第八届“泰迪杯”

数据挖掘挑战赛

作品名称：“智慧政务”中的文本挖掘应用

“智慧政务”中的文本挖掘应用

摘要：随着互联网技术的快速发展，网络问政平台逐步发展成为了解民意、汇聚民智、凝聚民气的重要渠道。因此，为提升政府的管理水平和施政效率，对各类社情民意相关文本数据的文本挖掘应用具有重大意义。

在数据预处理过程中，对从附件中提取出的所需文本使用结巴库来进行中文文本分词、去除停用词。

针对问题 1，对附件 2 中的文本进行数据预处理后，采用 TF-IDF 算法根据权重选取关键词，然后用高斯朴素贝叶斯分类模型，对各文本的关键词按照附件 1 的一级标签来分类。

针对问题 2，首先对附件 3 中的留言详情进行归类，通过制作语料库，用 TF-IDF 计算语料库中每个留言的词频向量的权值，设置阈值并比较分析，选出可归为同类的留言，再定义热度评价指标，通过计算热度指数得到关于附件 3 的热度评价表格。

针对问题 3，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、可解释性等角度对答复意见的质量给出一套评价方案:通过计算每个留言与对应答复的词频向量的余弦相似度，再计算所有相似度的算术平均值，得到答复的相关度；通过 Excel 查找每个具有完整性关键词的答复，计算完整性答复在所有答复中的占比，得到答复的完整度；通过 Excel 查找每个具有可解释性关键词的答复，计算其在所有答复中的占比，得到答复的可解释性。

关键词：TF-IDF 算法、高斯朴素贝叶斯分类算法、文本相似度

“智慧政务”中的文本挖掘应用

1. 背景与挖掘目标

1.1 问题背景

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。

1.2 挖掘目标

为提升政府的管理水平和施政效率，给出收集自互联网公开来源的群众问政留言记录和相关部门对部分群众留言的答复意见，深入分析信息数据，利用数据挖掘的方法解决下面的问题：

问题 1：群众留言分类

参考附件 1 中提供的分类三级标签划分体系，并根据附件 2 所给出的群众问政留言数据，建立关于留言内容的一级标签分类模型。

问题 2：热点问题挖掘

根据附件 3 所给出的群众问政留言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。

问题 3：答复意见评价

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、可解释性等角度对答复意见的质量给出一套评价方案。

2. 问题解析

通过已有的数据和问题的解决方向进行详细的问题分析：

2.1 群众留言分类

问题 1 群众留言分类流程如图 1 所示：

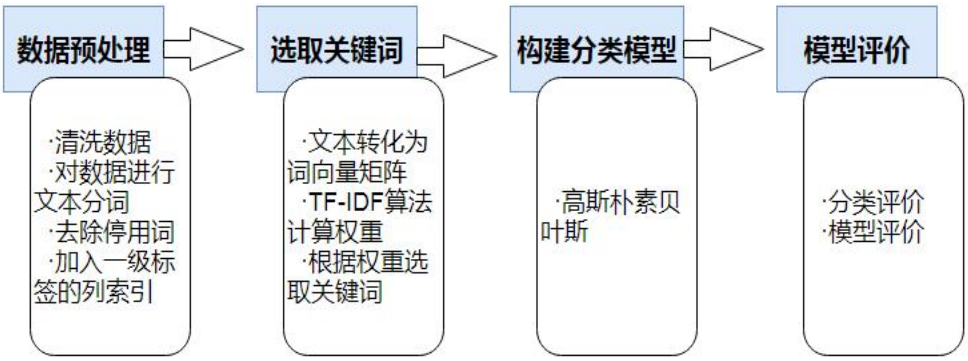

图 1 问题一流程图

数据预处理：

对附件 2 中的数据进行数据预处理，对文本数据进行文本去重，即仅保留重复文本中的一条记录，删除本实验没有用到的数据，接着通过 Python 对文本数据进行分词、去除停用词，加入附件 1 中所有的一级标签作为列索引。

选取关键词：

将预处理后的文本转化为词向量矩阵，使用 TF-IDF 算法，根据算法计算出的权重发现特征词，并选出该文本的主题的关键词。

构建文本分类模型：

使用高斯朴素贝叶斯分类方法，符合高斯分布的一个特征的观测值属于某个类别。

模型评价：

解决问题一后，用 F-Score 对完成分类贴标签的方法进行评价。公式：$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$，其中 P_i 为第 i 类的查准率，R_i 为第 i 类的查全率。

2.2 热点问题挖掘

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。

问题 2 热点问题挖掘的流程如图 2 所示：

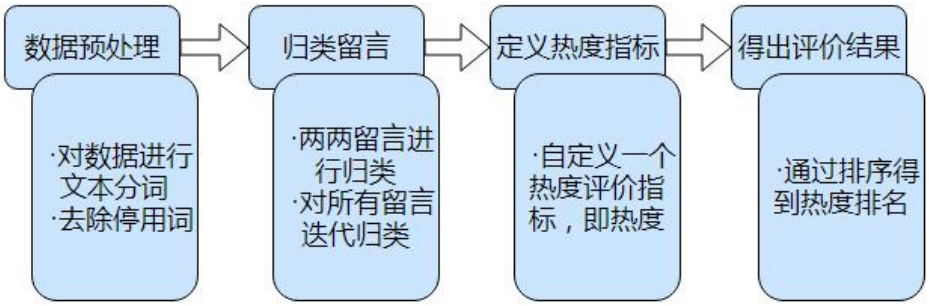

图 2 热点问题挖掘流程图

数据预处理：

通过 Python 对附件 3 中的留言详情进行结巴分词，选择合适且通用的停用词语库去除停用词。需要注意的是，此时的预处理不需要数据去重操作，不利于计算热度指数。

归类留言：

首先选定第一条留言，与其余留言分别两两进行归类（将在 4.3 详细介绍本文归类的方法），分析归类计算结果，当留言 TF-IDF 权值大于一个自定义阈值时，可认为这两条留言是同类的留言。排除选出的所有与第一条同类的留言，再次选定新的第一条留言，与其余留言分别两两归类，重复上一步动作，这样进行两两留言的迭代归类，最终得到分好类的所有留言。

定义热度评价指标：

通过数据的观察与分析，自定义一个热度评价指标为热度，算出热度指数。

热度=同类的留言数+点赞数+反对数，其中热度指的是对同一类问题的关注程度，点赞数可视为留言了相同的内容，反对数表明关注了同一个问题，可包含在热度内。

得出评价结果：

使用 Excel 对热度指数排序，得到相应热点问题对应的留言信息，并保存为“热点问题留言明细表.xls”；得到排名前 5 的热点问题，并保存为文件“热点问题表.xls”。

2.3 答复意见评价

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、可解释性等角度对答复意见的质量给出一套评价方案：

相关性：

将留言文本中的关键词与对应答复中的关键词作相关性比较，计算余弦相似度。通过留言详情与答复意见的相关度来评价关于相关性的答复质量。

针对附件 4 相关部门对留言的答复意见的相关性，通过计算相关度来评价质量，我们定义相关度的计算方法为：相关度=所有余弦值的算术平均值，其中余弦值为留言与对应答复的余弦相似度，这样的相关度可以评价出附件 4 的整体答复的相关性。

完整性：

根据每个答复的词频统计观察文本数据，包含有常用的答复关键词并且为完整的、全面的答复，可认为该答复具有完整性，计算具有完整性的答复在所有答复中的占比，则为附件 4 相关部门对留言的答复意见的完整度。

首先，将所有留言中出现的近义词语替换为同一个词语，保证统计数量的最大化，接着，使用 Excel 中的查找工具，查找出具有完整性的答复的所有结果，计算出其在所有答复中的占比，得到完整度。

可解释性：

在答复中运用法律法规、文献等权威资料，可认为该答复具有可解释性，得到所有具有可解释性的答复，并计算具有可解释性的答复在所有答复中的占比，则得到该相关部门对留言的答复意见的可解释度。

3. 数据预处理

对所用文本的数据预处理流程图如图 3 所示：

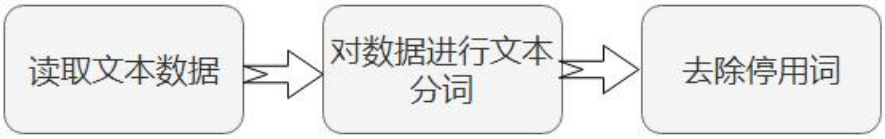

图 3 数据预处理流程图

读取文本数据：

在附件中，读取出解决问题时对应附件中所需的文本数据。

文本分词：

结巴库是 Python 中可以进行中文分词的第三方库，使用结巴分词的方法，可自动对中文文本进行词语的切分，获得单个的词语。

去除停用词：

语言表达中包含很多功能词，与其它词相比，功能词没有什么实际含义。中文表达中最常用的功能性词语是限定词，这些词在文本中描述名词和表达概念，如地点或数量，并没有

太多的实际含义。

在信息检索中，这些功能词的另一个名称是：停用词（stopword），称为停用词是因为在文本处理过程中如果遇到它们，则立即停止处理，将其扔掉，这样减少了索引量，增加了检索效率，并且通常都会提高检索的效果。

因此，根据附件中不同的文本主题，选择合适且通用的停用词语库，去除文本中的停用词。

4. 挖掘建模

4.1 选取关键词

对附件 2 留言进行文本分词后所得到的词语，显然会有许多不重要的词语，难以进行一级标签的分类，所以还需要进一步筛选出每个留言的文本关键词。在此，我们将使用 TF-IDF 算法来提取留言中的关键字，即计算出每个词的 TF-IDF 值，然后按降序排列，取排在最前面的几个词为关键词。

TF-IDF 算法：TF-IDF（词频-逆向文件频率）是一种用于信息检索与文本挖掘的常用加权技术；是一种统计方法，用以评估一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。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。

（1）TF 是词频。词频表示词条（关键字）在文本中出现的频率。这个数字通常会被归一化(一般是词频除以文章总词数)，以防止它偏向长的文件。

公式：$f_{ij}=\frac{n_{i,j}}{\sum_k n_{k,j}}$，即 $TF_\omega=\frac{\text{在某一类中词条}\omega\text{出现的次数}}{\text{该类中所有的词条数目}}$，

其中 n_{ij} 是该词在文件 d_j 中出现的次数，分母则是文件 d_j 中所有词汇出现的次数总和。

（2）IDF 是逆向文件频率。逆向文件频率：某一特定词语的 IDF，可以由总文件数目除以包含该词语的文件的数目，再将得到的商取对数得到。如果包含词条 t 的文档越少，IDF 越大，则说明词条具有很好的类别区分能力。

公式：$idf_i=\log\frac{|D|}{\left|\{j:t_i\in d_j\}\right|}$，其中，$|D|$ 是语料库中的文件总数。$\left|\{j:t_i\in d_j\}\right|$ 表示包含词语 t_i 的文件数目（即 $n_{i,j}\neq 0$ 的文件数目）。如果该词语不在语料库中，就会导致分母为零，因此一般情况下使用 $1+\left|\{j:t_i\in d_j\}\right|$。即 $IDF=\log\left(\frac{\text{语料库的文档总数}}{\text{包含词条}\omega\text{的文档数+1}}\right)$ 为防止分母为 0，故使分母加 1。

（3）TF-IDF 实际上是：TF * IDF 某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的 TF-IDF。因此，TF-IDF 倾向于过滤掉常见的词语，保留重要的词语。

4.2 构建分类模型

在留言文本选定的关键词数据中，选对留言的关键词数据进行切分，90%的数据作为训练集样本，10%的数据作为测试机样本。将训练集样本和测试集样本转换成 tf-idf 权值向量，接着使用高斯朴素贝叶斯模型 GaussianNB 来分类留言并贴标签，训练集用来对训练高斯模型，使模型能够分类文本留言，接着用测试集样本测试该模型是否可以将留言文本分类。

朴素贝叶斯算法：朴素贝叶斯法是基于贝叶斯定理 $P(B|A)=\frac{P(A|B)P(B)}{P(A)}$（$P(A)$非零）与特征条件独立假设的分类方法。

朴素贝叶斯-高斯模型：当特征是连续变量的时候，运用多项式模型会导致很多 $P(\mathrm{x}_i|y_k)=OP(x_i|y_k)=0$（不做平滑的情况下），此时即使做平滑，所得到的条件概率也难以描述真实情况。所以我们处理连续的特征变量时，采用高斯模型。GaussianNB 模型实现了运用于分类的高斯朴素贝叶斯算法。特征的可能性(即概率)假设为高斯分布:

$$P(\mathrm{x}_i|y)=\frac{1}{\sqrt{2\pi\sigma_y^{\,2}}}\exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^{\,2}}\right)$$，参数σ_y和μ_y采用极大似然估计。

模型分类步骤：（1）计算每个特征的平均值和方差，得到正态分布的密度函数，通过密度函数就能计算的到测试数据的密度函数值。（2）将密度函数值作为条件概率的值。（3）通过朴素贝叶斯方法计算测试样本所属一级标签类别。

分类模型评价：使用 F-Score 对我们的分类方法进行评价。公式：$F_1=\frac{1}{\mathrm{n}}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$，

其中P_i为第i类的查准率，R_i为第i类的查全率。通过 F-Score 可计算得出本次关于一级标签的分类模型的精度约为 71.1%，可认为具有良好的分类效果。

4.3 归类留言

附件 3 中的留言,归类的流程图如图 4 所示：

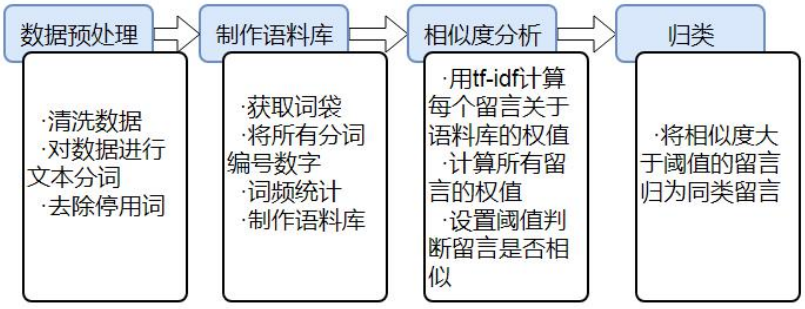

图 4 归类流程图

数据预处理：

对附件 3 中的留言详情数据进行处理，删除本实验没有用到的数据，留下与问题的解决有关的文本数据，接着通过 Python 对文本数据进行分词、去除停用词。

制作语料库：

通过 Python 的操作，首先用 dictionary 方法获得词袋，在词袋中用数字将文本分词后的所有词进行编号，对每个分词进行词频统计，接着制作出一个关于每一条留言详情所有词的语料库，语料库是一组向量，向量中的元素是一个二元组（编号，频次数），每个向量对应每一条留言的关键词组，可看成每个留言的词频向量。

相似度分析：

使用 TF-IDF 模型对语料库建模，建立循环计算每个留言词频向量的 TF-IDF 值，设置阈值为 0.1，通过比较分析每个词频向量的 TF-IDF 值来确定留言是否相似。

归类：

比较每个留言的 TF-IDF 值，大于阈值的留言归为同类留言，建立循环比较分析所有的留言。

4.4 实现答复意见评价

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、可解释性等角度对答复意见的质量进行评价：

相关性：

关于相关性的评价方案如图 5 所示：

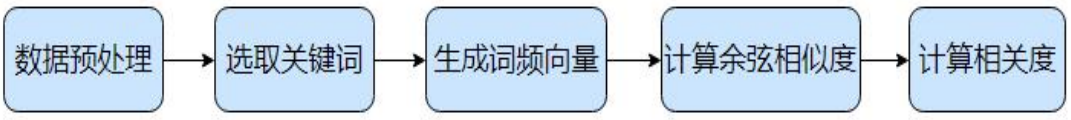

图 5 相关性的评价方案

（1）对附件 4 中的留言详情与答复意见进行数据预处理。

（2）使用 TF-IDF 算法分别选定留言详情与答复意见的关键词。

（3）将留言详情与答复意见中选出的 n 个关键词合并成一个 2n 维的关键词向量，即为生成大合集；再对两者间关键词进行词频统计，分别得到两个关于大合集关键词的词频统计数据，即对应关键词和该关键词出现次数，得到对应的词频向量。

（4）计算两个向量关于大合集的对应词频向量的余弦值。

（5）计算留言与答复的相关度，相关度=所有余弦值的算术平均值，其中余弦值为留言与对应答复的余弦相似度。

通过 Python 进行以上操作，算得关于附件 4 中答复意见与其对应留言的相关度达到 23.3%。

完整性：

根据对附件 4 中答复意见的数据以及实际情况的观察分析，得知回复可分为两种类型回复：可解决回复和不可解决回复，其中，不可解决回复代表该留言工作人员无法解答或解决，将留言转至相关部门答复。

关于完整性的评价方案为：

（1）通常多个文本中会出现同义词、近义词，影响结果的计算，因此，使用 Excel 将所有留言中会出现的同义词、近义词替换为同一个词语，保证统计数量的最大化。其中替换的有：你好-您好、谢谢-感谢、答复如下-回复如下、转至-转交。

（2）通过观察所有回复的词频统计数据，得到可认为与留言规范有关的完整答复应具有的词语，如表 1 所示：

表 1 关于附件 4 留言的词频统计数据

关键词（取前一部分展示）	词频
年	5181
工作	2750
情况	2524
您好	2395
市	2266
网友	2162
县	2100
回复	2054
收悉	1840
相关	1798
建设	1660
感谢您	1564
支持	1452

具有完整性的可解决答复应该同时具有以下词语：①开头：您好（首先问好）、收悉（表示已收到并了解留言）②中间：答复如下（开始回复）③结尾：感谢（表示感谢留言）、*年*月*日（表明答复时间）

具有完整性的不可解决答复应该同时具有以下词语：①开头：您好 ②中间：反映的问题、已转交 ③结尾：*年*月*日

（3）计算完整度，即计算完整性答复占比

使用 Excel 的查找功能，查找出所有可解决回复和不可解决回复中具有完整性的回复，计算完整性答复占比，得附件 4 中答复意见的完整度为 39%。

可解释性：

在答复中运用法律法规、文献等权威资料时，通常使用书名号（《 》）来表示该权威资料，因此，若具有书名号可认为该答复很可能具有可解释性。

通过 Excel 查找到所有具有可解释性的答复，并计算具有可解释性的答复在所有答复中的占比，得到该相关部门对留言的答复意见的可解释度为 29.7%。

5. 结论

对各类社情民意相关文本数据进行分析研究，实现通过计算机来划分各类留言和整理热点留言问题，避免了花费大量时间、精力且效率难以提高的传统工作模式，对提升政府的管理水平和施政效率具有重要意义。本文通过 TF-IDF 算法和高斯朴素贝叶斯分类模型，实现了对各类留言的分类；通过计算词频向量之间的 TF-IDF 权值，设置阈值，比较分析各类留

言的相似度，并通过热度评价指标来整理热点留言；通过计算答复的相关度、完整度、可解释度来评价答复意见的质量。

问题一解题中所用到的 TF-IDF 算法可以有效地找到留言中最具有代表性的关键词，有利于后面高斯贝叶斯根据 tf-idf 的权值向量来进行分类贴标签。问题二解题中迭代归类留言时，先把分好类的留言放入空集合里，再对其余留言进行相似度计算，这样很好避免了留言进行重复的相似度计算。在问题三中，评价相关性使用计算余弦相似度判断文本相似度的方法，在一定程度上具有优点，使用这种方法是一种易于理解且结果易于观察的方法，它可以快捷的计算出文本间的相似度吧，并通过余弦算法的结果（0-1 之间）判断出相似度的大小；完整性和可解释性均是先对所有答复进行词频统计，有数据表明出词语的使用次数，才可确定是否可用这些词语来表示完整性及可解释性。

在本文解题的方法中，我们还存在着不足的地方：问题 1 中使用的停用词词典不能做到齐全，影响数据预处理中脏数据的清除，预处理后文本数据还不够精简，在后面数据处理上会占用更多的空间内存；在问题 2 中，由于制作的语料库过大，是所有留言分词后的词语，可能会影响相似度计算的准确度，影响同类留言的归类；问题 3 中评价完整性时，可能会有代表具有完整性的词语没有考虑到，如替换同义词、近义词时，可能有少数词语缺漏，影响评价完整性。

参考文献

[1]石凤贵.基于 TF-IDF 中文文本分类实现[J].现代计算机,2020(06):51-54+75.

[2]贺科达,朱铮涛,程昱.基于改进 TF-IDF 算法的文本分类方法研究[J].广东工业大学学报,2016,33(05):49-53.

[3]刘惠,赵海清.基于 TF-IDF 和 LDA 主题模型的电影短评文本情感分析——以《少年的你》为例[J].现代电影技术,2020(03):42-46.

[4]王双成,高瑞,杜瑞杰.基于高斯核函数的朴素贝叶斯分类器依赖扩展[J].控制与决策,2015,30(12):2280-2284.

[5]高影繁,马润波,刘玉树.一种快速文本归类算法的设计与实现[J].北京理工大学学报,2006(12):1069-1072.

[6]张俊飞.改进 TF-IDF 结合余弦定理计算中文语句相似度[J].现代计算机(专业版),2017(32):20-23+27.

附录清单

序号	附件名称	备注
1	附件 2.xls	赛题数据的附件 2
2	附件 3.xls	赛题数据的附件 3
3	附件 4.xls	赛题数据的附件 4
4	Stopword.txt	停用词词典
5	代码 1.py	问题 1 的解题代码
6	代码 2.py	问题 2 的解题代码
7	热点问题表.xls	关于附件 3 中排名前 5 的热点问题
8	热点问题留言明细.xls	关于附件 3 中相应热点问题对应的留言信息
9	A.xlsx	关于附件 3 留言的归类结果
10	完整性词频统计.txt	关于附件 4 答复意见的词频统计数据
11	完整性和可解释性.docx	关于附件 4 答复的完整性和可解释性分析
12	代码 3.py	实现问题 3 的评价附件 4 答复相关性的代码
13	B.xlsx	关于附件 4 每条留言与答复之间的余弦相似度