基于自然语言处理技术的智慧政务系统

摘　要

随着网络的发展，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的重要渠道。但是各类社情民意相关的文本数据量不断攀升，也给政府带来许多工作量。借助大数据、 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。

对于赛题的要求，我们构建单标签文本分类模型、热点问题挖掘模型、答复意见评价模型。在以上的模型中采用正则表达式、去除停用词、jieba 分词等进行文本预处理，再将文本向量化表示，通过 TF-IDF 权重将稀疏的词向量矩阵降维处理。在单标签分类模型中采用监督文本分类算法朴素贝叶斯和支持向量机分别进行训练样本，在通过 F-Score 对这两种分类方法进行评价。在寻找热点问题中主要采取文本相似度进行聚类分析。在答复意见评价方案中依据其相关性、完整性、可解释性、及时性将答复质量分为 ABCD 四个等级。在以上实验结果中均得到较好的结果可大大减少相关人员的工作量。

以上模型均在 python 环境中实现。

关键词： jieba；TF-IDF；词向量矩阵；监督学习；文本相似度

Abstract

With the development of the Internet, wechat, microblog, mayor's mailbox, sunshine hotline and other Internet platforms have gradually become an important channel for the government to understand public opinion. However, the amount of text data related to social situation and public opinion is increasing, which also brings a lot of workload to the government. With the development of big data, cloud computing, artificial intelligence and other technologies, the establishment of intelligent government system based on natural language processing technology has become a new trend of social governance innovation and development.

For the requirements of the competition questions, we build a single label text classification

model, hot issue mining model, reply evaluation model. In the above model, we use regular expression, remove stop words, Jieba word segmentation to preprocess the text, and then quantify the text, and reduce the dimension of sparse word vector matrix by TF-IDF weight. In the single label classification model, the supervised text classification algorithm naive Bayes and support vector machine are used to train samples respectively, and the two classification methods are evaluated by F-score. In the search of hot issues, text similarity is mainly used for clustering analysis. According to its relevance, integrity, interpretability and timeliness, the quality of reply can be divided into four grades of ABCD.

All the above models are implemented in Python environment.

Keywords: Jieba; TF IDF; word vector matrix; supervised learning; text similarity

引言

随着网络的发展，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的重要渠道。但是各类社情民意相关的文本数据量不断攀升，也给政府带来许多工作量。借助大数据、 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。

对于本次赛题，主要需要运用文本的分类与聚类方面的技术去解决。另外，在题目中提到的各项指标的定义则是我们需要结合所给数据特征量化，合理分配各项指标的权重。由于中文文本对比英文文本，没有其天然的空格作为词之间的空格。这对计算机来说就较难模拟人对中文句子的理解，所以中文文本预处理将非结构化的数据转换成计算机可理解的结构化处理就由为重要。

在问题一的单标签分类模型有两个核心需要我们去解决。第一是中文分词，中文分词主要基于字符串匹配、基于理解和基于统计的分词方法，在本文我们采用 jieba 分词进行对文本的处理。第二是文本分类的算法设计，机器学习在人工智能的文本分类方面有较好的应用，其主要依赖于学习阶段，即产生一个能够对输入数据所表达的模式进行编码的模型或函数。根据不同的学习机制，机器学习算法基本可以分为三类：监督学习，无监督学习和半监督学习。根据赛题所给的文本-类别数据文件，我们采用监督学习文本分类算法，

常见的有朴素贝叶斯、支持向量机等分类器。

对于问题二中的热点问题挖掘，根据赛题要求，我们不单单要找出排名前 5 的热点问题，还要将其留言明细结果输出出来。这对热点挖掘常用的基于 LDA 的各种模型无法对其较好的应用，而且我们发现对于留言主题那种一句话的文本数据采用传统模型基于字面匹配的文本相似度算法就可得到较好的结果。将同一件事情的留言归在一起后，根据留言个数、时间跨度、点赞数等相关因素初步筛选出 30 类可能是热点问题的主题，再定义指标模型对其进行评分，找出前 5 热点问题。

对于第三题相关部门对留言的答复意见是反映给留言群众的，从群众的角度来看，如果知道答复意见的质量，可以判断用不用更换其他部门反馈意见，而政府则可以从较差的留言答复意见中去改进。所以我们从答复的相关性、完整性、可解释性，及时性入手将答复意见划分为 A、B、C、D 四个评价等级。

目 录

1.群众留言分类

1.1 文本预处理

在留言详情中含有电话号码、标点符号、英文字母等对接下来要做的中文分词带来一定的麻烦，因此我们采用正则表达式去提取留言详情中的中文。正则表达式是对字符串（包括普通字符（例如 a 到 z 之间的字母）和特殊字符（称为“元字符”）操作的一种逻辑公式，就是用事先定义好的一些特定字符、及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。正则表达式是一种文本模式，该模式描述在搜索文本时要匹配的一个或多个字符串。[1]

从网上下载停用词表后，去除附件 2 中留言详情的停用词。人类语言包含很多功能词。与其他词相比，功能词没有什么实际含义。这些功能词的两个特征促使在搜索引擎的文本处理过程中对其特殊对待。第一，这些功能词极其普遍。记录这些词在每一个文档中的数量需要很大的磁盘空间。第二，由于它们的普遍性和功能，这些词很少单独表达文档相关程度的信息。如果在检索过程中考虑每一个词而不是短语，这些功能词基本没有什么帮助。在信息检索中，这些功能词的另一个名称是：停用词（stopword）。称它们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。将这些词扔掉减少了索引量，增加了检索效率，并且通常都会提高检索的效果。

利用 jieba 模块对待分词文本进行分词。加载模块自带 dict.txt 词典，从自带的词典中构建待分词的语句的有向无环图 （DAG）。对于词典中未收录的词，使用 HMM 模型的 Viterbi 算法尝试分词处理； 已收录词和未收录词全部分词完毕后，根据有向无环图的最大概率路径使用 Python 的 yield 语法生成一个词语生成器，逐词语返回。

1.2 利用 scikit-learn 库构建文档-词频矩阵

我们接下来将分词后的文本数据进行结构化处理，并通过 TF-IDF 将稀疏的词向量矩阵降维处理。在 python 的 scikit-learn 库提供了很多数据结构化处理的工具，将这类结构化处理统称为“Feature Extraction”，即“特征抽取”，文本中的词汇出现的次数就属于“特征”中的一种。

sklearn.feature_extraction.tex 模块下定义的 TfidfTransformer 类可以将词频矩阵转换为标准化的 TF 或 TF-IDF 矩阵。CountVectorizer 类可以将文档集合转换为文档-词频矩阵，TfidfTransformer 类可以将文档词频矩阵转换为 TF-IDF 矩阵，而 TfidfVectorizer 类将这两个过程合并在一起，即可将原始文本数据直接转换为 TF-IDF 矩阵。

1.3 单标签文本分类

文本分类：给定文档集 D 与类别集合 C，类别集合包含了 L 个类别与及相应的标签。另外还给定了一个二值函数 F：D×C→{0，1}，即对每个文档-类别对[d_j,c_p]都赋予一个 0 或 1 的值，其中d_j是类别c_p的成员；如果赋值为 0，表示文档d_j不是类别c_p的成员。这样的文本分类是宽泛的，既包含了监督算法又包含了无监督算法。无监督算法是利用文档自身的属性来进行划分，然而在一般情况下，为达到高精度的文本分类，我们应当采用监督算法。如果对分类器没有特别的约束，那么同一文档可能会被赋予两个甚至更多的类别标签。这种情况下，我们称这个分类器属于多标签类型。如果我们要求分类器对每个文档都赋予单一的标签，这个分类器属于单标签类型。

当一个算法使用了人工标注或者人工辅助标注信息时，这样的算法称为监督算法。在标准情况下，类别集合以及每个样本文档所对应的类别是给定的。这些由专家标注的样本组成了训练集，可以用来学习分类函数。如果学习了该函数，那么就可以用它来对未知的新文档进行分类。通常，训练样本的规模越大，分类器的效果就越好。

为了对分类器进行评价，我们把它应用到一组已经事先确定类别的未知数据上，这样的数据集合我们称为测试集。如果分类器能够对测试集中大多数的数据做出正确的分类，么我们就认为训练过程和得到的分类器是合适的。

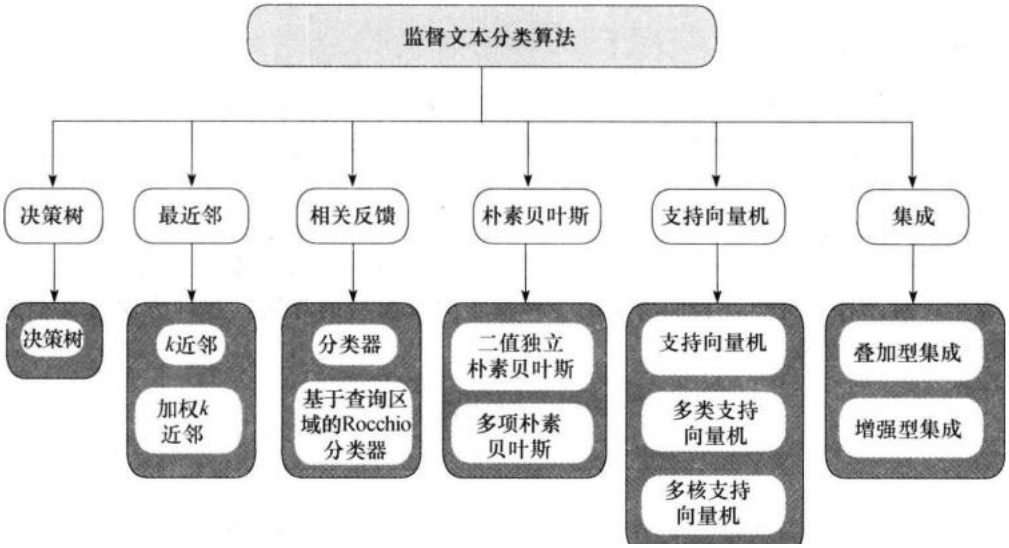

图 1 常用的传统机器学习算法图

1.4 模型评价

文本分类中的精度与召回率是信息检索中精度率与召回率指标的翻版，他们用来衡量文本分类器的质量。对于某个给定类别c_p的精度与召回率数值是按以下方法计算的。

$$P(c_p)=\frac{n_{f,t}}{n_f}$$

$$R(c_p)=\frac{n_{f,t}}{n_f}$$

精度是所有被分类器分配到类别c_p的文档中确实属于类别c_p的文档（根据测试集）所占的比例。召回率是所有确实属于类别c_p的文档（根据测试集）中被分类器正确分配到类别c_p的文档所占的比例。通常，把精度和召回率合并为一个指标会更便于使用，一种最常用的指标叫做 F 测度或 F 值。而最常用的 F 测度形式是通过赋予精度和召回率相同的权重得到的，即令α=1。这一指标也被称为F_1测度，计算形式如下：

$$F_1(c_p)=\frac{2P(c_p)R(c_p)}{P(c_p)+R(c_p)}$$

最终我们用朴素贝叶斯和支持向量机分类器分别进行训练样本，分别得到 F 值 0.87 与 0.88，这良好的数据结果体现该单标签分类模型的有效性，具体过程数据在附件当中。

2. 热点问题挖掘

2.1 TF-IDF 模型分析

TF-IDF（term frequency－inverse document frequency，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。TF-IDF 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[2]。

TF-IDF 的主要思想是：如果某个单词在一篇文章中出现的频率 TF 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。其中 TF 是词频，表示词条（关键字）在文本中出现的频率。IDF 是逆向文件频率，表现一个词语出现

的普遍程度。TF-IDF：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。

$$词频(TF) = \frac{某个词在文章中的出现次数}{文章的总词数}$$

图 2 TF 计算公式

$$逆文档频率(IDF) = \log(\frac{语料库的文档总数}{包含该词的文档数 + 1})$$

图 3 IDF 计算公式

$$TF - IDF = 词频(TF) \times 逆文档频率(IDF)$$

图 4 TF-IDF 计算公式

TF-IDF 算法优缺点：

优点:算法简单，快速，并且容易实现；能满足大量数据处理，使用较为广泛。

缺点：简单结构并没有考虑词语的语义信息，无法处理一词多义与一义多词的情况；IDF 的简单结构并不能有效地反映单词的重要程度和特征词的分布情况，使其无法很好地完成对权值调整的功能，所以 TF-IDF 算法的精度并不是很高。

2.2 文本相似度

本次我们将 TF-IDF 通过计算每个留言主提之间的文本相似度，然后根据相似度进行分类。相似度，实质就是计算个体间相似程度。

计算句子之间的相似度的方法是基于余弦相似度模型。余弦相似度就是通过一个向量空间中两个向量夹角的余弦值作为衡量两个个体之间差异的大小。把 1 设为相同，0 设为不同，那么相似度的值就是在 0~1 之间，所有的事物的相似度范围都应该是 0~1，如果不是 0~1

的话，就不是我们应该研究的事了，那是神经学家和生物学家的事了。余弦相似度的特点是余弦值接近 1，夹角趋于 0，表明两个向量越相似。

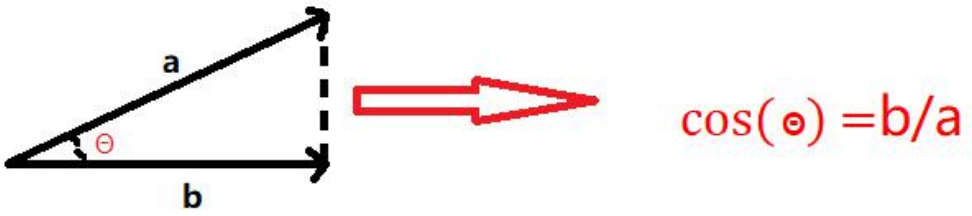

图 5 余弦相似度模型示列图

三角形越扁平，证明两个个体间的距离越小，相似度越大；反之，相似度越小。但是，文本的相似度计算只是针对字面量来计算的，也就是说只是针对语句的字符是否相同，而不考虑它的语义，那是另外一个研究方向来着。比如，句子 1：你真好看:。句子 2：你真难看。这两句话相似度 75%，但是它们的语义相差十万八千里，可以说是完全相反。又比如，句子 1：真好吃。句子 2：很美味。两个句子相似度为 0，但是语义在某个场景下是一致的。所以，简单结构并没有考虑词语的语义信息，无法处理一词多义与一义多词的情况。但是，本次不需要考虑到多词同义或一词多义的情况。所以，该模型比较契合本次计算。[3]

以下将简单介绍如何计算计算句子之间的相似度。我们利用 python 实现这个算法，就是把两个个体转换为向量，然后通过这个公式求出最终解.下面将以流程图展示该过程。

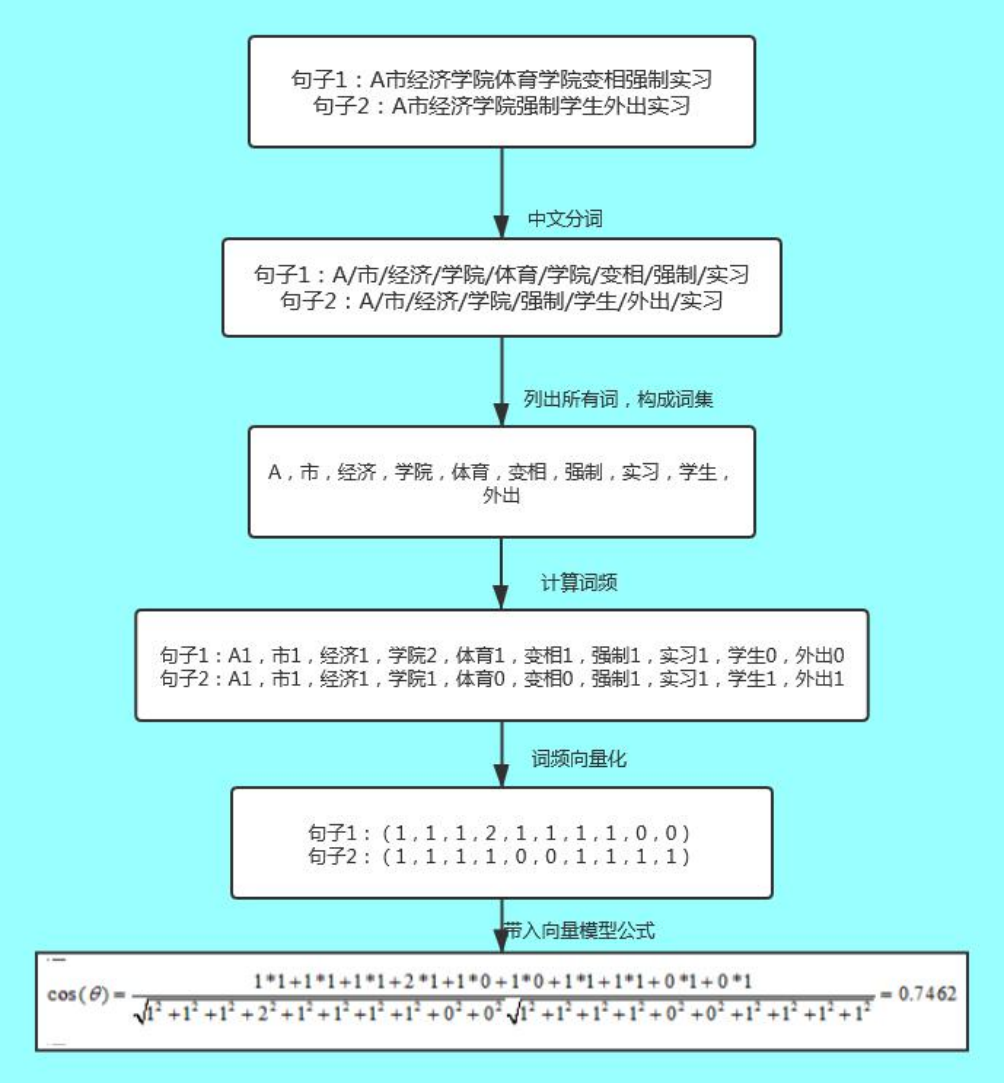

图 6 句子相似度计算流程图

由图可知，两个句子的相似度计算的步骤是：

①通过中文分词，把完整的句子根据分词算法分为独立的词集合

②求出两个词集合的并集(词包)

③伤计算各自词集的词频并把词频向量化

④带入向量计算模型就可以求出文本相似度

2.3 基于 TF-IDF 算法实现问题初步归类

本次利用 python 实现算法，计算出每个留言主题之间的文本相似度，然后提取出文本相似度大于 0.4 的留言主题归为同一类。主要步骤如下流程图：

第一步，读入数据即读入附件 3 中的留言主题作为原文本；

第二步，用 python 的 jieba 库进行分词；

第三步，导入文档“stop-word”，剔除掉停用词；

第四步，计算出每个词的词频；

第五步，制作语料库，首先用 dictionary 方法获取词袋，用数字对所有词进行编号，使用 doc2bow 制作语料库；

第六步，载入对比文档，本次的测试文档跟原文档是同一文档；

第七步，建立 for 循环语句，为及对文档里的每一条留言主题与其他的留言主题分别进行对比做准备；

第八步，用同样的方法把对比文本也转发为二元祖向量，即同样建立新语料库；

第九步，使用 TF-IDF 模型对语料库建模，计算每个词的 TF-IDF 值；

第十步，得出所有原文本与对比文本的相似度；

第十一步，给每个相似度结果建立索引；

第十二步，筛选出相似度大于 0.4 的结果，存储在列表里。继续进行下一轮循环；

第十三步，输出所有分好类的列表存储结果；

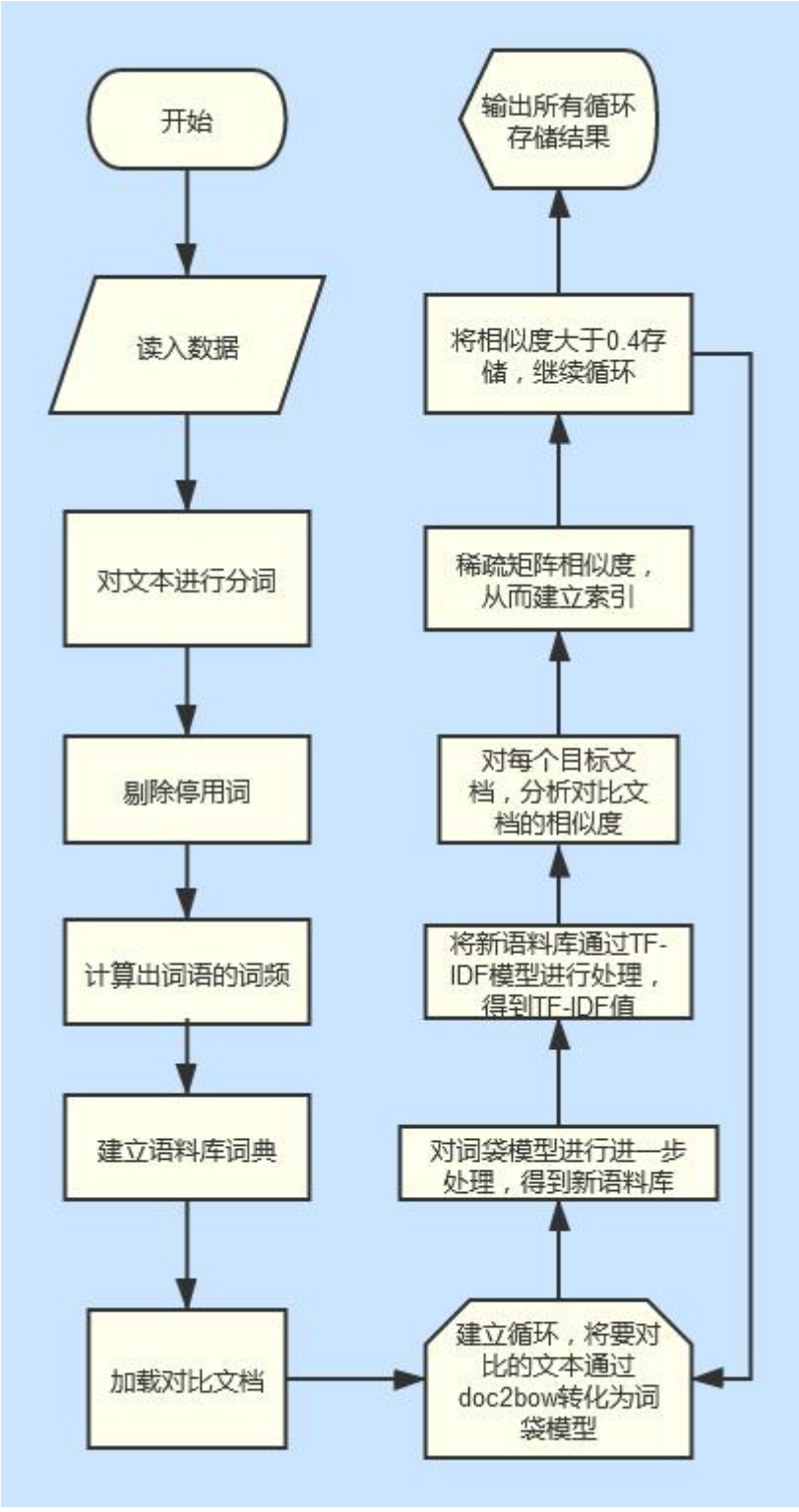

图 7 基于 TF-IDF 算法的流程图

2.4 文本聚类

经过 TF-IDF 算法的初步筛选，为了进一步得出更为理想的数据，本文将运用文本聚类

模型继续进一步进行筛选。本次还是继续使用 python 对同类的问题进行聚类处理。即将初步数据中的结果中同类问题相似大于 0.4 合并到一起，然后把其中重复的数据剔除掉，再选出同一类问题中出现次数最多的数据。由于考虑到点赞数和反对数对热度评定的影响。同时再选出同类问题中点赞数和反对数大于 20 的数据，最后得出 38 类问题.

2.5 热度评价

根据文本聚类得出的 38 类问题，我们将使用 hotness(t)模型计算出各类问题的热度。综合影响问题热度的各种因素，本文提出了以下热点定义模型：

$$\text{hotness}(\text{t}) = \frac{\text{n}}{N} * \left(\frac{\text{m}}{T_{\text{a}} - T_{\text{b}}} + \frac{N_{\text{t}}}{\sqrt{\sum_{\text{t=1}}^{\text{x}} N_{\text{t}}}} \right)$$

其中，N 为问题 t 中的总投票数（即赞成票数+反对票数），n 为投票数中的赞成票总数，m 为问题 t 在附近 3 中的提出次数。T_{a} 表示问题 t 最晚提出的时间，T_{b} 表示问题 t 最早提出时间。最后利用 ecxel 软件计算出每一类问题的热度。结果如下表所示。

表 1 热度计算结果表

问题 ID	留言主题	最晚提出时间	最早提出时间	赞成数	反对数	问题提出次数	时间跨度(天)	投票数	热度
2	举报广铁集团在伊景园滨河苑项目非法绑定车位出售	2019/9/1 14:20	2019/7/7 7:28	25	1	52	56	26	0.902351
16	A 市五矿万境 K9 县房屋出现质量问题	2019/9/19 17:14	2019/5/5 13:52	2106	0	7	137	2106	0.850841
36	承办 A 市 58 车贷案警官应跟进关注留言	2019/3/1 22:12	2019/2/21 18:45	1554	0	2	8	1554	0.840126
32	A7 县东六路下穿长永高速在月底能否如期通车	2019/10/12 13:36	2019/10/9 10:14	41	1	2	3	42	0.666363
3	A2 区丽发新城附近修建搅拌厂噪音、灰尘污染	2020/1/26 19:47:11	2019/11/2 10:18:00	49	2	48	85	51	0.561168

3.答复意见评价方案

本题最关键的是对于相关性的判断，相关性是后续评价的基础，所以相关性模型的好话直接影响了后续评价的效果。对于相关性，本文主要采用文本相似度来进行衡量。本题的相似度一定程度上与问题二相类似，但是却不能使用第二题的模型，因为 doc2bow 适合短文本,而附件 4 中的留言主题和答复意见普遍较长,因此本题的相似度计算采用 Word2Vec 模型。

3.1 Word2Vec 模型分析

Word2vec 是 Google 公司于 2013 年发布的一款基于 Deep Learning 的开源工具包，也是首款面向大众的 Deep Learning 学习工具。Word2vec（Word To Vector），顾名思义，它可以将词汇转换成向量形式，从而把对文本的处理转化为向量空间中的向量运算，方便地完成各种自然语言处理任务。Word2vec 以文本语料库作为输入，首先在训练文本数据集中构建一个词汇表，然后训练出每个单词的词向量作为输出，产生的词向量文件可以作为特征向量供后续的自然语言处理和机器学习等算法使用。

Word2Vec 包含了两种训练模型，分别是 CBOW 和 Skip_gram 模型，本文主要使用 Skip_gram 模型，如图 8 所示。

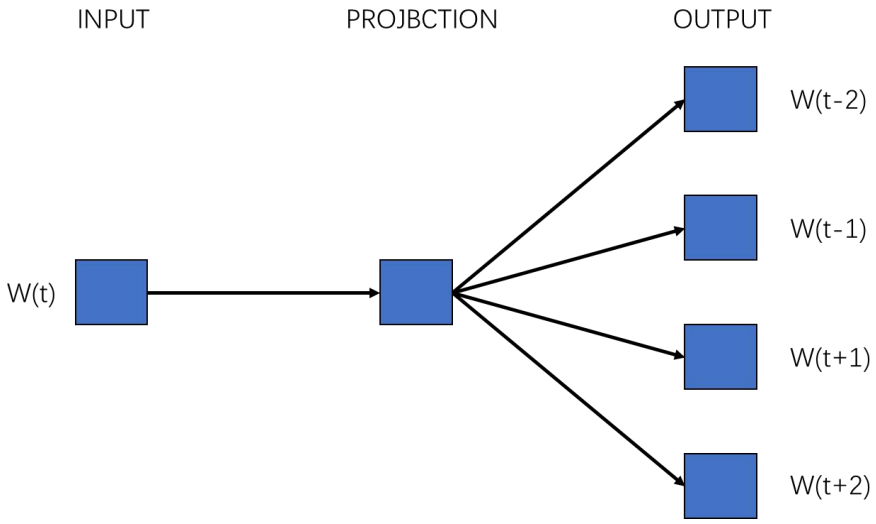

图 8 Skim_gram 模型

Skip_gram 模型是利用当前词预测其上下文。这个模型的核心算法 Tomas Mikolov 在

文献[4][5]中给出了核心算法详细过程。

Word2Vec 模型的优缺点[6]：

优点：

①由于 Word2Vec 会考虑上下文，相较于 Embedding 方法，效果要更好；

②相较于 Embedding 方法维度更少，避免了维数灾难，所以速度更快；

③通用性很强，可以用在各种 NLP 任务中。

缺点：

①由于词和向量是一对一的关系，所以多义词的问题无法解决；

②Word2vec 是一种静态的方式，虽然通用性强，但是无法针对特定任务做动态优化。

3.2 基于 Word2Vec 算法实现

本问利用 python 实现算法，主要步骤如图 9。

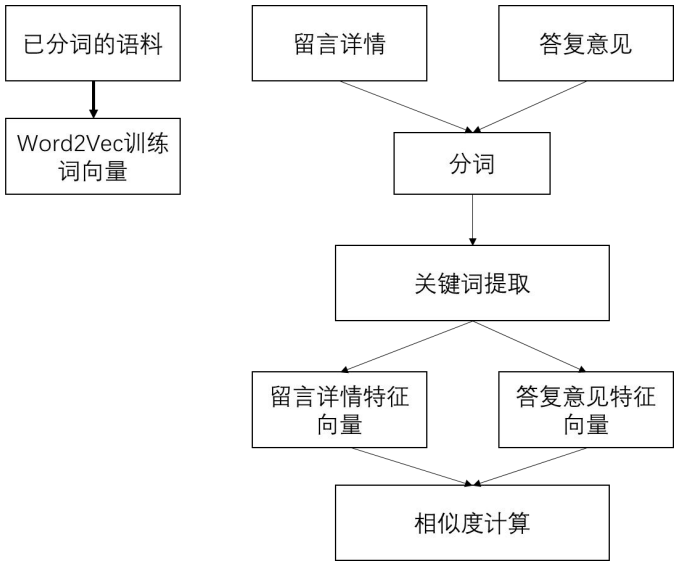

图 9 Word2Vec 模型计算相似度

3.3 评价模型

3.3.1 相关性

对于相关性，本文以附件 4 中的留言详情以及答复意见的文本相似度进行评价，将不相关答复、转交给其他部门的答复、空白答复、收悉但无答复等答复定义为不相关的答复。但部分答复意见中含有如“教育局”、“公安局”等关键词且不相关，通过 Word2Vec 模型计算相似度则会导致相似度偏高，从而影响评价的效果。

通过观察，发现这类答复意见普遍较短通常小于 100 个字符。所以再将这部分的答复进行聚类，从而划分为 E 类。

3.3.2 完整性

对于完整性，本文采用文本长度进行度量，在答复意见具备相关性的条件下，答复意见的文本长度越长，则答复越完整。本文以 400 为界限，划分完整与不完整。

3.3.3 可解释性

可解释性，本文采取提取答复意见中的相关文件和政策的名称，若答复意见具备相关性和完整性且答复意见中具备相关的文件和政策，那么其具有可解释性。

3.3.4 及时性

此外本文还添加了及时性这一评价指标，若回复不及时，则会造成市民重复留言造成，留言中重复的留言增多，加重了市政平台的负担，同时会引起市民对有关部门的反感。对于及时性，若答复意见具备上述性质，则答复时间与留言时间的差值越小，则评价越及时。本文以 2 个星期为界限划分及时与不及时。

3.4 模型的流程

第一步，读入数据即读入附件 4。第二步，用 python 的 jieba 库对进行分词。第三步，导入文档“停用词表.txt”，剔除停用词。第四步，利用 Word2Vec 进行词向量的训练。第五步，计算留言详情与答复意见的相似度，第六步，筛选出字符长度小于 100 的答复意见进行聚类。第七步，将相似度小于 0.3 的答复和聚类的结果中类元素大于 3 的答复划分为 E 类，其余为 D 类。第八步，将 D 类答复中文本长度大于 400 的答复划分为 C 类。第九步，将 C 类答复文本中含有政策或者文件的答复划分为 B 类。第十步，将 B 类答复中答复时间间隔小于 2 个星期的回复划分为 A 类。具体流程图如图

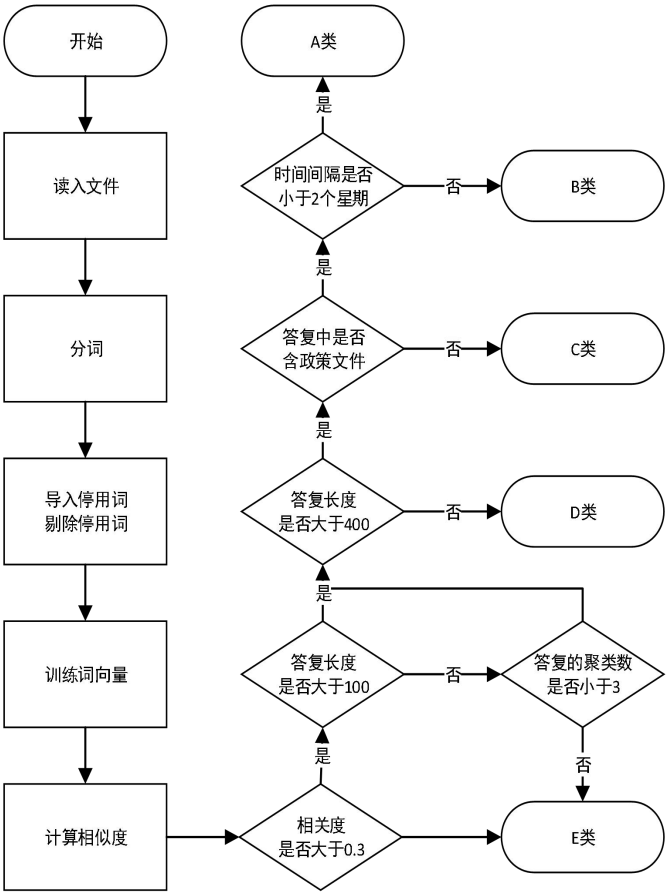

图 10 评价模型流程图

总结

综上所述，通过以上三个模型建立的智慧政务系统极大减轻的政府的工作量。实验结果根据检验，模型均有较好的有效性。还有一些难题需要我们去加以改进，如在分类模型中面对无标注数据该如何处理，避免热点问题中不因为没有合理定义一些重要指标，导致结果有较大分歧，在答复意见如何更好的解释质量的好坏，建立一个规范系统。

参考文献

[1]胡军伟, 秦奕青, 张伟. 正则表达式在 Web 信息抽取中的应用[J]. 北京信息科技大学学报(自然科学版),2011,26(6):86-89.

[2]TF-IDF 算法： https://blog.csdn.net/asialee_bird/article/details/81486700

[3] NLP 文本相似度(TF-IDF): https://www.cnblogs.com/liangjf/p/8283519.html

[4]Mikolov T, Chen K, Corrado G, et al.Efficient Estimation of Word Representations in Vector Space[J].Computer Science, 2013。

[5]LeQV,Mikolov T.Distributed Representations of Sentences and Documents[J].Computer Science, 2014,4:1188-1196。

[6]https://blog.csdn.net/weixin_43612023/article/details/101475460。