“智慧政务”中的文本挖掘应用

目　录

摘　要

随着政务信息化建设，问政方式的演变与科学技术的发展并驾齐驱,技术的完善和公众意识的觉醒,供公众参与问政的方式和途径越来越多样。网络问政平台在政府和群众之间起到"民意连心桥"作用,通过平台,群众可以问政于政府，投诉举报、求助咨询；政府可以问计于民,倾听民声、汇聚民意、凝聚民智。越来越多的数据使问政服务迈向大数据时代。

本文通过使用了词嵌入 Word2vec 的 CBOW（Continuous Bag of Words）模型，TF-IDF 模型，句子相关度计算模型，朴素贝叶斯的多项式模型等几个常见的文本挖掘模型对网络问政平台的留言信息以及答复等相关数据进行剖析挖掘，以期解决政府提升管理水平以及施政效率等问题，对建立基于自然语言处理技术的智慧政务系统建言献策。

关键词：政府；查问；模型

Abstract

With the informatization of government affairs, the evolution of the way of inquiring about politics, the development of science and technology, the improvement of technology and the awakening of public awareness, the ways and means for the public to participate in inquiring about politics are becoming more and more diverse. The online questioning platform serves as a bridge between the government and the masses. Through the platform, the masses can ask questions from the government, complain, report and seek help; the government can ask questions from the people, listen to the voices of the people, gather public opinions, Gather people's wisdom.

This article uses several common text mining models such as the CBOW model embedded in Word2vec, the TF-IDF model, the sentence correlation calculation model, the Naive Bayes polynomial model, and so on. Analyze and mine the data, with a view to solving the problems of the government's improvement in management level and governance efficiency, and suggestions for establishing a smart government system based on natural language processing technology.

Key Words：Government；Inquire；Model

1.目标分析

1.1 对群众留言进行分类处理

随着互联网快速发展，群众问政的留言错综复杂，数据量也越来越大，如何有针对性并且高效地分析群众留言变成一个繁重的工程，所以根据需求，在收集到群众留言的文本信息后直接建立关于留言内容的一级标签分类模型。

1.2 挖掘群众留言的排名前五的热点问题

留言分类后通过几种指标及时发现关于留言内容前五的热点主题，以便相关部门有针对性地处理，提高服务效率。

1.3 答复意见评价改进

将留言与答复进行分析，包括答复与留言内容的相关性、答复的完整性、答复的可解释性，需要解决如何对答复评价、实现并进行改进。

2.数据理解

对赛题给出的 4 份附件数据进行理解、分析，确定进行挖掘方向。

附件一：给出三个级别的分类标签做建模参考。

给出带标签的留言主题与详情，根据所给的一级标签进行训练建模。

附件二：对留言主题赋予热度指标，进行热度排序。

附件三：热度指标 1：根据算出的不同地址不同类别所出现的频数；热度指标 2：将留言主题持续的时间长度来进行分类，时间距离越接近现时、持续时间越长的问题热度指数越高，时间距离越远离现时持续时间越短的热度指数越低；热度指标 3：算出留言详情的点击率，根据点击率来赋热度指标。

附件四：给出问政留言的对应答复与答复时间，根据答复时间长短，答复内容相似度等指标评价答复内容。

3.数据准备与预处理

数据来源：第八届“泰迪杯”数据挖掘挑战赛 C 题数据

数据量：附件一：517 条 附件二：9210 条 附件三 4326 条 附件四：2816 条

数据清洗：

重复值：无

空值：无

分词：采用 jieba 分词的隐马尔科夫链模型

去停用词：中文停用词列表

字符串规范化：处理转义字符

4.模型介绍

4.1TF-IDF 模型原理及算法

4.1.1 模型原理

TF（Term Frequency，缩写为 TF）：也就是词频

即一个词在文中出现的次数，统计出来就是词频 TF，显而易见，一个词在文章中出现很多次，这个词对文本来说是非常重要的，然后使用一些停用词的语料库去掉“的”“是”这样的词， 避免干扰统计的情况。

IDF（Inverse Document Frequency，缩写为 IDF）：称为逆文档频率

衡量一个词是不是常见词。如果某个词比较少见，但是它在这篇文章中多次出现，那么它很可能就反映了这篇文本的特性，正是我们所需要的关键词。当词频相同时，对词设置重要性调整系数，即赋予权重，这个权重就是逆文档频率，它的大小与一个词的常见程度成反比。

知道了"词频"（TF）和"逆文档频率"（IDF）以后，将这两个值相乘，就得到了文本一个词的 TF-IDF 值。某个词对文章的重要性越高，它的 TF-IDF 值就越大。所以，排在最前面的几个词，就是文档的关键词。

4.1.2 模型计算公式

4.1.2.1. 计算词频 TF

考虑到文本有长短之分，为了便于不同文本的比较，进行"词频"标准化。

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

或

$$\text{词频(TF)} = \frac{\text{某个词在文本中的出现次数}}{\text{该文本出现最多次数的词的出现次数}}$$

4.1.2.2. 计算逆文档频率 IDF

需要一个语料库（corpus），用来模拟语言的使用环境。

$$\text{逆文档频率(IDF)} = \log\left(\frac{\text{语料库的文本总数}}{\text{包含该词的文本数}+1}\right)$$

如果一个词越常见，那么分母就越大，逆文档频率就越小越接近 0。分母之所以要加 1，是为了避免分母为 0（即所有文档都不包含该词）。log 表示对得到的值取对数。

4.1.2.3. 计算 TF-IDF 值

$$\text{TF} - \text{IDF 值} = \text{词频（TF）} \times \text{逆文档频率（IDF）}$$

可以看到，TF-IDF 与一个词在文档中的出现次数成正比，与该词在整个语言中的出现次数成反比。所以，自动提取关键词的算法就很清楚了，就是计算出文档的每个词的 TF-IDF 值，然后按降序排列，取排在最前面的几个词。

4.2 词嵌入 Word2vec 的 CBOW（Continuous Bag of Words）

4.2.1 模型原理

先讲一讲 one-hot 词向量和 distributed representation 分布式词向量

one hot 的表示形式：word vector = [0,1,0,…,0]，其中向量维数为词典的个数|V|，当前词对应的位置为 1，其他位置为 0。distributed 的表示形式：word vector = [0.171,-0.589,-0.346,…,0.863]，其中向量维数需要自己指定（比如设定 256 维等），每个维度的数值需要通过训练学习获得。

虽然 one-hot 词向量构造起来很容易，但通常并不是一个好选择。一个主要的原因是，one-hot 词向量无法准确表达不同词之间的相似度，如我们常常使用的余弦相似度。对于向量 x,y∈Rd，它们的余弦相似度是它们之间夹角的余弦值

$$\frac{x^T y}{\|x\|\|y\|} \in [-1,1]$$

word2vec 工具的提出正是为了解决上面这个问题 。它将每个词表示成一个定长的向量，并使得这些向量能较好地表达不同词之间的相似和类比关系，它的目标是将一个词表示成一个向量。

word2vec 是 Google 于 2013 年开源推出的一个用于获取 word vector 的工具包，它简单、高效，因此引起了很多人的关注。 word2vec 的作者 Tomas Mikolov 在两篇相关的论文[1,2]中并没有谈及太多算法细节，但是开创了无监督词嵌入的新局面，让大量之后的 NLP 工作基于 word2vec（或类似的改进工作）的词嵌入再做其他任务。

举个简单例子，判断一个词语的情感，是积极还是消极。用机器学习的思路，我们有一系列样本(x,y)，这里 x 是词语，y 是它们的情感类别，我们要构建 f(x)->y 的映射，但这里的数学模型 f（比如神经网络、SVM）只接受数值型输入，而 NLP 里的词语，是人类的抽象总结，是符号形式的（比如中文、英文、拉丁文等等），把人类理解的词语表示成在数学空间里的“词语”，这个数学空间里的“词语”能够帮助计算机去理解我们人类的语义，所以需要把他们转换成数值形式，或者说——嵌入到一个数学空间里，这种嵌入方式，就叫词嵌入（word embedding），而 Word2vec ，就是词嵌入（ word embedding）的一种。

Word2vec 中两个重要模型是：CBOW 模型和 Skip-gram 模型

因为这里我们使用的是 CBOW 模型，所以下面简单介绍下 CBOW 模型的原理

拿一个词语的上下文作为输入，来预测这个词语本身，就是 CBOW 模型。注意：在 Hidden layer（隐藏层神经元）上没有激活功能，所以有些文章上的图示中没有标示出 hidden layer，而是直接就到输出层，或者是将 hidden layer 称作为投影层，为什么呢，因为这个所谓的隐层的激活函数其实是线性的，所以有的人就不叫它隐藏层了，后面就称它为投影层吧。后面的输出神经元使用 softmax 激活函数。

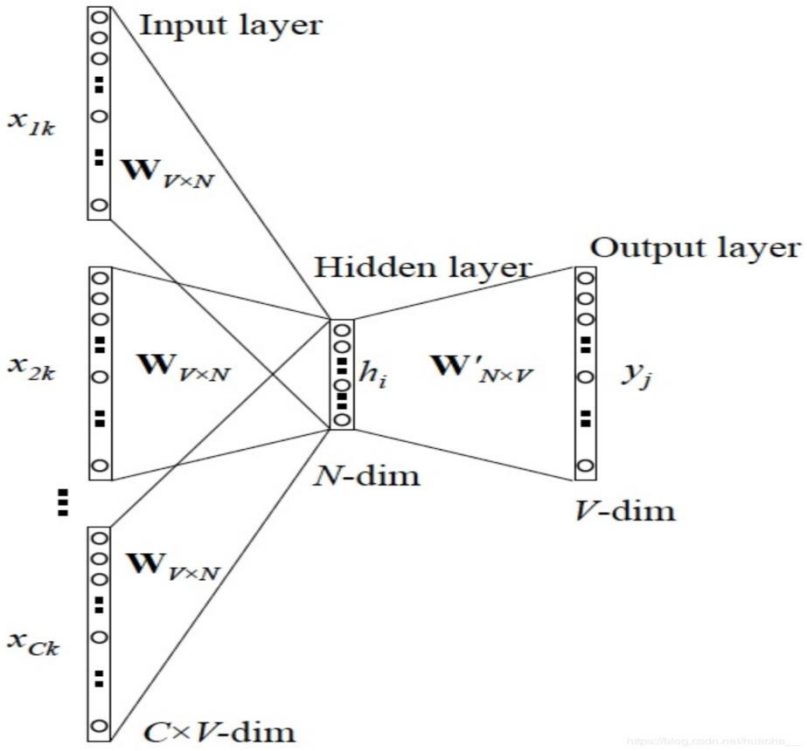


图 4.2.1　CBOW 模型剖析

上图中我们可以理解为 C 个输入单词的维度是 V 维（可以理解为词库中共有 V 个词，那么 V 维 onehot 向量就可以唯一的表示这个词语），当语料库中的单词数量很多的时候，V 值会超级大。CBOW 模型是拿一个词语的上下文作为输入，来预测这个词语本身（中心词周围的多个词来预测这个中心词），那么对应到上图中，输入就是有 x1k、xCk、...、xck 这些上下文词语共 C 个，每一个的长度是 V，输出就是 y 这个中心词 1 个，长度也是 V。在这个网络中我们的目的不是跟一般的神经网络一样去预测标签，而是想要得到完美的参数：权重，X 和这个权

重相乘能够唯一的表示这个词语，同时需要提到一点的是，这个词向量的维度（与隐含层节点数一致）一般情况下要远远小于词语总数 V 的大小，所以 Word2vec 本质上是一种降维操作。

4.2.2 模型计算公式

CBOW 模型假设基于某中心词在文本序列前后的背景词来生成该中心词。例如，在本题的一条文本序列“A 市”“ 限卖”“ 房产”“ 政策”“ 一刀切”里，以“房产”作为中心词，且背景窗口大小为 2 时，CBOW 模型关心的是，给定背景词“A 市”“ 限卖”“ 政策”“ 一刀切”生成中心词“房产”的条件概率，如图 2 所示，即

$$P(\text{“房产”}|\text{“A 市”},\text{“限卖”},\text{“政策”},\text{“一刀切”})$$

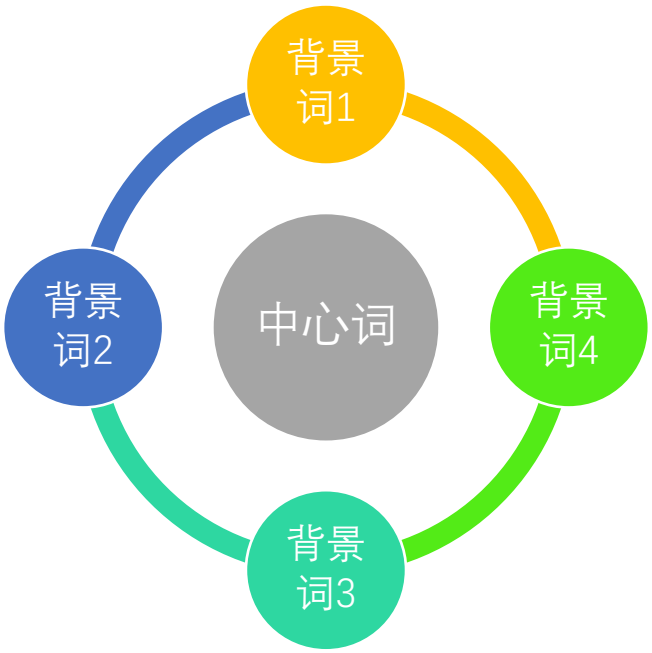


图 4.2.2　CBOW 模型关心给定背景词生成中心词的条件概率

因为 CBOW 模型的背景词有多个，我们将这些背景词向量取平均，然后计算条件概率。设 $v_i \in R^d$ 和 $u_i \in R^d$ 分别表示词典中索引为 i 的词作为背景词和中心词的向量。设中心词ω_c 在词典中的索引为 c，背景词ω_{o1}，…，ω_{o2m} 在词典中的索引为 o1，…，o2m，那么给定背景词生成中心词的条件概率

$$P(w_c \mid w_{o1}, \ldots, w_{o2m}) = \frac{\exp\left(\frac{1}{2m}\boldsymbol{u}_c^\top(\boldsymbol{v}_{o1} + \ldots + \boldsymbol{v}_{o2m})\right)}{\sum_{i \in \mathcal{V}} \exp\left(\frac{1}{2m}\boldsymbol{u}_i^\top(\boldsymbol{v}_{o1} + \ldots + \boldsymbol{v}_{o2m})\right)}.$$

为了让符号更加简单，我们记 Wo={ωo1，…，ωo2m}，且v̄=(v o1，…，v o2m)/(2m)，那么上式可以简写成：

$$P(w_c \mid \mathcal{W}_o) = \frac{\exp\left(\boldsymbol{u}_c^\top \bar{\boldsymbol{v}}_o\right)}{\sum_{i \in \mathcal{V}} \exp\left(\boldsymbol{u}_i^\top \bar{\boldsymbol{v}}_o\right)}.$$

给定一个长度为 T 的文本序列,设时间步 t 的词为ω(t),背景窗口大小为 m。CBOW 模型的似然函数是由背景词生成任一中心词的频率

$$\prod_{t=1}^{T} P\left(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)}\right).$$

训练 CBOW 模型

CBOW 的最大似然估计等价于最小化损失函数

$$-\sum_{t=1}^{T} \log P\left(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)}\right).$$

注意到

$$\log\, P(w_c \mid \mathcal{W}_o) = \boldsymbol{u}_c^\top \bar{\boldsymbol{v}}_o - \log\, \left(\sum_{i \in \mathcal{V}} \exp\left(\boldsymbol{u}_i^\top \bar{\boldsymbol{v}}_o\right)\right).$$

通过微分，我们可以计算出上式中条件概率的对数有关任意一个背景词向量 v oi（i=1，…，m）的梯度

$$\frac{\partial \log\, P(w_c \mid \mathcal{W}_o)}{\partial \boldsymbol{v}_{o_i}} = \frac{1}{2m}\left(\boldsymbol{u}_c - \sum_{j \in \mathcal{V}} \frac{\exp(\boldsymbol{u}_j^\top \bar{\boldsymbol{v}}_o)\boldsymbol{u}_j}{\sum_{i \in \mathcal{V}} \exp(\boldsymbol{u}_i^\top \bar{\boldsymbol{v}}_o)}\right) = \frac{1}{2m}\left(\boldsymbol{u}_c - \sum_{j \in \mathcal{V}} P(w_j \mid \mathcal{W}_o)\boldsymbol{u}_j\right).$$

有关其他词向量的梯度同理可得，背景词向量作为词的表征向量，实际上我们就是取得 input-hidden 这个词向量（weight 矩阵），而不是后面带着 loss 那一部分，这样我们也很容易可以对 loss（训练方法）进行修改。

4.3 句子相关度计算模型

4.3.1 词形相关度

词形相关度主要依赖于两个短文本的共现词语的频率，即两个待比较的文本经过分词、停用词过滤等预处理之后，两个短文本中共现词语的频率越高，相关度越大。在计算频率时，用出现相同词语的次数来计算，当同一词语在两个短文本中出现的次数不同时，以出现次数少的计数，词形相关度可用公式 4-1 计算：

$$Sim_{sa}(A, B) = \frac{same(A, B)}{count(A) + count(B) - same(A, B)}$$

其中，same(A,B)表示两个短文本中相同词语的次数，而 count(X)表示短文本 X 包含的词语个数（包含相同词语）。

4.3.2 词序相关度

词序指的是构成文本的基本元素所组合的顺序。在中文文本中，词语所在的位置能反映出该文本的特殊含义，因此对于文本理解起到了不可忽略的作用，甚至是决定意义。

在对短文本语义相关度研究中，需要考虑词序，例如:“青蛙吃昆虫”与“昆虫吃青蛙”两句话的词形相似度为 1,而语义相似度并不一致。在相关度计算中，词序可以作为一个权重较弱的特征来考虑。短文本的词序主要表现为词语在每个短文本中序号的差值，设 same（A,B）为在短文本 A 和 B 中共出现且只出现一次的词语集合，对短文本 A 中的词语依次进行编号，按照此编号来形成短文本 B 的词序向量，进而得到该向量所构成的自然数的逆序数，代表两个短文本中词语顺序不同的个数，逆序数若越大，词序相似度就越低。因此，词序相似度可用如下公式计算：

其中，Inverse(A,B)表示短文本 B 相对于 A 的逆序数，|same(A,B)|same(A,B)-1|为逆序数最大值的 2 倍。

计算词序的过程如图 4.1 所示，通过计算得出该文本的词序向量 321 的逆序数为 3。说明文本 B 相对于文本 A 有 3 处词序不同的情况，分别是:（青

蛙，吃)、(青蛙，昆虫)、(吃，昆虫)。通过计算可得出两句话的词序相关度为0。

青蛙吃昆虫——>青蛙(1),吃(2),昆虫(3)

昆虫吃青蛙——>昆虫(3),吃(2),青蛙(1) ——>计算 3 2 1 的逆序数

$$Sim_{inv}(A,B)=\begin{cases}1-\dfrac{2Inverse(A,B)}{|same(A,B)-1||same(A,B)|} & |same(A,B)|>1\\ 1 & |same(A,B)|=1\\ 0 & |same(A,B)|=0\end{cases}$$

4.3.3 基于词对的语义相关度

在计算短文本相关度时，仅考虑词形、词序等句子结构具有一定的局限性，例如：“互联网的迅速发展使地球成为了信息网络村”与“智能终端和 4G 网络的出现标志着科技新时代的到来”两句经过分词处理后，得到的词语向量分别为（互联网，地球，信息，网络村）和（智能终端、4G 网络、科技、时代），两句话的共现词语数为 0，通过计算可知两短文本之间的词形相关度与词序相关度均为 0，与人工理解出现了差异。从分词结果可以看出“互联网”与“4G 网络”、“科技”两对词语间的相关度较高，“信息”与“科技”之间也具有一定的语义相关度。因此在计算短文本相关度时，除了考虑句子结构之间的信息，还需考虑句子的深层语义信息。

短文本在语义上的相关度依赖于词对之间的关联，综合利用短文本的结构相关性以及语义上的相关性可以更好地判断两个短文本之间的相关程度，并且语义上的相关性占有很大的比重。传统的短文本相关度计算方法是通过对各词对的相关度求平均值来获取的，为了突出更强的语义信息，本文选取两个词对中相关度最高的词对计算。

4.3.3.1 词对相关度矩阵构建

对于待比较的两个短文本 A,B，首先经过分词以及停用词过滤等，由于维基百科概念解释文档只包含了名词，所以只考虑具有实际意义的名词。在构建矩阵

之前，首先得到两个短文本的特征词集合向量

A={a1,a2,.....,an}、B={b1,b2.....,bm}，通过两向量的笛卡尔积计算两词语的相关度，构建的两个向量词对相关度矩阵如下：

$$S = \begin{Bmatrix} s_{11} & s_{12} & \dots & s_{1m} \\ s_{21} & s_{22} & \dots & s_{2m} \\ \dots & \dots & \dots & \dots \\ s_{n1} & s_{n2} & \dots & s_{nm} \end{Bmatrix}$$

其中，s_{ij}表示短文本 A 中的第 i 个特征词与短文本 B 中第 j 个特征词之间的相关度。

4.3.3.2 最大匹配组合选择

本算法从相关度矩阵 M 中由大到小地选取 min(m,n)个词语组合的相关度值，要求每个词语只出现一次，即在另一短文本中找到与其互为相关度最大的词。选取的规则为：选择矩阵中相关度最高的词a_{ij}，即两个短文本中相关度最高的词语对，去掉第 i 行第 j 列，形成一个(m-1)×(n-1)的矩阵，将该矩阵作为待选择的矩阵，重复以上步骤直到矩阵为空。经过选择后，设最大序列为 maxL={a23,a12,a3i,...aml}。将平均值作为两个短文本之间的相关度，公式为：

$$\mathrm{Sim}_{\mathrm{word}}(A, B) = \frac{a_{23} + a_{12} + a_{3i} + \dots + a_{m1}}{\min(m, n)}$$

结合词形词序的计算，可以得出基于词对的语义相关度的计算方法，流程图如图 4-2 所示，对于带比较的短文本，经过预处理得到短文本向量，由于提取的语料库限定为名词，因此此处只保存具有实际意义的名词。首先，通过短文本向量的构造计算词形相关度、词序相关度，记为结构相关度，接着利用词语间相关度的方法构造相关度矩阵，选择最大匹配组合，计算语义相关度。最终得出短文本结构与语义相结合的相关度的计算公式：

$$Sim(A, B) = \alpha \times Sim_{si}(A, B) + \beta \times Sim_{inv}(A, B) + \gamma \times Sim_{word}(A, B)$$

其中：经实验，当 25.0，15.0，6.0 时结果最优。

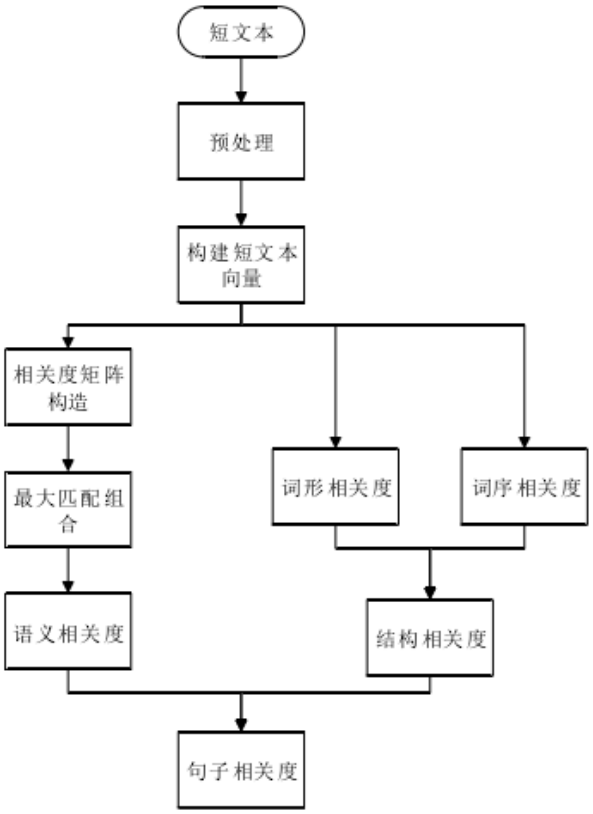


图 4.3 短文本处理相似度流程图

4.4 朴素贝叶斯的多项式模型（MultinomialNB）

多项式模型是朴素贝叶斯最常用的三种模型中的一种，常用于文本分类，特征是单词，值是单词出现的次数

当特征是离散的时候，使用多项式模型。多项式模型在计算先验概率$P_{(yk)}$和条件概率$p_{(xi|yk)}$时，会做一些平滑处理，具体公式为：

$$P_{(yk)} = \frac{N_{yk} + \alpha}{N + k\alpha}$$

N 是总的样本个数，k 是总的类别个数，N_{yk}是类别为 yk 的样本个数，α是平滑值

$$P_{(x_i|y_k)} = \frac{N_{y_k,x_i} + \alpha}{N_{yk} + n\alpha}$$

N_{yk}是类别为 yk 的样本个数，n 是特征的维数，$N_{yk,\mathrm{xi}}$是类别为 yk 的样本中，第 i 维特征的值是 xi 的样本个数，α是平滑值。当α=1 时，称作 Laplace 平滑，当 0<α<1 时，称作 Lidstone 平滑，α=0 时不做平滑。

5. 模型构建

5.1 文本分类

对附件 2 的留言详情划分训练集 70%和测试集 30%，再用列表推导式提高转换效率，分别进行训练集和测试集的分词，通过训练集得出词向量权重表，词向量权重表通过多项式模型进行分类，将分词以后的测试集转换到词向量权重表，通过训练好的模型得出训练结果，通过准确率和召回率算出 F1-Score。

$$\mathrm{F_1{-}Score}: \quad F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i+R_i}$$

$$\text{准确率}: \ \mathrm{P} = \frac{TP}{TP+FP}$$

$$\text{召回率}: \ \mathrm{R} = \frac{TP}{TP+FN}$$

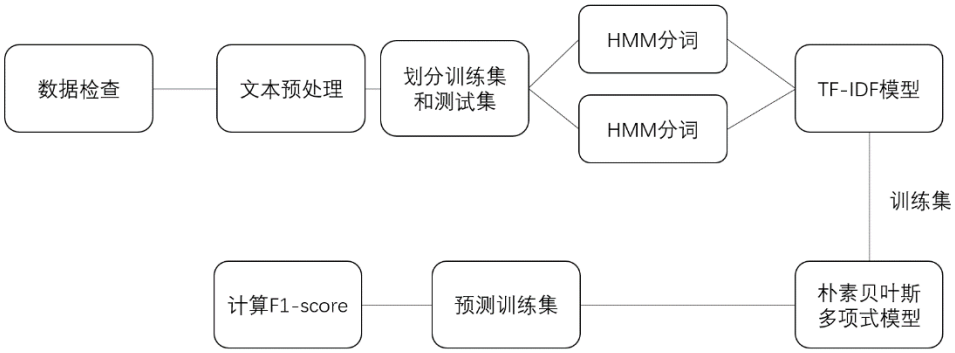


图 5.1 问题 1 的文本分类的流程图

5.2 文本聚类

用 TF-IDF 词向量模型，通过 K-means 将句子分类，效果不理想。用 Word2vec 里的连续词袋（CBOW）模型得到词与词的相似度，然后进行归一化，用 K-means 进行聚类，并得出轮廓系数，考虑到类别数，从 6-20 选取适当的轮廓系数，经过大量反复测试，综合考虑最终确定簇数为 10。统计每条文本不同类别出现的次数，出现最多次数的类别为该文本的类别，提取留言主题的地址信息和人群信息，将其合并为新字段“地址|人群”，按地址人群和类别分组计算留言的次数、最近和最远留言时间以及点赞数和反对数，得出以下指标：“留言次数”“留言频率”“点赞率”。以三者为构成因素计算不同地址、人群、不同类别主题的热度指标，依据热度指标进行排序，得出前五的热点问题。最终将三个指标进行整合处理，确定热点指数，参数满分为 100。

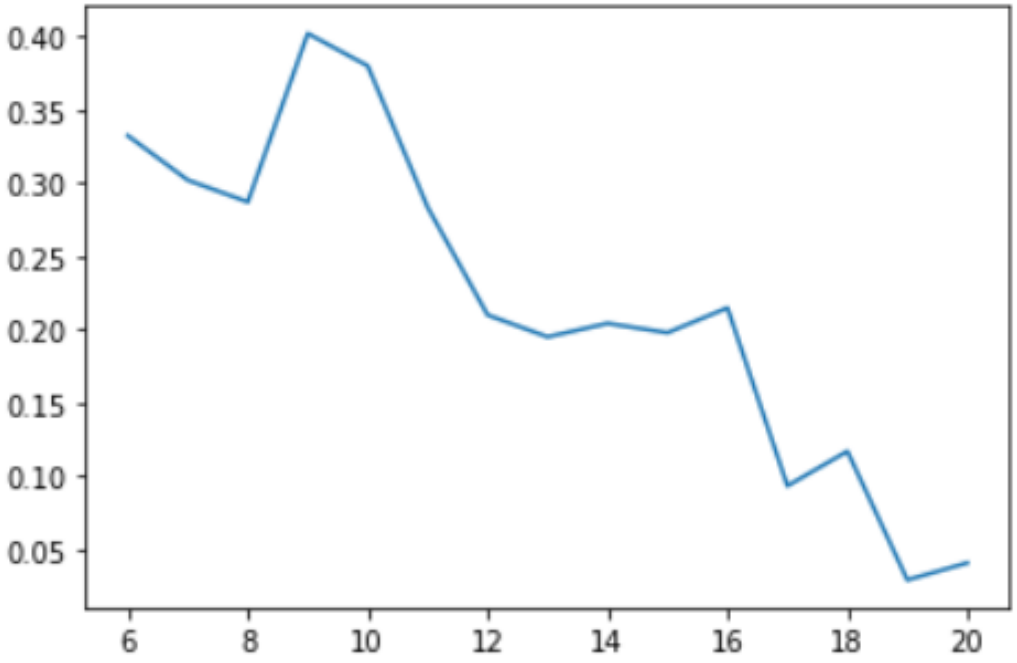
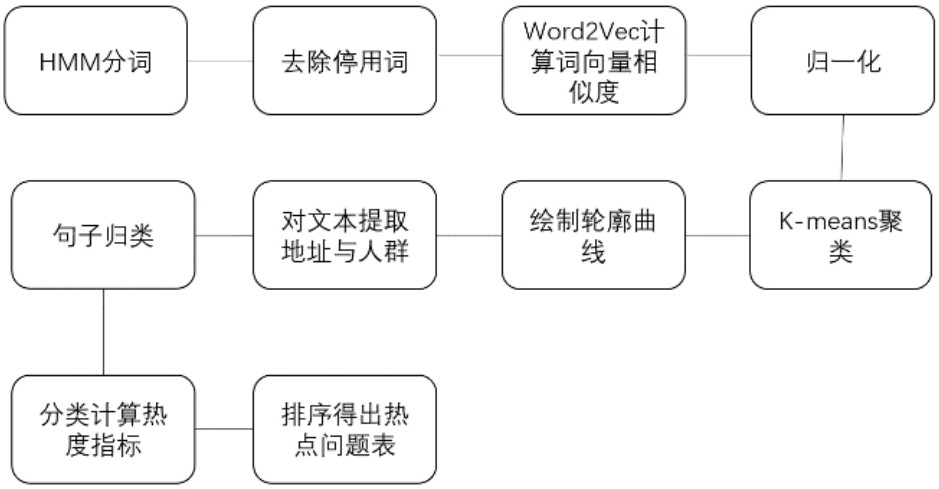


图 5.2.1 轮廓曲线



文本聚类

图 5.2.2 问题 2 的文本聚类流程图

5.3 答复评价

首先观察到附件四的答复意见有重复描述收到答复的前缀，所以将其去掉。如图 5.3

现将网友在平台《问政西地省》栏目向胡华衡书记留言反映“A2区景蓉花苑物业管理调查核实情况向该网友答复如下：您好，首先感谢您对我们工作的信任和支持，关于目给胡华衡书记留言，反映“A2区景蓉花苑物业管理有问题”的情况已收悉。现将我情况答复如下：经调查了解，针对来信所反映的“小区停车收费问题”，景蓉华苑业年4月10日至4月27日以“意见收集方式”召开了业主大会，经业委会统计，超过三分同意收取停车管理费，在业主大会结束后业委会也对业主提出的意见和建议进行了认

图 5.3.1 留言答复的前缀语句

这一部分的模型根据质量指标分为两个部分，质量指标 1：文本相关度；质量指标 2：答复效率。

5.3.1 文本相关度

文本相关度包括结构相关度和句子相关度，结构相关度又包括词形相关度、

词序相关度；句子相关度即语义相关度

词形相关度公式：

$$\frac{\text{两个短文本中相同词语出现的次数}}{\text{第一个短文本的词语个数} + \text{第二个短文本的词语个数} - \text{两个短文本中相同词语出现的次数}}$$

词序相关度和语义相关度公式在前文具体的模型介绍里都有具体列举以及解释，这里不再赘述。

Synonyms 库可以用于自然语言理解的很多任务：文本对齐，推荐算法，相似度计算，语义偏移，关键字提取，概念提取，自动摘要，搜索引擎等。这里我们将词义用 synonyms 库进行相似度计算，经过大量重复训练计算，综合考虑，最终赋予权重为：25%词形相似度 15%词序相似度 60%语义相似度=句子相关度，作为评价的质量指标 1。

5.3.2 留言答复效率

将留言时间和留言答复用 datetime 库将文本型时间转为 datetime 类型，新增“相隔时间”字段：（答复时间-留言时间）。将相隔时间归一化，新增“时间大小”字段：(相隔时间-时间间隔中最小值)/(相隔时间最大值-相隔时间最小值)。经过大量反复测试，综合考虑最终确定权重分布：90%相似度，10%时间。

最终将两个指标综合训练，得出答复评价模型，参数满分为 1。

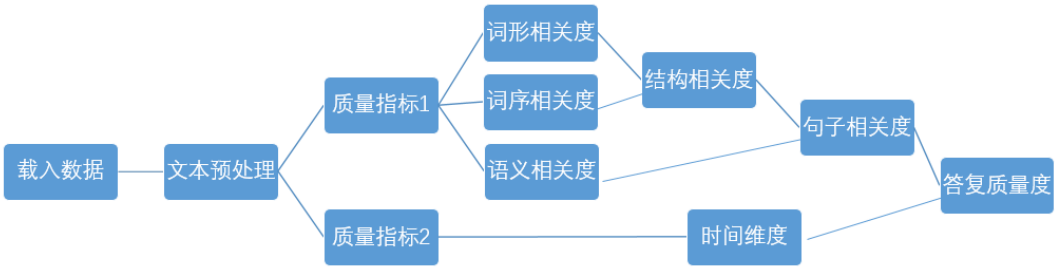


图 5.3.2 句子相关度计算模型流程图

6.模型评价与改进

6.1 TF-IDF 算法

优点：简单快速，结果比较符合实际情况。

缺点：单纯以"词频"衡量一个词的重要性，不够全面，有时重要的词可能出现次数并不多。而且，这种算法无法体现词的位置信息，出现位置靠前的词与出现位置靠后的词，都被视为重要性相同，这是不正确的。

改进：对全文的第一段和每一段的第一句话，给予较大的权重。

6.2 朴素贝叶斯的多项式模型

优点：（1）这个模型属于生成式模型，通过计算概率来进行分类，可以用来处理多分类问题；（2）对小规模的数据表现很好，适合多分类任务，适合增量式训练，算法也比较简单

缺点：（1）需要计算先验概率；（2）对输入数据的表达形式很敏感；（3）由于朴素贝叶斯的“朴素”特点，所以会带一些准确率上的损失

6.3 计算文本聚类

在做文本聚类时，主要精力都放在了考虑词的相似度上，从词聚类到句子聚类的方法有些粗糙，模型效果、精度等并不是特别好，并且忽略了词序的重要性。若使用长短期记忆人工神经网络（lstm）捕捉词句子信息，结合发挥的效果会更加具有优势。

6.4 句子相关度计算模型

首先，要算出向量词对相关度矩阵，将两个文本分词成两个向量，通过两个向量的笛卡尔积，计算二者相似度，但是由于笛卡尔积的公式性质等导致计算量过大，所以改用 synonyms 库代替计算，效果虽然不够精确，但是简单、高效、便捷。

7. 总结

通过这次比赛能够深有体会地总结得到文本挖掘的内在含义，通过使用了几种常见的文本挖掘模型来对 C 题的留言主题、留言内容、答复等数据进行分析与探究。在文本挖掘中最重要最基本的应用是实现文本的分类和聚类，前者是有监督的挖掘算法，后者是无监督的挖掘算法。文本挖掘它又是一个多学科混杂的领域，涵盖了多种技术，包括数据挖掘技术、信息抽取、信息检索，机器学习、自然语言处理、计算语言学、统计数据分析、线性几何、概率理论甚至一些从未接触过的领域。同时也深刻体会大数据时代已经来临，各行各业都在使用、分析以及挖掘数据，即使是复杂的自然语言，计算机也能转换成可以识别的编码来进行计算、分类、聚类等等。我们也能认识到自身所涉猎构建的模型仍有巨大的提升空间，因此我们也更加需要磨练、扎实自己数据分析的基本功。

8. 参考文献

[1] 这孩子谁懂哈.TF-IDF 算法详解.[DB/OL].
https://blog.csdn.net/zhaomengszu/java/article/details/81452907
[2] 大饼博士 X. word2vec 算法原理：跳字模型（skip-gram）和连续词袋模型（CBOW）.[DB/OL].
https://blog.csdn.net/xbinworld/java/article/details/90416529
[3] Qmei 在学习. synonmys 处理文本： 词语句子相似度比较.[DB/OL].
https://blog.csdn.net/weixin_41824534/article/details/99004566
[4] 风弦鹤. Word2vec 原理及其 Python 实现.[DB/OL].
https://blog.csdn.net/huacha__/java/article/details/84068653
[5] 荆琪.基于维基百科的短文本相关度计算.[D].山西:太原理工大学,2017,06
[6] Distributed Representations of Sentences and Documents
[7] Efficient estimation of word representations in vector space