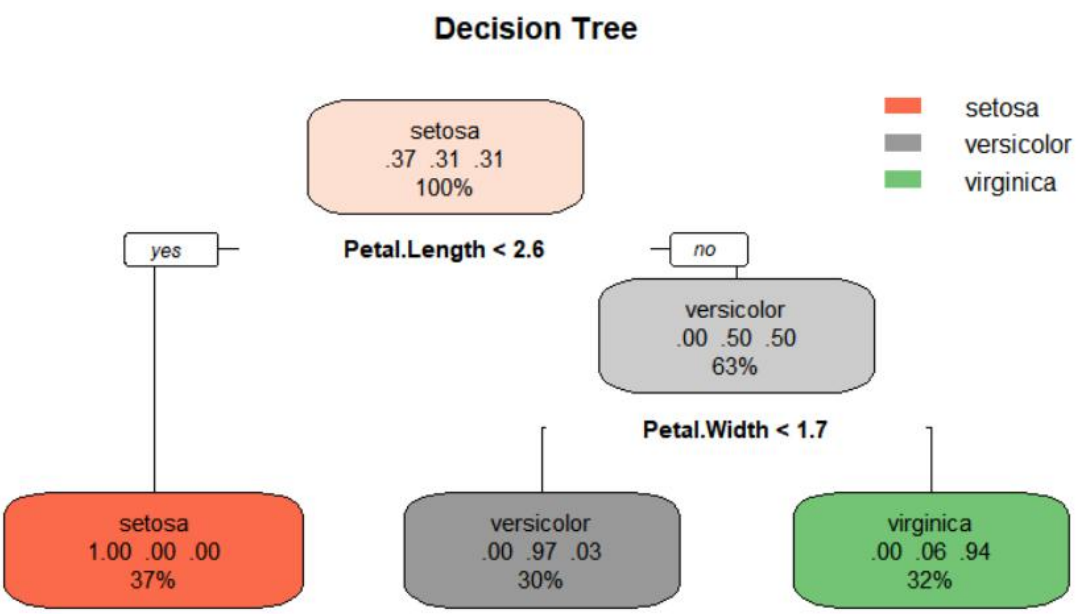
```
> fit<-rpart(Species~.,data=train,control = "tc")
```

用建立好的模型分别对训练集和测试集进行预测，并计算准确率

```
> train.pred<-predict(fit,train[,-5],type = "class")
> table(train$Species==train.pred)['TRUE']/length(train.pred)
    TRUE
0.952381
> test.pred<-predict(fit,test[,-5],type = "class")
> table(test$Species==test.pred)['TRUE']/length(test.pred)
     TRUE
0.9777778
```

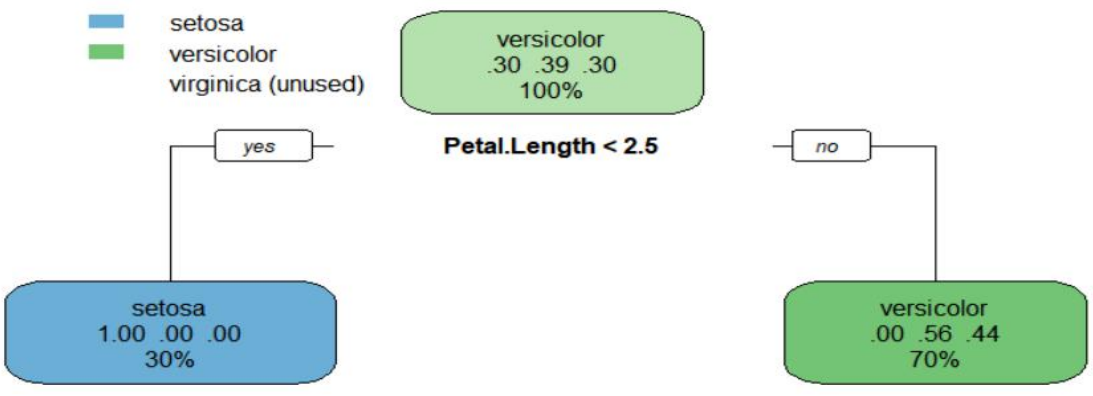
画决策数分类

```
> rpart.plot(fit,main="Decision Tree")
```



进行剪枝

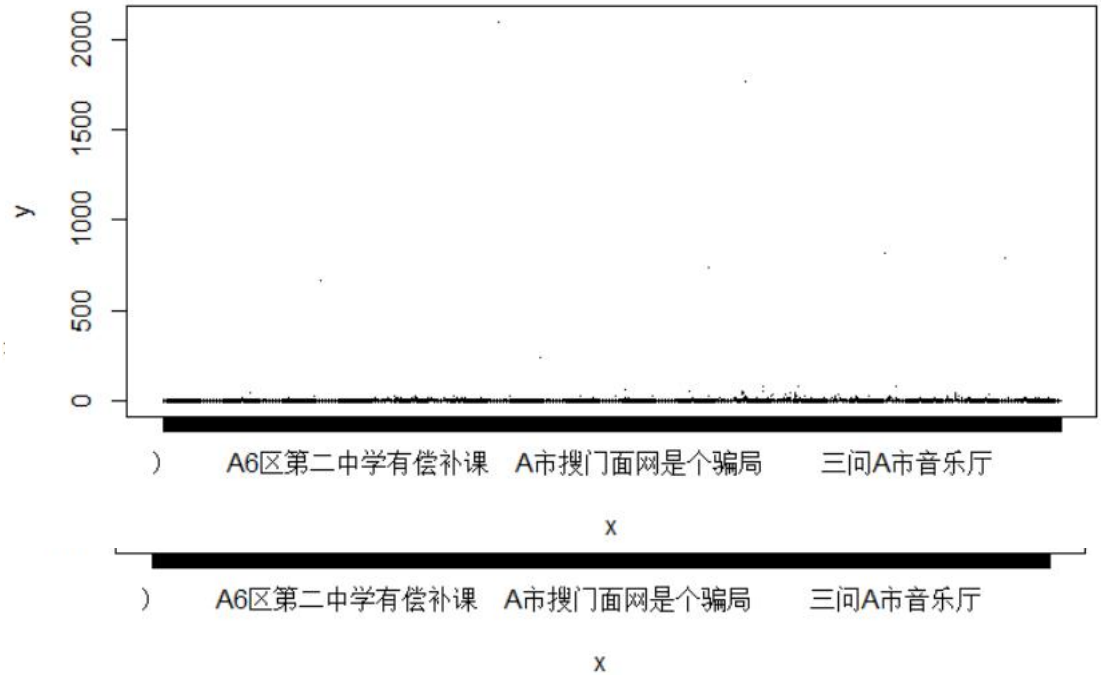
```
> plot(x$留言主题,x$反对数)
```



```
> fit$cptable
         CP nsplit rel error
1 0.500000      0  1.000000
2 0.421875      1  0.500000
3 0.010000      2  0.078125
    xerror       xstd
> prune(fit,0.4219)
n= 105

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 105 64 versicolor (0.3047619 0.3904762 0.3047619)
  2) Petal.Length< 2.45 32  0 setosa (1.0000000 0.0000000 0.0
000000) *
  3) Petal.Length>=2.45 73 32 versicolor (0.0000000 0.5616438
 0.4383562) *
> rpart.plot(prune(fit,0.4219))
> plot(x$留言主题,x$点赞数)
```



```
> summary(x)
     留言编号
 Min.   :188006
 1st Qu.:214143
 Median :239837
 Mean   :239954
 3rd Qu.:264800
 Max.   :360114

      留言用户
 A00031618:  92
 A00051608:  17
 A00074795:  17
 A00083527:  17
 A00032346:  12
 A00054222:  10
 (Other)  :4161
                                                  留言主题
 A7县新国道107距我家仅3米，相关政府部门为何不同意拆迁？:    4
 坚决反对在A7县诺亚山林小区门口设置医院                :    4
 投诉小区附近搅拌站噪音扰民                            :    4
 A市江山帝景新房有严重安全隐患                         :    3
 A市能不能提高医疗门诊报销范畴                         :    3
 A市能否设立南塘城轨公交站？                           :    3
 (Other)                                               :4305
```

针对问题 3

3.1 针对相关部门给群众的回复会存在以下问题

1） 答复时效不及时

例如一件事情的解决办法是较为清晰的，但相关部门迟迟不回复群众，针对问题给群众提供解决办法。导致部门工作效率低以及群众的一丝不满。

2） 答复行文不规范不严谨

答复的内容没有时间落款还可能出现字写错的状况，或是内容与群众反馈的问题不相匹配，答复不具有相关性，还是没能解决群众反馈的问题。

3） 答复内容模糊，糊弄交差

答复的内容会直接出现“网友：您好！留言已收悉”、“您好！您的留言已收悉。您反馈的问题不在外面的管辖范围之内，您可以咨询其他相关部门”、“经查全省人口信息系统，安沙镇万家铺村登记为农村地区。”“您好！您的意见已收悉，我们将迅速向有关部门反馈并作积极处理。”、“您好！您通过平台《问政西地省》的留言收悉，已转有关部门调查答复反馈，谢谢”、“您好!您所反映的问题，已转交”、“已收悉”……这些答复含糊其辞，不具有对问题的可解释性，并不能真正的帮助到广大群众。还有

的甚至没有答复。

习近平强调，各级党委、政府和领导干部要坚持把信访工作作为了解民情、集中民智、维护民利、凝聚民心的一项重要工作，千方百计为群众排忧解难。要切实依法及时就地解决群众合理诉求，注重源头预防，夯实基层基础，加强法治建设，健全化解机制，不断增强工作的前瞻性、系统性、针对性，真正把解决信访问题的过程作为践行党的群众路线、做好群众工作的过程。

这里从答复的相关性、完整性、可解释性等角度给出对群众留言问题给出答复的基本格式。一、首先对群众或网友的留言监督表示感谢；二、留言的问题可能涉及到的法律、法规或法章对此的政策；三、针对群众的留言问题，相关部门经过核查是否能给出一个可解释的解决方案；四、 希望群众继续关心、关注此问题的建设和发展感谢并十分感谢广大群众对我们工作的支持和关心。最后写上时间落款。

因此，在此次数据挖掘比赛中，我们将针对附件 4 相关部门对留言的答复意见，从答复的相关性、完整性、可解释性等角度运用 python 软件、分词操作、提取关键词等步骤对答复意见的质量给出一套评价方案

3.2、文本预处理

文本预处理是自然语言处理中的关键步骤,本研究中的文本预处理包括统一文本格式、去除无关干扰信息、分词等。

为了便于后续分词操作,需要先将两种文档转换成 txt 格式文本,字符编码为 UTF-8 格式。其次,在去除无关干扰信息方面,由于本研究是对答复意见进行评价,故去除文本中无答复意见部门以及与答复意见无关部分,同时为使机器评分结果更准确,去除各个报告中的标题句描述。再次,在对文本进行分词时,使用 jieba 分词工具进行分词。

3.3 导入附件 4 答复意见的文本数据并利用 python 的第三方库提取关键词以及绘制词云图

1）、代码如下

```
import jieba
import wordcloud
f = open("C://Users//Asus//Desktop//附件 4 答复意见.txt", "r", encoding="utf-8")
```

```
t = f.read()
f.close()
ls = jieba.lcut(t)
txt = " ".join(ls)
w = wordcloud.WordCloud( \
    width = 1500, height = 1200,\
    background_color = "white",
    font_path = "msyh.ttc"  )
w.generate(txt)
w.to_file("answer.png")
```

2)、词云图如下

3）、关键词统计

代码：

```
import jieba
txt = open("C://Users//Asus//Desktop//附件 4 答复意见.txt", "r",
encoding='utf-8').read()
words  = jieba.lcut(txt)
counts = {}
for word in words:
    if len(word) == 1:
        continue
    else:
```

```
        counts[word] = counts.get(word,0) + 1
items = list(counts.items())
items.sort(key=lambda x:x[1], reverse=True)
for i in range(15):
    word, count = items[i]
    print ("{0:<10}{1:>5}".format(word, count))
```

结果：

问题	3075
进行	3055
工作	2750
情况	2524
您好	2395
网友	2162
回复	2054
反映	2053
如下	1975
我们	1859
收悉	1840
相关	1798
建设	1660
感谢您	1564
关于	1460

2） 分析

针对答复意见，绘制出词云图以及对关键词排序，由此可知，出现频率最高的是回复如下、留言、收悉、进行、我们工作、反映问题、支持、理解、建设感谢您等词汇。可知这些词汇在评价一个答复意见的完整性具有较强的作用。

3.4 基于逻辑规则的机器自动评分

基于逻辑规则的机器自动评分方法需要用规则将继续教育机构理论评价模型中的三级指标“翻译”为机器评分标准,即将专家制订的三级指标用逻辑规则进行实现。

第一步,基于词向量对指标中涉及的关键词进行词表扩展。利用预训练词嵌

入(Pretrained Word Embeddings)资源获取词向量(Li et al., 2018),其中,每个词可表示为一个 300 维的实数向量,两个向量夹角的余弦值可以标识词语的相似度,利用这个方法计算与目标词相似度最高的 10 个词,即向量空间中最接近的 10 个邻居。

第二步,利用正则表达式将机器评分标准转译为程序语言。正则表达式能够高效地检索和匹配语言特征,故而能够在扩展词表的基础上对评分标准进行较好的算法实现。我们将针对答复意见继续使用机器评分标准逐一用正则表达式编译为程序语言,每条正则表达式的匹配结果均可对应相应分值。

3.5 基于文本相似度的机器自动评分

基于逻辑规则的评分方法虽然采用预训练词向量进行了词表扩展,但由于文本语言表述灵活,规则无法对其表述进行穷尽性描写,实际测试中仍存在规则覆盖不全、召回率低等现象。为了解决这个问题,以提升文本挖掘与评价的鲁棒性,本文针对基于逻辑规则的机器评分与专家评分一致性差的指标,进一步探索文本深层次语义特征,提出了基于文本相似度的评分方法。具体步骤如下:

第一步,构造该指标满分表述。即上述讲到的模板均达到点上

第二步,对满分表述和待评分文本进行分词。

第三步,利用预训练词嵌入资源获取每个词的词向量,对其语义进行表征。

第四步,基于词向量对满分表述和待评分文本进行语义表示。

第五步,计算待评分文本与满分表述向量之间的夹角余弦值,用于表征其相似度(分布在 0～1 之间),由此可预估该文本得分。

基于文本相似度的评分方案能够对任意待评价文本进行语义表示,通过计算其与满分评述的相似度来估计其得分,这种方案不受词表和知识限制,具有较好的灵活性和可扩展性。

文本挖掘技术为答复意见的评价构建了相应的理论评价模型和机器自动化评价方案。但仍需要不断改进。

参考文献

[1]曾雪元,宫伟国,胡云峰,任吉祥.基于决策树算法构建缺血性卒中复发的预测模型[J].吉林中医药,2020,40(04):437-440.

[2]胡治宇.基于 Hadoop 的网络舆情关键字监控体系分析[J].公关世界,2020(06):80-82.

[3]周鑫,刘文松,林峰,杨东,胡竹青,张锦辉,管荣飞.基于文本分析的南瑞集团 186 客服业务能力优化[J].软件,2019,40(12):115-117.

[4]刘威.基于文本挖掘的阅读推广热词分析——以重庆数字图书馆网为例[J].四川图书馆学报,2019(06):15-19.

[5]金吉琼,刘鸿,郑赛晶.基于在线评论文本挖掘技术的电子烟市场消费热点分析[J].烟草科技,2019,52(12):106-114.

[6]郭玉娟,胡韧奋.基于文本挖掘的继续教育机构评价方法新探[J].开放学习研究,2019,24(06):8-14.

[7]王琴,张炯.数据挖掘在移动客户投诉分析中的应用研究[J].湖南邮电职业技术学院学报,2018,17(04):25-27+42.