第八届泰迪杯数据挖掘挑战赛 C 题

作品名称：“智慧政务”中的文本挖掘应用

摘要：本次题目主要以探寻文本挖掘的应用，解决留言平台在处理大量留言时从人工转向使用计算机的问题。提高精度的同时，也提高工作的效率。

关键词：文本挖掘

目录

1.问题背景

网络的普及使我们的生活进入了极其方便的时代，衣食住行的问题可以在家，又或者说在手机上可以得到快速且可靠的解决。如今，各式各类的应用软件的出现，解决了一个又一个生活中的痛点问题，不单单是衣食住行，似乎大部分问题都能在各类应用软件上得到解决。而在此基础上我们付出的代价，仅仅是注册一个一个又一个账号，这对大部分人来说是无关痛痒的甚至不能说是代价。一个个账号，一次次授权使得服务商可以更好的得到用户的使用感受以及找到更好的客户需要从而推出更好的服务。每个用户的使用记录，如购买记录、商品评价甚至浏览的页面记录都是商家策略调整的基础，在这个网络紧密关联生活的时代，用户量是非常大的，而他们所产生的数据量之大是可想而知的。这是个信息爆炸的时代，这是一个大数据的时代。

在这些繁多且杂乱的数据中，评论数据也是构成中很大一部分和很重要的存在，像观影后的评论，外卖的评价，对社会热点事件的议论，凡此种种都是反映人群想法与见解的重要依据。因而许多商家都很关注这些评论的数据，甚至政府也将其作为了解民意、汇聚民智、凝聚民气的重要渠道。随着越来越多的人通过微博、微信、各种热线平台等问政平台评论，政府将其作为重要渠道的愿意并不难理解。然而中国网民的基数是十分庞大的，加上这一方式的便捷性很高，渐渐的，评论数据的累计越来越多。前所未有的数据量，如果依靠传统的人工处理方式来解决，一方面数据量的庞大让人工的工作量加大的同时却并不能很好的去处理，劳民伤财，另一方面人非机器终究有着生理极限制约，在如此大的工作量下，犯错是无法回避的。既然人工无法很好的完成，那么就需要借助智能计算机来完成。

大数据时代下孕育了借助计算机的数据处理技术，即数据挖掘。计算机强大的计算能力且能无差别批量处理的能力，让处理类似这类问题看到了曙光。但是有利就有弊，计算机本质上只能处理数值信息，并不能处理评论类的文字信息。故而需借助自然语言处理技术。本题背景就是探究文本挖掘在文本类数据中的应用，并解决“智慧政务”中的难题。

2.数据挖掘

2.1 简介

上世纪 90 年代起，数据库技术和网络技术的高速发展，海量数据随之产生，各类数据层出不穷。然而带来大量丰富多元的数据的同时，没用系统的同一的数据输入、存储格式，过多的无效信息，必然使得信息产生信息距离，想要提取数据中的有效特征数据变得异常艰难，想要将信息转化为经济收益更是难上加难。约翰·内斯伯特(John Nalsbert)称为的“信息丰富而知识贫乏”窘境。在这种背景下，人们迫切寻求对数据的深入研究，排除无用信息提取有效体征的方法。也就是数据挖掘。

数据挖掘基于统计学，机器学习，人工智能，模式识别，可视化技术等技术，对数据进行处理，实现分类、预测、估值、聚类、相关性分析，一定程度上提取数据的特征，使得原始数据可以转化为可为人使用的有效数据。

2.2 数据挖掘步骤

2.2.1 数据清洗

原始数据是由各类应用软件或者网络平台的后台数据库提供，只是单纯的记录下用户的输入数据，没有针对性并且由于人为的失误，又或者后台的失误，会造成一部分数据的丢失、错乱。同时，根据每个问题的差异，我们所使用到的数据也会有所差别，所以在数据挖掘开始的部分，我们需根据具体的问题来选择相应的数据，同时应该对异常的数据进行处理，防止在之后的操作中出现错误和不准确。

2.2.2 数据分析

问题的差异性和独特性，决定了需要对数据进行分析。找到与问题相关最大，输出影响最大的数据字段。好的数据分析对于之后的模型建立以及挖掘的效果有着重要的作用。原始数据中可能包含成千上百的字段，因此在这一部分中是很耗人精力的，故而需要选择一种具有好的界面展现和强大功能的工具。一般在数据挖掘中常用的编程语言有：MATLAB 、Python、R 语言等，使用的软件就是对应的编译软件。通常，原始数据中有这众多的属性，但并不是每个属性都是我们所需要，又或者没有我们期待的属性，这是时候就需要通过属性规约，属性转换来获得需要的属性数据。

2.2.3 模型建立

对数据的预处理之后，我们会获得解决问题所需要的数据，但仅仅是数据。怎么要提取数据中的特征进而解决问题，得到我们所期望的效果输出，就得通过建立模型来将问题转换为数学上的问题，通过解决数学问题来得到最终的效果输出。常见的数学模型有：微分方程模型、线性规划模型等。好的数学模型建立是得到满意输出结果的重要基础。

2.2.4 评价模型

通过上述的处理过后，我们会得到效果输出，然而这个效果输出是否令人满意，是否真实有效，并不能判断。这时候就需要建立评价模型，通过评价模型获得输出的评价。进而通过评价的结果对模型的建立进行调整，使得模型更加完善，输出的结果更加符合我们的期待。

2.2.5 实施

一切准备就绪后就是将完善后的模型应用在实际的问题上，得到我们想要的数据形式，可以是将数据进行分类，来对特定分类进行推送或者其他操作；也可以是对数据进行预测，得到策略的后续走向，来进行调整。具体的数

据应用可以根据问题的不同而做不同的选择。

3.问题一

3.1 问题介绍

在处理网络问政平台的群众留言平台时，需根据群众留言的内容进行分类以便分派给各个职能部门处理。目前在进行这项工作时，大部分是依靠人工，根据相关的工作经验来分类。具体分类如下图。这就存在着工作量大、工作效率低、出错率高的情况。现在要建立一个分类模型，来代替人工分类，提高分类效率和分类精度。

3.2 问题分析

3.2.1 数据预处理

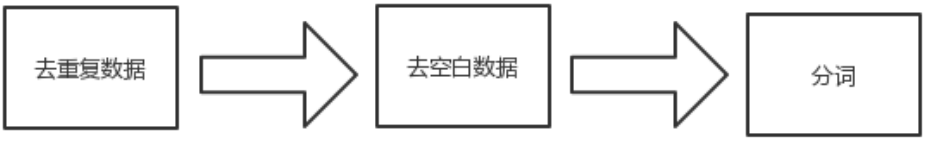
现有附件二中的数据如下。数据中的留言内容、一级分类为重要内容，故在数据预处理中要去除留言内容以及一级分类内容为空的值。数据中如果出现重复的留言信息，在建立分类模型时会影响分类的效果，因此也需要去除。

其次，现有的数据是一条条留言信息与其相应的信息，换言之就是一条条字符型数据和对应的其他类型的数据，主要相关的数据如留言的具体内容大部分是字符型数据。由于计算机并不能直接对字符串直接进行分析，所以需要对字符型数据进行转化，转换成计算机所能直接分析的数值数据处理。

如何将一条字符转换成数值数据呢？这里就需要用到自然语言处理技术。所谓自然言语，就是我们平时说的、写的语言文字。例如“大爷，吃了吗？”，在你我看来这是最简单的打招呼所用到的客套话，简单易懂，但对于计算机来说，这句话无异于天方夜谭。因此要想对自然语言做处理的话就得经过处理，首先就得将完整的句子做分词处理，将一句完整的话拆成一个个单词，按上面的例子举例就是“大爷”+“吃了”+“吗”+“？”。就本题而言我们采用的编程语言是 Python，Python 中与分词相关的库是 jieba 库，可以通过调用这个库来实现分词操作。

经过分词操作后，句子会变成由一个个单词组成的列表。由于自然语言的特

性，一句话里可能会存在一些没有意义却又出现在大量不同的句子中的词语，这些词语可能是谓词如度量单位：天，月，年，吨等，也可能是数字，英文单词，它们在句子中的地位可有可无，一般我们称它们为停用词。因此，在处理时最好是将它们去掉。同时，在分词时，有一些词是特定的，比如专业名词、建筑名，不符合一般的组词规律，可能在分词过程中会被拆散。因此，需要新建一个 txt 文档并写入不希望被拆开的词语，再通过特定的命令，以此保留想要的词语。



3.2.2 数据分析

现有数据的属性有：留言编号、留言用户、留言主题、留言详情、留言时间、一级分类。关于留言的一级标签分类，较为相关的几个属性为留言主题、留言详情、一级标签。由于本题主要解决的是留言内容的分类问题，故而留言用户、留言编号、留言时间相关信息，即并不影响留言的一级标签分类。

3.2.3 模型建立

分类之前先聚类，就是将拥有相同特征的数据聚集在一起，形成几类，并分析出各个类的差异，再根据需要分类的数据进行处理，寻找与之最接近的类别。数据预处理后，数据被分割成一个一个词组还有相应的一级分类标签。部分数据已经有了明确的分类，所以接下来的主要工作就是找出各个分类的特征。

句子中的特征就是句子中重要的词语，所以找出句子中重要的词语也就找出了一条句子的特征，接下来就是如何确定一个词语在句子里的重要程度。TF-IDF 算法，就是一个可以计算词语重要程度的算法。

如何确定一个词条是否是句子的关键词条？一个词条的基本的属性有：在语句中出现的次数（也就是频率）、词条在语句中位置、词语的词性等。tf-idf 中 tf

(Term Frequency) 是词频的意思，指的的是计算一个词语在句子中出现的频率，频率越高的词语，它在语句中的地位越就越高，越能代表语句的中心意思。Idf (inverse document frequency) 是逆文件频率，tf 计算的是一个词语在语句中出现的频率，idf 计算的是一个词语在不同的文档中出现的频率。我们获得的数据是很多个文档，一个词语在多个不同的文档出现的频率越高代表着它在多个文档中被提及，那么它在的价值越就越高，它与其他的词语相比地位越就越高，即权重越高。

TF-IDF 算法：

$$tf\text{-}idf_{\omega} = tf_{\omega} * idf_{\omega}$$

$$tf_{\omega} = \frac{N_{\omega}}{N}$$

$$idf_{\omega} = \log\left(\frac{Y}{Y_{\omega}+1}\right)$$

Tf-idf 算法通过词频与逆文档频率相乘，即强调了在本文档的权重也顾及了在它在其他文档中的权重，比之只考虑在本文档的词频更加全面的刻画了词语的权重。

由于是以频率考量标准，那么对于出现在大量文档中却没有实际意义的词语的去除就有着很高的要求，否则就会出现无意义的词的权重大于真正的关键词。

通过算法得到了每个词语的 Tf-idf 权重后，值得考虑的还有每个词语之间的相对位置。相同的词语集合，可以有着多种排列组合，形成不同的句子，当然也可能表达着不同的意思。所以，对当前的词语集合，可以形成一个词袋向量，就是将所有词语都放到一个向量中，对应一个相同的大小的全零向量，出现的词语就将全零向量中该词语对应的位置的更新为 1，从而形成对应的向量。由于词袋向量中每个的词语的位置是固定的，形成的向量后就能表现每个词语的相对位置。就例如：“你们吃饭了吗”，“你们吃的北京烤鸭吗”。词袋向量以表达成{你们,吃饭,吃的,北京，烤鸭}，第一句对应的就是{1，1，0，0，0}，另一句是{1，0，1，1，1}，当然，这只是简单的举例。有了词袋向量后，只需要将对应的词语的 tf-idf 权重更新到形成的向量中对应的位置后。这样一来，就从多个方面来将语句转化成计算机能识别的数值数据了。

经过上述的处理后，就是对获得的数值数据进行处理、训练。从而使计算机能辨别什么样的数据是属于什么类别的，也就是可以达到分类的目的。要达到计算机能识别数据的类型，也就是要建立一个分类的依据，换言之就是建立一个规则，告诉计算机，什么样的数据表现就是什么类别。在这里我们用的“规则”是多项式贝叶斯分类器。多项式贝叶斯建立在朴素贝叶斯上，要了解多项式贝叶斯就得先了解朴素贝叶斯[1]。

朴素贝叶斯：

$$P(Y \mid X) = \frac{P(Y)P(X \mid Y)}{P(X)}$$

其中，

$$P(X \mid Y) = P(X_1, X_2, \ldots X_n \mid Y) = P(X_1 \mid Y)P(X_2 \mid Y)\ldots P(X_n \mid Y)$$

P(Y|X)叫做后验概率，P(Y)叫做先验概率，P(X|Y)叫做似然概率，P(X)叫做证据。

多项式贝叶斯[2]：

先验概率：

$$P(C = c) = \frac{\text{属于类c的文档数}}{\text{训练集文档总数}}$$

条件概率：

$$P(\omega_i \mid c) = \frac{\text{词}\omega_i\text{在属于类c的所有文档中出现的次数}}{\text{属于类c的所有文档中的词语总数}}$$

预测：

$$\begin{aligned}
&\arg\max_{c \in C} P(c \mid w_1, w_2, \ldots, w_n) \\
&= \arg\max_{c \in C} [P(c)P(w_1, w_2, \ldots, w_n \mid c)] \\
&= \arg\max_{c \in C} [P(c)P(w_1 \mid c)P(w_2 \mid c)\ldots P(w_n \mid c)] \\
&= \arg\max_{c \in C} [\log P(c) + \log P(w_1 \mid c) + \ldots + \log (w_n \mid c)]
\end{aligned}$$

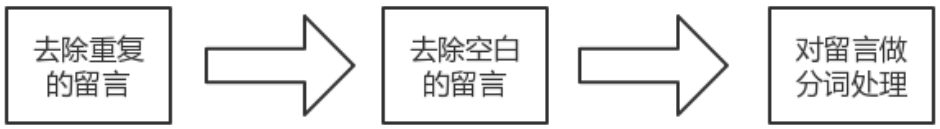
4.问题二

4.1 问题介绍

留言的平台上会获得众多的留言，对于平台的管理者来说希望的是通过平台上的留言来获得用户的问题，从而让相关部门去解决。留言多了，问题也就多了，那么问题的解决优先级是怎么确定呢？也就是想解决什么问题呢？当然是先解决多人提出的问题、影响大的问题、紧急的问题，这些就是所谓的热点问题。问题二就是想通过对留言数据的处理，在茫茫的留言中找出热点的留言，并建立一个热点指标来体现留言的热点程度，更好的确认要先解决什么留言。要求找出热度前五的留言，形成热点问题表和热点问题详情表

4.2 问题分析

4.2.1 数据预处理



4.2.2 数据分析

热点事件也就是在某一时段内，同一事件被多名不同用户提起的事件。能直接表现出来的这一点的就是相似留言的条数，更加直观的是一条留言的反对数和点赞数。当然，仅仅靠这两点是不够的，还有时间的跨度、留言内容对社会的影响，就算是不同的社会身份的人提出的留言也有对留言是否是热点事件有着或多多少的影响。

从获得数据来看，我们能获得以下的资料：留言编号、留言用户、留言主题、

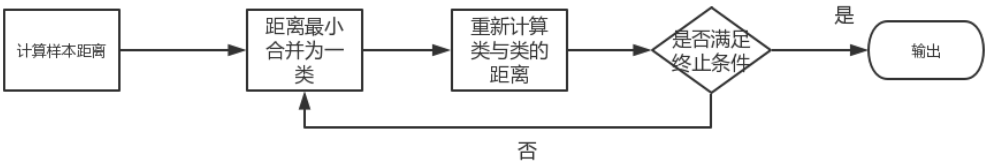
留言详情、留言时间、反对数、点赞数。由于没有留言用户的更多数据，可以抛开不同的留言用户的影响。同时，我们也不知道什么类别的事件对社会或对大部分人群会造成更大的影响，所以，应该获得更多的关注的数据应该是每条留言的反对数、点赞数、相似的留言条数、留言的时间跨度。前三点共同体现留言的关注度，后者体现的是时间的跨度。

4.2.3 模型建立

要解决问题首先聚集相似的留言，只有聚集好相似的留言才能对相同问题留言的点赞数，反对数，留言条数，也就是关注度，以及确定问题留言的时间宽度进行计算，才能计算出不同留言的热度指标。

聚集相似留言，就是聚类。现有的聚类算法有 K-means、DBscan、层次聚类法等，传统的 K-means 聚类要预先设定好要聚类的类数（k 值），对于热点问题的挖掘来说，要预先给出 k 值一件很难得事，虽然有 DBI 指标可以测试不同 k 值的聚类效果，但对于几千条甚至几万条数据来说，一个个试 k 值，不仅计算量大，而且耗时长。主要是无法确定 K 值。DBscan 的缺点也是样本集较大时，聚类收敛时间较长。所以我们将目光转向层次聚类法，虽然它的处理速度同样不快，但它的优点则是不用预先给出聚类的个数，而是通过输入的簇间的距离来自行调整簇的个数。

层次聚类法[3]的原理为：现将每一条留言看做一个个样本，计算每个样本之间的距离，将样本最近的样本聚成一类，再计算类与类之间的距离，距离最近的类合并成一个大类，直到最终聚成一个类或者终结条件达成。流程图如下：



一般鉴定相似度使用的是欧式距离，但对于文本数据，大量数据表明使用余弦距离更好。

余弦相似度：

$$\text{dist}(A, B) = 1 - \cos(A, B) = \frac{\|A\|_2 \|B\|_2 - A * B}{\|A\|_2 \|B\|_2}$$

热度指标建立：在将相似的留言取出后，计算留言的关注度、时间跨度

关注度：

$$关注度 = 赞成数 + 反对数$$

时间跨度：

$$时间跨度 = 最晚时间 - 最早时间$$

热度指标：

$$热度指标 = \frac{关注度}{时间跨度}$$

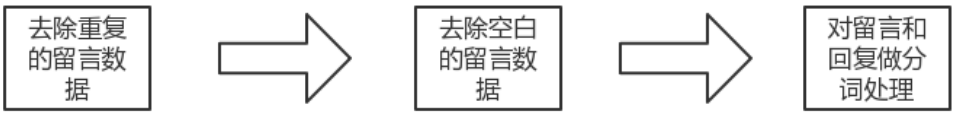
5.问题三

5.1 问题介绍

有留言就有回复，回复的质量与平台的口碑有着密切的关系。问题三要求形成回复的评价指标。给每一条留言的回复评分，从而改进回复的质量。

5.2 问题分析

5.2.1 数据预处理



5.2.2 数据分析

问题四要求对回复信息进行评价，既然要评价就得全面的评价。那么如何全面的评价呢？

我们对于留言的回复最关注的就是这个评价是否是针对我们的留言所进行的回复，换言之就是回复与留言之间的相关性。只有相关性大，回复才有意义。

如果相关性小，也就意味着牛头不对马嘴，决解不了问题的回复只不过是一串字符串，没有任何作用。其次，在确保相关性的前提下答复的有效也是很重要的。如果一条答复只是相关却并没有完整的解决，只解决部分问题，那么这条答复的价值也会随之下降。最后，每个留言的用户都想自己的问题能够在最短的时间得到回复。毕竟随着时间的增长，也会磨掉用户的耐性，使得对留言平台的信服力下降。

综上所述，本题我们打算从评论的相关性、完整性、回复时间长短三个方面去评价。

从附件四的数据来看，我们可以得到以下信息：留言编号、留言用户、留言主题、留言详情、留言时间、答复详情、答复时间。与此相关的数据，显而易见的就是留言主题、留言详情、答复详情、留言时间、答复时间。其中留言主题与留言详情都是表达留言用户所要反映的问题，留言主题是留言详情的大概说明，答复详情则是对问题的全面解答，在留言的相关性的分析中，留言详情所能贡献的比留言主题要大，故主要以留言详情为依据。

5.2.3 模型建立

回复的相关性，可以通过做对留言的详情和对应的答复详情的相似度计算来得到相应的相关性。

回复的完整性，通过对答复的长度来定义。确保相关性的情况下，答复的长度越长，意味着内容的详细、全面。由于无法做到让计算机做到像人一样去识别内容具体的内容，故只好从长度入手。一般认为长度大于 100 则认为该留言完整。完整则将其完整性得分赋值为 1，否则为 0。

回复的时间长短，可以从答复时间和留言时间的差值来获得。一般在一星期以内（7 天）认为快速回复，时间得分为 1，一个月内回复则为及时回复，时间得分为 0.8，三个月则为拖延回复，时间得分为 0.6，其他则为垃圾回复赋值为 0。

$$回复得分 = \frac{时间得分 + 相关得分 + 完整得分}{3}$$

参考文献：

[1] 何伟. 基于朴素贝叶斯的文本分类算法研究[D].南京邮电大学,2018.

[2] 张伦干. 多项式朴素贝叶斯文本分类算法改进研究[D].中国地质大学,2018.

[3] 韩忠明,陈妮,张慧,杨伟杰.一种非对称距离下的层次聚类算法[J].模式识别与人工智能,2014,27(05):410-416.

[4] 叶雪梅. 文本分类 TF-IDF 算法的改进研究[D].合肥工业大学,2019.