“智慧政务”中的文本挖掘与应用

摘要

近年来，随着互联网的广泛应用和网络的不断创新发展，各种网络问政平台如雨后春笋。在这个大数据时代，网络上的各种信息数据杂乱无章，这对于我们划分整理信息的相关部门也是转型性的挑战。大数据、AR 等技术蓬勃发展促使政府的智慧政务系统，必须通过创新与改变来提高施政效率与管理水平。本文将通过从互联网收集来的群众问政留言记录与相关部门的答复意见进行文本信息的挖掘与分析。

针对问题 1，首先我们需要读训练样本进行去重的处理，防止重复留言等干扰性留言。接着我们利用 Hanlp 的命名实体识别功能，对我们的文本进行名词提取，并将所获得的地名加入 jieba 分词的词库中，以提高模型的准确率。进而我们将分好的词向量化，利用 TF-IDF 对每篇文本转化成权重向量，再设计三层的 BP 神经网络，将我们的词向量作为输入，留言类别作为输出。最后对 BP 神经网络进行训练，得到“智慧财政”的文本分类器。该分类器可以对民众留言进行快速分类。模型的准确性检验在测试集中的准确率为 90.61%，F1-score 为 0.8977。

针对问题 2，首先重复问题 1 的步骤对数据进行去重和分词并且利用 TF-IDF 转换成数值型数据，然后我们利用 DBSCAN 算法，将我们的文本向量进行无监督聚类处理，找到同类留言中数目最多 105 类。同时结合每个类别的支持数和反对数，并利用线性权重法得到我们的综合评价指标，最后得到我们排名前 5 的热点问题。

针对问题 3, 由于要对相关部门的答复意见的质量给出一套评价方案，我们从答复的相关性、完整性、可解释性这三个角度进行分析并建立评价模型。首先重复问题 1 的步骤对留言和回复进行去重和分词并且利用词袋模型转化成数值型数据，采用相关概率评分和关键词赋权加和法。然后对三个性质的数值进行归一化处理。最后按 4:2:4 进行加权求和处理到我们最终评分。

关键词:BP 神经网络,DBSCAN 算法, 命名实体识别,Word2Vec 模型, 相关概率评分.

Summary

Recent years, with the wide application of the Internet and continuous innovation and development of network, a variety of platforms of Network governance are springing up. However, the various information and data in this Big Data Era on the Internet is disordered, which is a transformational challenge for relevant departments to systematize and sort out the information. Using rapidly developing technologies of Big Data and AR, Intelligent Government System can be renovated so as to improve the efficiency and governing ability. In this paper, we will do data-mining and analyze the text information through records of people's political messages collected from the Internet and replies of relevant departments.

Problem 1. We need to de-duplicate training samples to prevent repeated messages and other disturbing messages at first. Then we use HANLP's named entity recognition function to extract nouns from the text, and add the obtained place names into the vocabulary of JIEBA segmentation to improve accuracy of the model. Next, we vectorize and use TF-IDF to transform each text into weight vector, and then design a three-layer BP neural network, taking our word vector as input and message category as output. Finally, BP neural network is trained to get the text classifier of "intelligent finance". This classifier can quickly classify public messages. The accuracy of the model in test set was 90.61%, and the F1 score was 0.8977.

Problem 2. We first repeat steps of the solutions to Problem 1 to de-duplicate and segment the data, and use TF-IDF to convert it into numerical data. Then we use DBSCAN algorithm to do unsupervised clustering of our text vectors, and find the maximum number of 105 kinds of similar messages. At the same time, we combine the support number and objection number of each category, and use the linear weight method to get our comprehensive evaluation index. And, we get the top 5 hot issues.

Problem 3. We analyze the "relevance", "integrity" and "interpretability" of replies. First, we repeat tsteps of solutions to Problem 1 to de-duplicate and segment the messages and replies, and then use the word bag model to transform them into numerical data, using correlation probability score and keyword weighted sum method. Then we normalize values of the three properties. Last, the weighted sum processing is carried out based on a 4:2:4 ratio until our final score is obtained.

keyword: BP neural network, DBSCAN algorithm, named entity recognition, Word2Vec model, correlation probability score

目录

1 挖掘目标

本次建模的目标是通过用户对相关问题的留言、一级标题和相关部门的相应答复评论，利用 jieba 中文分词等工具对用户留言和相关答复进行分词处理。然后用 TF-IDF 和词袋模型对文本进行数值化处理，然后利用 BP 神经网络算法、DBSCAN 模型和 Word2Vec 模型达到以下三个目标：

1. 利用文本分词和文本特征提取的方法对文本数据进行数值化处理，再结合附件所给的一级标题对数据进行分组处理，然后将训练数据带入模型中调整我们的内部参数，最后对我们的测试集数据进行分类验证模型的准确率.

2. 根据用户留言情况，使用 DBSCAN 模型对用户留言进行无监督据类处理，然后结合支持数和反对数，利用线性权重加权得到综合评价指标，最后取排名前五的作为热点问题。

3. 根据用户留言和相关部门的答复，分别从三种性质对答发的字段进行数值化处理，然后赋权求和得到综合评分。

4. 总流程图如下

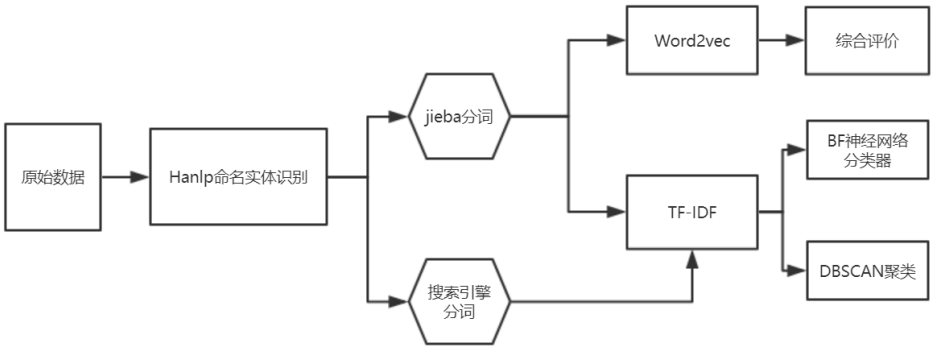

图 1: 总流程图

2 分析方法与过程

本文主要包含以下步骤:

1. 步骤一、数据的预处理。由于题目中所给出的数据存在部分相同的留言，所以我们首先对原始数据进行去重处理，接着在此基础上进行 jieba 分词。在分完词后，我们把一些没有意义的单词进行删除，最后利用 TF-IDF 对我们词语进行数值化处理。

2. 步骤二、把我们的词向量输入我们设计的三层 BP 神经网络，调整调整我们的网络参数，得到我们的文本分类器。

3. 步骤三、利用 DBSCAN 算法对文本向量进行无监督聚类处理，以得到同类留言中数目最多 105 类。最后结合支持数和反对数，通过线性权重法得到我们的综合评价指标，并选取排名前 5 的作为热点问题。

4. 步骤四、在回复中进行关键词匹配法，将我们认为完整性的关键词进行匹配，算出我们的完整性评价值。然后利用相关概率评分对回复和留言进行比对得到我们的相关性和可解释性评价值。

3 问题 1 分析方法与过程

3.1 问题 1 的流程图

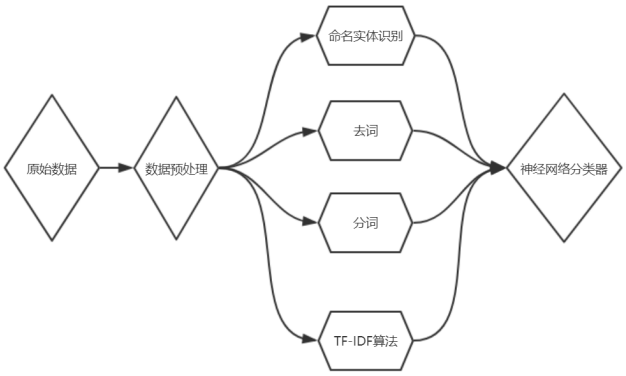

图 2: 问题 1 的流程图

3.2 数据预处理

附录所给我们的是一个巨大且又大量冗余的文本数据，例如：”李书记您好，感谢您的阅读。十二五期间…”显然这些繁琐的文本内容并不便我们从文本内容中提取出关键信息。并且，文本留言数据里面也存在大量价值含量低且无实际意义的信息。如果将这些无用的信息也引入分词中，必然会对词频统计或者感情分析造成很大的影响。考虑到前面的修饰词语和重复留言对文本分类或者文本聚类带来的巨大的影响，我们需要先对数据进行预处理。

数据预处理具体步骤：

1. 命名实体识别在分词中可能出现分词不准确，而分词的结果的准确性又会对后面结果造成误差，从而影响到民众留言的分类，在民众留言中出现了多个地名和人名字如: 反映 A7 县春华镇金鼎村水泥路、自来水到户的问题, 如果直接用 jieba 分词会出现下列的情况:

   反映/A7/县/春华/镇金鼎村/水泥路/、/自来水/到户/的/问题

   这类分词明显是有错误的，地点分词不准确，可能对后面分类问题造成不小的影响，因此我们使用 Hanlp 对文本主题进行命名实体识别，将文本中地名汇集起来，我们将所选好的地名保存为 didian.txt 中。

2. 分词在对民众留言信息分析之前，首先要把非结构化的文本数据转化为计算机能够识别的结构化系统。在附件 2 中民众留言，以中文文本的方式给出的数据。为了便于转化，先要对民众留言进行中文分词处理。

   本文采用的是强化 jieba 的方法，将命名实体识别中的地点添加到 jieba 分词中从而提高分词的准确性:

3. 去词

   由于文本留言数据里面存在大量价值含量比较低，甚至没有价值的数据，比如标点符号，或者你，我，他，语气词，结尾词等一系列修饰词，这些东西的引入必将会让我们的数据分析造成很大的影响，得到的结果也比将存在问题，那么选词之后就必须将这些由影响，价值低的数据删除。

   本文使用使用 python3 中的 jieba 工具先将文本切分成很多个词组。然后我们找到一个 stopwords.txt 来存放一些价值含量低，表意不清，标点等，由此得到了我们的词列表。

4. TF-IDF

   得到我们的单词列表之后还，但是电脑并不能识别这些字符串是干什么的，有什么含义，因为计算机只能识别一些数字，所以我们还需要对字符串进行数学化处理，就文本数据转化成计算机能够数据的数字，同理计算机也不能识别类别数据，所以也需要对我们的各种分类类型进行数学化处理。

   这里我们采用 TF-IDF 算法，把民众留言转化成权重向量. 具体算法如下：

   第一步：计算词频，既 TF 权重 (Term Frequency)

$$\text{词频}(TF) = \text{文本中某个词出现的个数} \tag{1}$$

   考虑到不同文章长度可能不一样，为了便于文章的比较，所以对“词频”进行标准化处理：

$$\text{词频}(TF) = \frac{\text{文本中某个词出现的个数}}{\text{文本的总词数}} \tag{2}$$

   第二步：计算 IDF 权重，既逆文档率 (IDF), 需要建立一个语料库 (corpus) , 主要是用来模拟语言环境的，IDF 越大，表示此特征在分本中分布越集中，说明该词在区分该分布内容属性的能力强。

$$\text{逆文档频率}(IDF) = \log(\frac{\text{语料库的文本总数}}{\text{包含该词的文本数 +1}}) \tag{3}$$

   第三步：计算 TF-IDF 的值

$$TF - IDF = \text{词频}(TF) \times \text{逆文档频率}(IDF) \tag{4}$$

这样就通过 TF-IDF 将我们的文本数据转化乘带有权重的数值型数据。

3.3 BP 神经网络分类器

经过上面步骤的处理我们已经得到了计算机可以识别的数据，但是这个时候文本数据类型分布比较集中，相同的类型排列在一起，这样切分训练集和测试集的时候就必然会造成误差，所以我们将我们的所得到的文本数据和类别数据同时进行离散化处理，然后将他们切分成训练集和测试集合两类。

切分好数据数据之后我们运用的神经网络 BP 算法

1. BP 神经网络算法原理

   BP 神经网络又叫多层感知器，有输入层，一层或者多层隐含层及输出层构成，学习过程由信息的正向传播然后计算出误差，再由误差的反向传播，来修改模型里面的参数，采用梯度最速下降法不断对数据里面的参数权重等目的是为了达到误差期望最小. 可帮助研究人员研究简化工作，本文要求对各种留言进行，一级标题的分类，故可以利用神经网络算法来建立彼此之间的关联。其结构如下图所示：

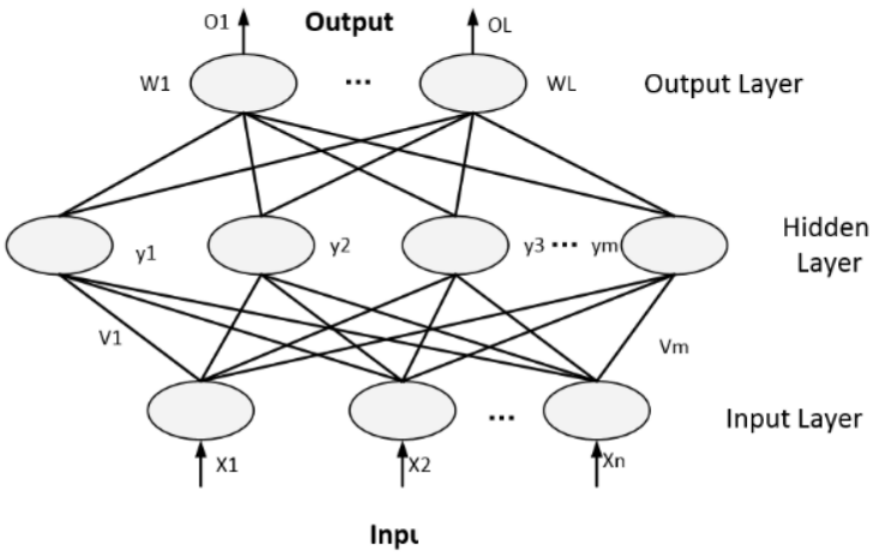

图 3: 结构图

2. 前向传播我们把每个文本的向量 $(x_1, x_2, x_3....)$ 作为输入，隐藏层 $(y_1, y_2, y_3...)$，最后的 (w_1, w_2, w_3) 作为输出层。其中第 h 个隐层神经元的输入是:

$$a_h = \sum_{i=1}^{n} v_{ih} x_i$$

其中 v 是输入层到隐藏层的权值矩阵，第 h 个隐层神经元的输出是:

$$y_h = f(a_h - \gamma_h)$$

其中 γ 是偏正项，f 是激活函数，同理第 j 个输出神经元的输入是:

$$\beta_j = \sum_{h=1}^{m} w_{hj} y_h$$

同理第 j 个输出层神经元的输出是:

$$\hat{o_j} = f(\beta_j - \theta)$$

其中 θ 也是偏正项目，那么我们要优化的函数为

$$E = \frac{1}{2} \sum_{j=1}^{L} (\hat{o_j} - o_j)^2$$

其中输入层到隐藏层的输出函数为 $f(x) = sigmoid(x)$ 隐藏层到输出层的输出函数为：$f(x) = softmax(x)$

3. 反向传播:

反向传播主要是对误差进行梯度下降算处理使得函数 E 尽可能的小，则我们说分类效果好。为了求 E 的最小值点优化我们的两个权值矩阵因此我们对方程 E 进行求导，利用梯度下降算法求出我们的各个变量的变化情况：

$$\Delta w_h j = \eta \hat{o_j}(1 - \hat{o_j})(o_j - \hat{o_j}) y_m$$

$$\Delta \theta_j = -\eta \hat{o_j}(1 - \hat{o_j})(o_j - \hat{o_j})$$

$$\Delta v_i h = \eta y_h (1 - y_h) \sum_{j=1}^{l} w_{hj} \hat{o_j}(1 - \hat{o_j})(o_j - \hat{o_j}) x_i$$

$$\Delta v_i h = -\eta y_h (1 - y_h) \sum_{j=1}^{l} w_{hj} \hat{o_j}(1 - \hat{o_j})(o_j - \hat{o_j})$$

然后逐步循环迭代，优化我们的分类器。

4. 它的思想是利用逐层贪婪训练的方法，把原来多层的的神经网络分解成一个个小网络，每次训练一小部分，然后将钱一层的输出作为后一层的输入，最后连接一个分类器。然后这样就可以得到一个神经网络的权重，然后我们对该权重进行反向传播来修改我们的数据，经过多次循环迭代，最终得到一个分率准确率比较好的模型。

在 python3 种利用已经开发好的工具包，用上面这个思想对我们的模型进行训练，将训练。然后将我们的测试数据带入检验模型的准确率。和没每个指标查全率，然后利用 F-Score 算法对分类方法进行评价。F-Score 分类评分计算如下：

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2P_iR_i}{P_i + R_i}$$

4 问题 2 分析方法与过程

4.1 问题 2 流程图

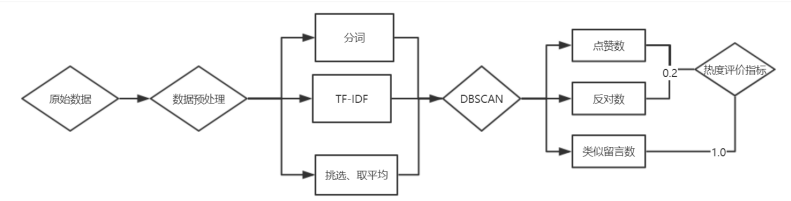

图 4: 问题 2 流程图

此问题的分析流程大致分为以下三步：

1、数据预处理。首先使用分词对文本的进行数据化处理；然后为了减少计算量，我们使用 TF-IDF 算法进行向量化；最后考虑到数据的真实性与有效性，我们对数据进一步地挑选取平均。

2、使用 DBSCAN 进行无监督聚类。

3、找出热点问题。充分使用点赞数与反对数，将其求和后将得到的参数与类似文本在文中出现的次数进行加权求和。对求和值进一步数值化得到热度指标。

4.2 数据预处理

1. 分词处理

   首先对文本进行数据化处理，在 jieba 中加入命名实体识别的地名还是不够的。比如”魅力之城小区”有些留言可能会写”魅力小区”或者”魅力之城”，虽然字符串不同，但是他们所指的地方都是同一个，如果同第一问的分词方法肯定会造成比较大的误差，为了减少这种误差我们在进行 jieba 分词的时候采用搜索引擎分词法，将数据尽可能的细分，以提高相同文本的相似度，进而加强分词准确性。

2. TF-IDF 同第一问的使用方法相同，将所分好的词进行取停用词处理，确保得到有效的数据，从而减少没必要的计算量。然后将我们分的好词进行 TF-IDF 向量化处理得到我们的文本向量权重矩阵。

3. 挑选、取平均在附录所给的数据中，存在个别点赞数或者反对数奇高的情况，我们认为这是不太真实的。为了避免可能存在的刷赞等行为。我们将点赞数目和反对数之和大于 100 的部分取百分之一，在和原来的 100 相加，最后赋权取值。

4.3 DBSCAN

要想在众多文本中文本中找到相同一地点，相同的事件，则说明两篇文本中具有比较高的相似度。转化成数学问题则表示，两个向量之间的距离很接近, 所以我们想到用无监督学习的聚类方法。

DBSCAN 算法又称为僵尸感染模型主要是通过下面 4 个步骤，将样本点进行无监督学习的分类：

第一步：检测数据中是否存在为分类的点 p，如果 p 被处理 (认为属于每一个簇或者标准为噪声点)，则坚持其领域，若对象不小于我们所给的距离最小值点，则建立新的簇，将其中所有点加入候选集中 N。

第二步：对候选集 N 中所有未被处理的对象 q，检查其领域，若至少包含 minpts 个对象，则将这些对象加入到 N，若 q 未加入任何一个簇，则将 q 加入 c 。

第三步: 重复步骤 2，继续坚持 N 中未处理的对象，直到候选集 N 为空。

第四步：重复步骤 1-3，直到所有的点都归入某一个簇或者标记为噪声点。

4.4 线性权重法

通过上诉步骤我们将文本数据聚成了多类，然后单单靠数量每种类别的数量来评论我们觉得比较欠妥，本文采用的是类别和点赞数和反对数共同决定的我们认为不管是点赞还是反对都是关注这个问题所以才去评论的，所以我们认为点赞和反对是等价的，我们将点赞数和反对数相加，然后主观的赋予不同的权重进行线性权重法，计算得到我们的热度指标。

5 问题 3 分析方法与过程

5.1 流程图

计划流程图如下：

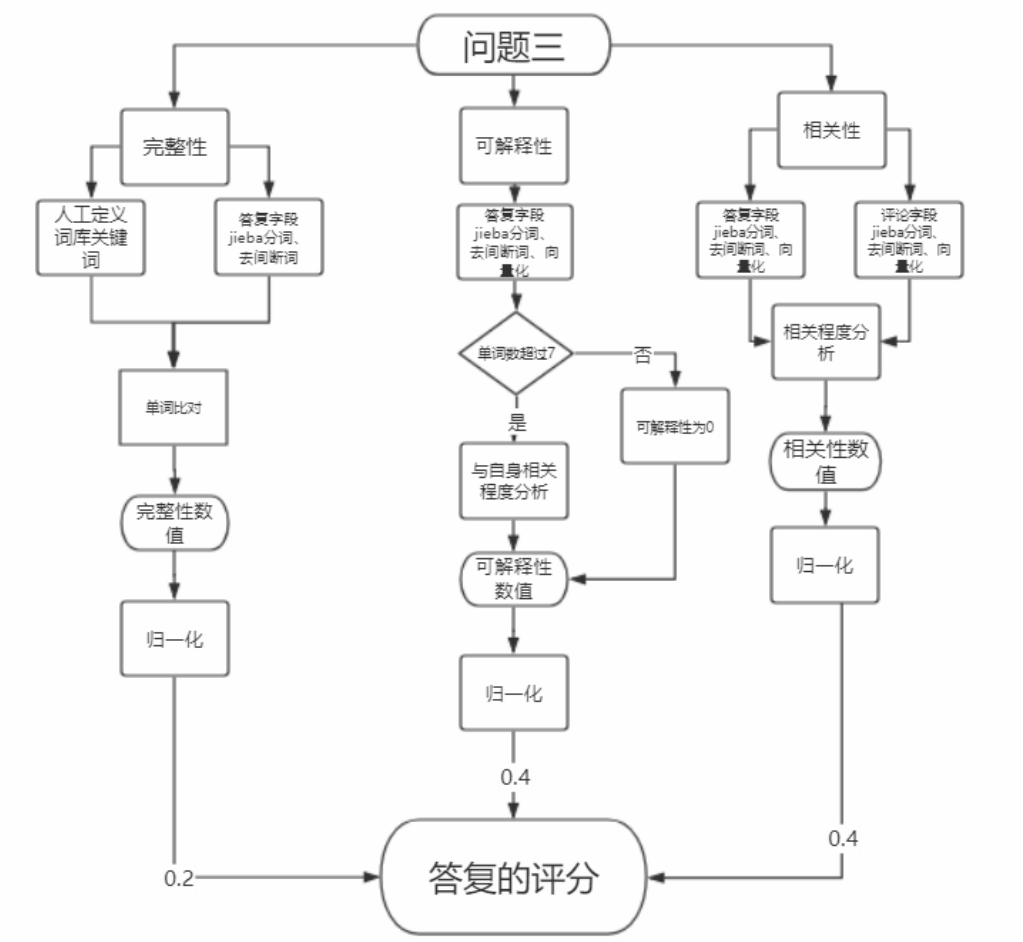

图 5: 问题三总流程图

此问题的分析流程大致分为以下三步：

1. 完整性。首先，我们人工从留言的开头、中间和结尾分别定义出词库的关键词，再把定义出的关键词放入 A 库。同时，我们对答复字段进行 jieba 分词和去关键词两类操作，把数据处理后各字段的单词放入 B 库。接着，把 A、B 两库的单词进行对比得到完整性数值。最后进行归一化处理；

2. 可解释性。首先对答复字段进行 jieba 分词、去间断和向量化三类操作，然后把数据处理后各字段的单词放入 B 库。接着我们把单词数与 7 对比，若超过则与自身相关程度分析，否则判定其解释性为 0. 得到可解释性进行归一化；

3. 相关性。分别对评论字段和答复字段进行 jieba 分词、去间断词和向量化三类操作，然后分别把数据处理后的单词放入 A 库和 B 库。最后对两库进行相关程度分析，得到相关性数值，同样对数值进行归一化处理；

4. 完成三种性质的分析后，对三种性质分析后的数值加权平均得到数字集 T，然后对 T 带入 (T*5+1) 进行运算，结果就近取整，得到的数值当作此答复的星级。

5.2 数据预处理

同理我们将附件四的数据中，留言详情、答复意见这两栏中的文本进行 jieba 分词，并去除间断词和空格符保存在一个列表，然后将每个词按照上面的步骤进行词袋模型向量化处理，但是只有向量化却不能准确的反应出单词和单词之间的关系，这对我们的答复评价起不了太大作用，这时我们引入 Word2vec 他可以反应两个单词之间的关系，然后我们将它生成一个字典。这个字典将用于未来的相关性和可解释性的文本转化数字指标的分析。

1. 我们训练得到了我们的词向量。为了将文本感情分析转化成机器学习的问题，我们需要先将单词进行数值化。将单词数值化的方法在 python 中常见的 one-hot 方法, 其中心思想就是将词语映射成不同词总数维的向量，比如: 市工商局 [1 0 0 ....] 等，这样就把我们的单词数量化了。

   但是，这样也存在一个问题，词向量不能很好地表示词语之间联系，也就是两个词之间存在的相互关系很难从上述词向量中反应出来，从而无法反应语义的关联性。

   所以我们选用了 Distributed Represcentation 这一方法来反应关联性。因为这种方法能翻译存储词与词之间的距离远近，而且使用到 Distributed Representation 表示的向量可以称为词向量。例如领导可以表示为 [0.03665977 -0.032901917 -0.065880395 -0.026848406 -0.03411947....] ，这样两个词意相近的词语距离会比较相近，词义关联不大的距离会相隔较远。

   一般而言，不同的训练方法或者语料库训练得到的词向量是不一样的，所训练出来的维度也会有所不同。所以在这里我们采用 word2Vec 算法来解决这一问题。

2. Word2Vec 模型简介

   Word2Vec 是 Tomas Mikolov 领导的团队研究出的一种算法，也是前几年比较火的一种让文本中单词向量化的方法，它能通过神经网络机器学习算法来训练 N-gram 语言模型，并且在训练中得到词向量。

Word2Vec 采用神经网络语言模型具体涉及到 Skip-gram 模型和 CBOW 模型是他们混合得到得到的产物。具体模型如下：

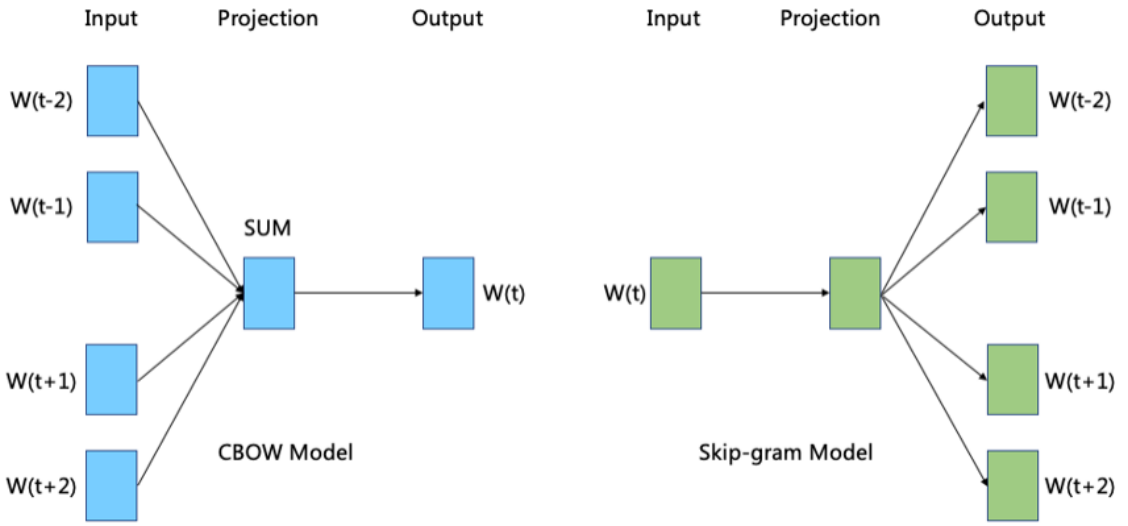

图 6: Skip-gram 和 CBOW

其中 $\omega(t)$ 表示当前单词，$\omega(t-1),\omega(t-2)$ 表示前面 1，2 个单词，$\omega(t+1),\omega(t+2)$ 表示后两个单词，CBOW 是使用行下文去预测目标单词来训练词向量，Skip-gram 是使用目标词来预测周围词的训练模型。

而 Skip-gram 的目标函数如下：

$$l(\theta) = arg \max_{\theta} \prod_{\omega \in Test} \prod_{c \in Context(\omega)} P(c|\omega;\theta)$$

取对数之后形式为:

$$L(\theta) = arg \max_{\theta} \prod_{\omega \in Test} \prod_{c \in Context(\omega)} \log P(c|\omega;\theta)$$

其中 w：要预测的下一个词，然后它的维度是 (D, 1)，D 指的就是词汇表的大小，也就是词汇表中一共含有多少个不一样的单词。

$context(\theta)$ 表示的是上下文，维度 ((n-1)m,1),n-1, 是指在当前 text 中单词数据减一。m 表示每个单词的维度，这里我们设置的是 50 维。

text 表示训练的文本.

其中 $\theta = [\mu, v]$,μ 表示单词作为上下文的当初向量，v 代表了单词作为中心词的词向量

w,c 在上下文出现，所以他们更有可能相识，然后我们就需要把 $P(c|w;\theta)$ 越大越好。通过处理我们就能得到下面这个概率:

$$P(c|\omega;\theta)=\frac{e^{\mu_c*v_\omega}}{\sum_{c'\in corpus}e^{\mu'_c*v_\omega}}$$

同理运行逻辑回归的思路，Skin-Gram 可以得到如下式子:

$$l(\theta)=arg\max_{\theta}\prod_{(c,w)\in D}P(D=1|c,w;\theta)\prod_{(c,w)\in\tilde{D}}P(D=0|c,w;\theta)$$

$$\begin{aligned}L(\theta)&=arg\max_{\theta}[\sum_{(c,w)\in D}P(D=1|c,w;\theta)+\sum_{(c,w)\in\tilde{D}}P(D=0|c,w;\theta)]\\&=arg\max_{\theta}[\sum_{(c,w)\in D}log\sigma(u_c\cdot v_w)+\sum_{(c,w)\in\tilde{D}}log\sigma(-u_c\cdot v_w)]\end{aligned}$$

然后我们对各个未知数求一个偏导:

$$\frac{\partial L(\theta)}{\partial_{\mu_c}}=[1-\sigma(\mu_c\cdot v_\omega)]\cdot v_\omega$$

$$\frac{\partial L(\theta)}{\partial_{\mu'_c}}=[\sigma(-\mu_{c'}\cdot v_\omega)-1]\cdot v_\omega$$

$$\frac{\partial L(\theta)}{\partial_{v_\omega}}=[1-\sigma(\mu_c\cdot v_\omega)]\cdot\mu_c+\sum_{c'\in N(\omega)}[\sigma(-\mu_{c'}\cdot v_\omega)-1]\cdot\mu'_c$$

得到偏导之后，我们就能够采用梯度下降法取更新我们的参数:

$$\mu_c=\mu_c-\alpha\frac{\partial L(\theta)}{\partial_{\mu_c}}$$

$$\mu'_c=\mu'_c-\alpha\cdot\partial g(x)\partial L(\theta)\partial\mu'_c$$

$$\mu_\omega=\mu_\omega-\alpha\frac{\partial L(\theta)}{\partial_{\mu_w}}$$

然后呢为了让训练样本和测试样本数据更加的均匀，不会出现一下某些地方类别过于集中，这样会对我们的训练样本和测试样本造成不小的误差，所以我们将上诉所得到的数据进行乱序处理，来达到让我们的数据分布更加均匀，来减少数据误差。

我们利用上诉算法方法加上 python3.7 软件实现，将留言中各个词向量转化成 50 维的矩阵，得到了数值化后的矩阵，保存在文件夹下面的 word.text 中。

5.3 相关性

我们认为相关性就是留言详情与答复意见的联系。

例如：

留言详情：”请问，A 市有退休工资的残疾人士能够领取护理补贴吗？”

答复意见：”你好。如果你是重度残疾人，是可以领取护理补贴的。”

因为它的答复意见是围绕着留言详情的，所以我们认为他们的相关性是比较强的。

因为相关性体现了留言详情和答复意见的逻辑契合度，而逻辑契合度可以看成两个文本出现在一起的相关概率，又因为两个文本间词语相关概率可以体现文本的相关概率。

因此我们将留言的词语放入 A 库中，在将答复意见的词语放入 B 库，然后我们对留言详情与与之对应的答复意见中的每一个单词分别用 Word2vec 建立的字典进行相关性分析，计算并筛选出相关性大于等于 0.4 的单词组的数目。将单词组的相关性数据求均值，由此得到这一答复的相关性指标。为了去量纲，我们将全部相关性指标归一化处理。得到最终相关性指标。归一化处理公式如下：

$$x = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5}$$

相关性流程图：

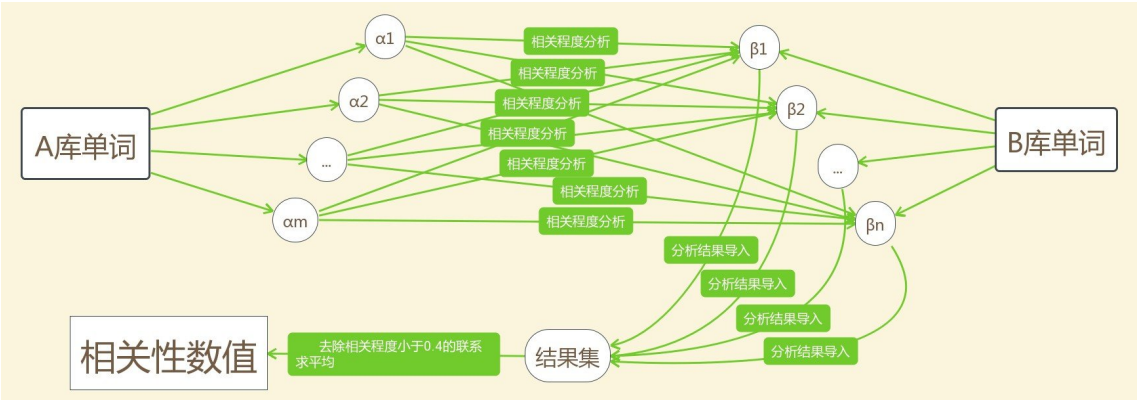

图 7: 相关性流程图

5.4 完整性

我们认为完整性就是答复的格式是否规范的体现。格式规范例如：开头称呼您（你）好、内容中表示收到留言、结尾具备日期等。

例如：”网友：您好！您的留言已收悉，现回复如下：由于您已买房且办理了提取住房公积金用于归还住房按揭贷款，故不能再办理租房提取公积金。感谢您对住房公积金工作的关心与支持！　2019 年 8 月 23 日” 因为它包含了开头，内容，结尾中的应具备的关键词，所以我们认为它的完整性好。

因此我们采取关键词分析的方法，将文本开头、内容、结尾将用到的格式的关键词取出。通过分析每一答复是否具备此关键词而得到完整性的分数。去量纲归一化处理算出最终完整性指标。

我们分别对留言的开头、中间、结尾人工定义出关键词放入 A 库中，然后将答复字段数据预处理过后的词语放入 B 库中。对比 B 库中的单词是否存在 A 库中，我们将 A 库中的词语权重放入在” 完整性权重值.txt”, 进而得到了我们的完整性的评价值。再用 5公式消除量纲得到归一化计算出最终完整性指标。

完整性的流程图如下：

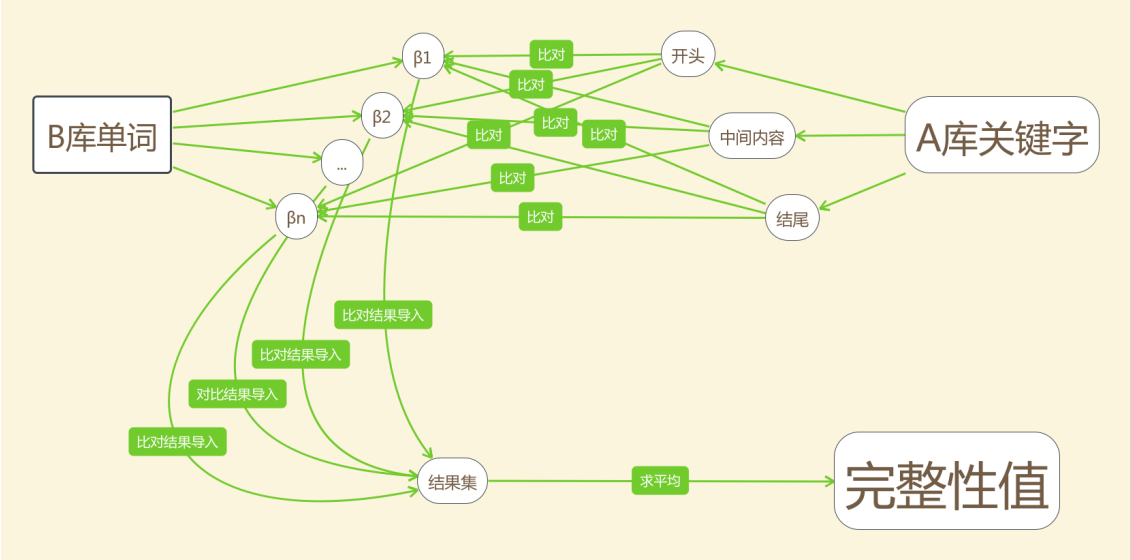

图 8: 完整性的流程图

5.5　可解释性

我们认为可解释性就是对答复的解释是否到位的体现。通俗的讲可以认为是答复意见的意思表达是否清楚。

例如：”您好！您所反映的问题，我办已转交交通、交警等相关部门。感谢您对我县工作的关注，欢迎您和各位网友继续进行监督！　E9 县网信办　2020 年 1 月 9 日” 我们认为他回答了为什么处理不了，是因为事情不归他们管属于交通部门，已经解释清楚了为什么不可以，所以我们认为它的可解释性是高的。

我们认为文段回答的逻辑完整性与与可解释性密切相关，而文本中单词的逻辑完整性越高平均相关概率就越高，于是我们对答复意见是否与自身单词相关计算判断是否具有可解释性。

我们将每一条答复信息的词语加入到 B 库中，对同一答复意见的单词与单词之间用 Word2vec 建立的字典进行相关性分析。对相关程度指数的绝对值求均值，我们用这个值来代表可解释性指标。再去量纲归一化处理得到最终的可解释性方案。

可解释性流程图如下：

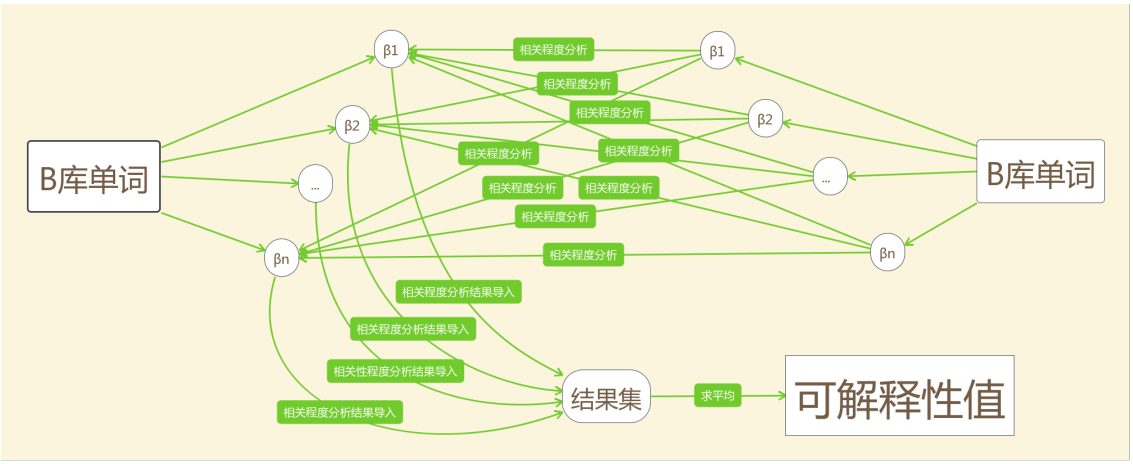

图 9: 可解释性流程图

5.6 评价方案

通过上诉算法得到的三类指标值，我们将相关性、完整性、可解释性三个指标的权重分别设置为 0.4：0.2：0.4。然后用线性权重法计算我们的指标，公式如下：

$$y = n_1 * x_1 + n_2 * x_2 + n_3 * x_3$$

这样就将三个指标代表的文本数据转化成数字指标，得到对文本的最终评价。

6 结果分析

6.1 问题 1 结果分析

通过不断改变训练数据的次数，我们得到的结果也越来越好。我们将经过多次训练反馈得到的精确率可视化如下:

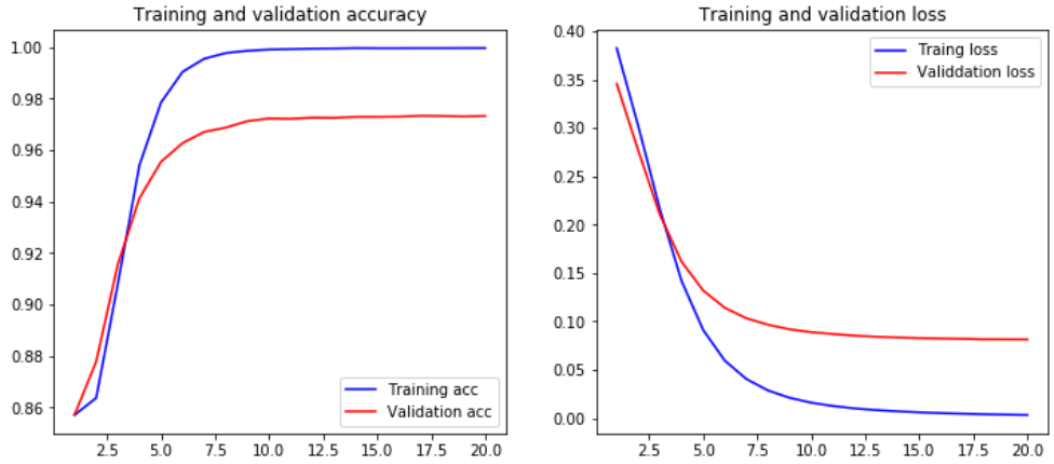

图 10: loss 和 accuracy

我们每次取其中 300 条作为训练样本，并把它们传递给 200 个神经元来训练我们的模型。同时将剩余的留言作为测试指标，以测量训练样本和测试样本的准确线性。经过多次迭代，和不断优化参数和迭代次数，我们发现经过 20 次左右迭代就能达到较高的准确率。

同样，我们从上述图像也能看出，训练样本的精确率接近百分之 100，精确率达 dao 97%。而且不管是训练样本还是测试样本，它们的精确度都呈现出逐渐上升的趋势。这更加有力地说明神经网络算法拥有较好的分类能力。

但是，准确率高不一定代表建立的模型就好，所以还需要通过其它方法来进一步评价我们的模型。为此我们使用 F-Score 评价方法对我们模型进行打分。首先将测集样本传入，准确率为：90.61% ，最后利用极大似然规律将所得到的结果带入 F-Score 中，计算得到以下结果:

$$F1 = 0.8977$$

可以看出这个评分还是较高的，由此可进一步说明我们模型适用于该题。

6.2 问题 2 结果分析

针对问题二，我们对留言主题进行聚类分析，得到了 49 个类别。我们认为在给政府的留言中可能存在刷赞的行为，点赞过高的有点不和情理。所以，为了减少

特殊值对结论的影响，我们在将支持数与反对数求和后，将大于 100 的值取其百分之一作为参数。然后将处理过后的参数赋以 0.2 的权重，同时加上每条评论出现的次数，最终通过线性加权法求得我们的热点指数。

其中热度指数最高的 5 个，它们热点指数分别为 34、32、27.4、14.4、12.4。我们将排名前面 5 的热点问题放入了附录的“热点问题表.xls”，并且将具体的留言放入了” 热点问题留言明细表.xlcs”。

所以我们得到排名前面 5 的热点问题分别为：丽发新城附近搅拌站噪音扰民问题；伊景园滨河苑捆绑销售车位问题；郝家坪小学何时扩建问题; A 市国家中心建设问题；人才租房购房补贴问题问题。

6.3 问题 3 结果分析

通过题目中所给的三个关键指标分别加上权重来进行分析评判打分。我们运用 word2vec 模型建立的字典，通过单词间的关系的分析得到相关的数据，进而对关键指标的好坏评判和采纳标准做出来多次测试，最终得到评判最接近人工评判的一个评价评分算法。我们将评分结果转化成五个星级分布，具体分布情况如下图：

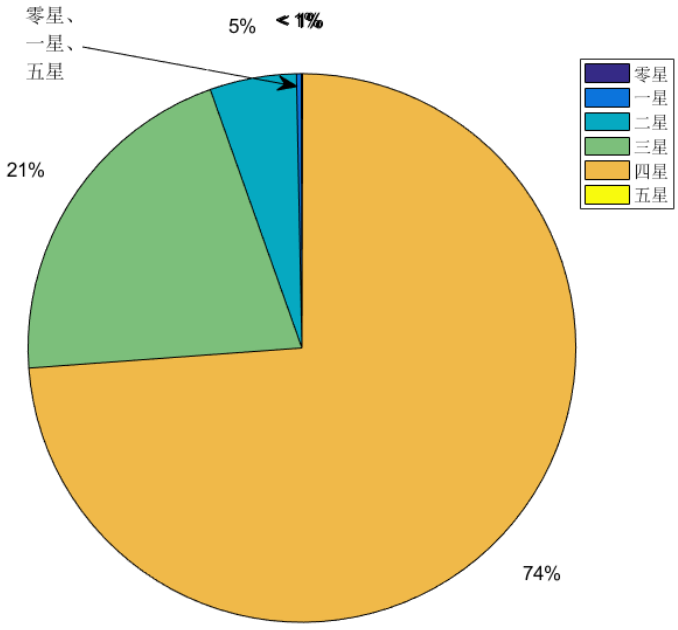

图 11: 星级饼图

结合实际的答复情况和星级评分的分布情况，我们认为制定的评分方案有着的一定的公正性与准确性，得到的结果也符合预期。而且在目标答复数量增加时，星级的评定准确度与客观性也会提高，直到接近人工评断。所以我们认为这个评价方案是可行的，可以在一定程度上代替人工，减少时间和人工成本。

7 结论

为解决这些问题，最重要的步骤就是对行政留言进行分类。但如今这庞大的数据量，已经不像往常一样，依靠人工对留言划分和整理就能得到结论。

传统的浅层机器学习的方法是通过遍历词典，对整个句子的得分进行加总，这方法是构建相应的感情词典。虽然他没有充分使用语义，也没有体现出词和词的关系，但是对于多分类数目比较多的问题，与多二分类，SVM，贝叶斯，卷积升级网络相比，常见的神经网络算法效果还是比较好的。

本文通过建立数据挖掘模型，得到了有一定价值的结论。并且实现了对留言文本数据的指向性分析，以及一定程度上加深了对点赞数、留言问题等文本内容的挖掘认识，而这些结果对问政平台和留言问题中所提到的相关单位都具有一定程度上的指导性意义。比如，民众可以了解问政平台中所发布的留言文本中数据挖掘得到的信息，在参与互动或线下交流之中，促使相关单位进行调整或修改，达到更好的服务民众的目的。

其中题中所给数据偏少，导致有关训练数据偏少，所以我们的模型仅对题目所给数据有较强可解释性。这也是我们今后对文本数据的挖掘研究中可以进一步深入讨论的地方，但是我们根据研究思路撰写本论文，通过实验验证了本模型的可行性，基本实现本赛题设立的目标。

参考文献

[1] 熊富林, 邓怡豪, 唐晓晟.Word2vec 的核心架构及其应用 [J]. 南京师范大学学报(工程技术版),2015,15(01).

[2]《Word2 vec 的核心架构及其应用》• 熊富林，邓怡豪，唐晓晟 • 北邮 2015 年.

[3] https://blog.csdn.net/zhaomengszu/article/details/81452907

[4] https://www.jianshu.com/p/9161679c8d87

[5] https://max.book118.com/html/2017/0729/125042744.shtm

[6] David M. Blei, AndrewY. Ng, Michael I. Jordan, LatentDirichlet Allocation, Journal of Machine Learning Research 3, p993-1022,2003

[7] https://www.cnblogs.com/mainmonkey/p/3813859.html

[8] https://xueshu.baidu.com/usercenter/paper/show?paperid=2b05cb2dc1e5028879b4b97c

[9] https://www.leiphone.com/news/201705/TMsNCqjpOIfN3Bjr.html