摘要

NLP（自然语言处理）是人工智能中一个困难但极其重要的分支，其在社会活动中的应用意义重大。本文针对问政平台上群众留言，构建了基于 LSTM 的自然语言处理模型，该模型可以提取留言的关键内容进行标签分类，解决了第一问需求。

数据预处理阶段中，我们对原始文本数据集进行分词和去停用词处理。利用 Tokenizer 将文本转化为序列，再经过映射和填充转化为定长的向量，输入到 LSTM 模型中训练。经多次实验，对比损失函数和准确率的数值变化，并结合 LSTM 的性质，可以认为本实验模型性能基本稳定。最终得到模型准确率为 82.88%，F1-score 为 80%，可见模型对于群众留言一级分类的有效性。

针对第二问热点问题的挖掘，我们构建了基于 Single-Pass 的聚类模型。首先对群众留言详情的文本数据进行分词和去停用词，再利用 word2vec 完成词向量的转化训练。接着输入 Single-Pass 模型中，利用余弦函数计算矩阵之间的相似度，据此完成文本的聚类并记录同类文本频次，以及其时间集合。为了挖掘群众关注的热点问题，我们设置包含四个指标的评价体系（问题频次、点赞/反对数、问题持续时间和危害程度），引入层次分析法综合性地量化问题的热度，挖掘最受群众关注的五个问题。

针对第三问如何评价答复意见，我们设置了一个评价方案，结合前两问的文本处理模型，能得挖掘出答复意见文本的相关性、完整性、有效性和时效性四个指标，再使用熵权法计算每一项指标权重，从而综合衡量答复意见的质量。

关键词： NLP　LSTM　word2vec　Single-Pass 熵权法

目录

一、简介

1.1 背景

随着互联网的普及和高速发展，线上平台开始成为人民办公、娱乐的另外一大途径。同时，相比于以前的问题投诉要到不同的特定部门的传统模型，且伴随着民意反馈不知其路和意见反馈流程繁杂等一系列的问题，由于“足不出户可知天下”的互联网性质，从而可将民意反馈简单化。开设线上平台，让人民只用一部手机、一个网络，随时随地即可反馈所遇不公或社会问题。

微信、微博、市长信箱、阳光热线等网络问政平台逐渐变成政府了解民意、集聚民智的途径，而相应的留言信息也开始不断增加，相比通过人工标注留言信息及信息分类的传统方式，更增加了很大的难度和带来了极大的挑战。与此同时，随着大数据、人工智能等技术的发展，建立基于自然语言处理技术的政务系统已经是社会治理发展的新出炉，可以提高政府的解决问题的效率和提高民众的幸福感。

建立智能的留言模型,能够智能地识别每条留言信息的类型和对一段时间的留言信息进行整合从而得到“民生热点”的目的，同时对留言答复进行评价。不仅可以减轻相关部门的工作量，也能帮助相关部门对于社会问题快速响应，及时跟进民意并作出工作调整，更有利于提高政府部门工作效率。

1.2 挖掘目标

①针对留言内容，构建一个智能文本分类模型。该模型可以智能地识别留言内容对应的一级标签，从而减轻部门的工作量。在具体的使用上，对于每一条留言，训练后的模型可以根据文本内容判断所属类型，得到对应的一级标签。

②我们需要构建一个文本聚类模型以挖掘热点问题。该模型可以对一段时间内的留言信息进行分类并降序，从而得到该段时间的“热点问题”，并记录“热点问题”的具体留言内容、时间、留言数以及对应地点或人群。同时建立热度评价体系，对“热点问题”的热度进行量化评价。

③设计答复意见的评价方案。经过多个方面，我们从相关性、有效性和时效性三个角度来综合评价答复意见的质量。

1.3 挖掘流程

以一级标签分类模型为例，挖掘过程如下（图 1）所示，主要分为两大部分：预处理部分和模型部分。其中以及标签分类模型的预处理包括分词，去停用词，序列化/字典化以及填充字符。模型部分为核心步骤，为了获得留言的一级分类标签，将预处理得到的词向量放入 LSTM 模型，最终输出分类结果。

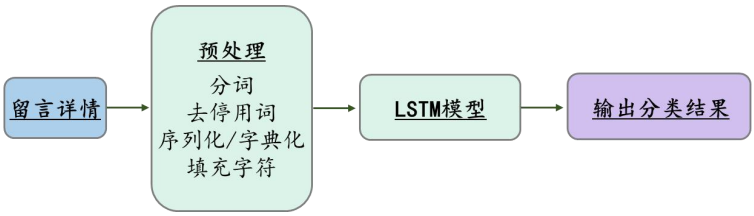


图 1：流程图（一级标签分类模型为例）

二、实验评估

2.1 实验平台

实验环境配置如下表 1 所示：

CPU	Intel(R) Core(TM) i5-7300HQ CPU
显卡	Intel(R) HD Graphics 620
内存	8 GB
操作系统	Windows 10

表 1：实验环境配置

我们主要利用 Anaconda，以 Python 为开发语言，基于 Keras/Tensorflow 开源机器学习框架构建一级标签分类模型；运用 Gensim 中的 Word2vec 将留言转化为词向量进行文本聚类。（软件版本配置：Python 3.7.3，Tensorflow 1.13.1）

2.2 实验数据来源

本次实验数据均主要自出题方以 excel 形式所给的四个附件：附件一.xlsx、附件二.xlsx、附件三.xlsx 和附件四.xlsx。其中，附件二.xlsx 用于一级标签分类模型的训练及（初步）测试；基于附件一.xlsx 中与附件二.xlsx 重合的一级分类部分下对应的三级分类自编留言，以用于进行自定义（实例）预测；附件三.xlsx 和附件四.xlsx 分别用于热点问题挖掘及答复意

见评价模型。

2.3 实验评价指标

对于本文中建立的一级标签分类模型，我们在题目的基础上，采用准确率（Accuracy）、精确率（Precision）、召回率（Recall）、 F1-Score、宏平均（Macro-averaging）和加权平均（Weight-averaging）来对模型的表现效果进行评价。

为方便描述以上评价指标，我们先对混淆矩阵进行说明。

对于二分类问题，可据样例的真实类别和分类模型的预测类别将其划分为四类：

真正例（True Positive，TP）：真实类别为正例，预测类别为正例。

假正例（False Positive，FP）：真实类别为负例，预测类别为正例。

假负例（False Negative，FN）：真实类别为正例，预测类别为负例。

真负例（True Negative，TN）：真实类别为负例，预测类别为负例。

其对应的混淆矩阵如下表所示：

预测类别	真实别类	
	正例	负例
正例	真正例 （True Positive，TP）	假正例 （False Positive，FP）
负例	假负例 （False Negative，FN）	真负例 （True Negative，TN）

表 2：二分类混淆矩阵

①准确率（Accuracy）

准确率，指所有预测正确的样例占总样例的比例，即

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

②精确率（Precision）

精确率，也叫查准率，指被正确预测的预测类别为正的样例占全部预测类别为正的样例的比例，即

$$Precision = \frac{TP}{TP + FP}$$

③召回率（Recall）

召回率，又称查全率，指被正确预测的真实类别为正的样例占全部真实类别为正的样例的比例，即

$$Recall = \frac{TP}{TP + FN}$$

④F1-score

在精确率和召回率两个指标发生冲突时，我们难以在模型之间进行比较，因此引入精确率和召回率的加权调和平均数 F-score 对模型进行评价。

$$F - score = (1 + \beta^2)\frac{Precision \times Recall}{\beta^2 \times Precision - Recall}$$

其中，β 是用于平衡精确率和召回率的参数：如果 β <1，表示精确率比召回率重要；如果 β >1，表示召回率比精确率重要。

令 β =1，即认为精确率和召回率一样重要，得到 F1-Score：

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

F1-score 是统计学中用来衡量二分类模型精确度的一种指标。

然而，我们建立的一次标签分类模型是一个多分类模型，每两两类别的组合都对应一个混淆矩阵。为此，引入了宏平均（Macro-averaging）和加权平均（Weight-averaging）

⑤宏平均（Macro-averaging）

对每一类别的 F1-score 进行简单算术平均，即

$$\text{Macro} - \text{averaging} = \frac{1}{n}\sum_{i=1}^{n} F_i = \frac{1}{n}\sum_{i=1}^{n} \frac{2P_iR_i}{P_i + R_i}$$

其中，F_i表示第 i 类的 F1-score，P_i表示第 i 类的精确率，R_i表示第 i 类的召回率。

⑥加权平均（Weight-averaging）

对每一类别的 F1-score 进行加权平均，权重为各类别数在总样例数中所占比例，即

$$\text{Weight} - \text{averaging} = \sum_{i=1}^{n} \frac{S_i}{S} F_i = \sum_{i=1}^{n} \frac{S_i}{S} \times \frac{2P_iR_i}{P_i + R_i}$$

其中，S_i为第 i 类的样例数，S 为总样例数。

三、预处理

3.1 分词

中文文本与外文文本最大的区别在于词与词之间没有明显的界限,从文本中提取词语分析时需要分词。本文采用 Python 开发的中文分词模块 jieba 进行行分词。

3.2 去停用词

在文本处理中,停用词是指那些功能极其普遍,与其他词相比却没有什么实际含义的词,通常是一些单字，单字母以及高频的单词，比如中文中的“我、的、了、地、吗”等，英文中的“the、this、an、a、of” 等。这些停用词出现频率高，对文本意义贡献小，一般在预处理阶段就将其删除，避免对文本造成负面影响。本文所用的停用词，是将哈工大停用词表、百度停用词表、四川大学机器智能实验室停用词库等停用词表合并且去重后，再经过修改得到的。

3.3 序列化/字典化

Tokenizer 是一个用于向量化文本，或将文本转换为序列（即单个字词以及对应下标构成的列表，从 1 算起）的类。本文在利用 Tokenizer 将经过分词和去停用词的文本词语序列化成词向量列表。

3.4 填充字符

模型的输入数据为固定长度，因此需要对序列进行填充或截断操作。小于固定长度的序列用 0 填充，大于固定长度的序列被截断，以便符合所需的长度。

四、一级标签分类模型

4.1 RNN 和 LSTM

- **RNN 基础**

传统的循环神经网络（Recurrent Neural Network，RNN)，是一类以序列数据为输入，

在序列的演进方向进行递归且所有节点按链式连接的递归神经网络。近年来由于其性能良好，逐渐代替深度神经网络（Deep Neural Network，DNN）成为主流自然语言处理建模方案。相比于 DNN，RNN 的核心思路是网络隐含层有回边，隐含层可以通过这些链接，进而保存并利用历史信息来辅助处理当前数据。简单模型展开后如下图所示：

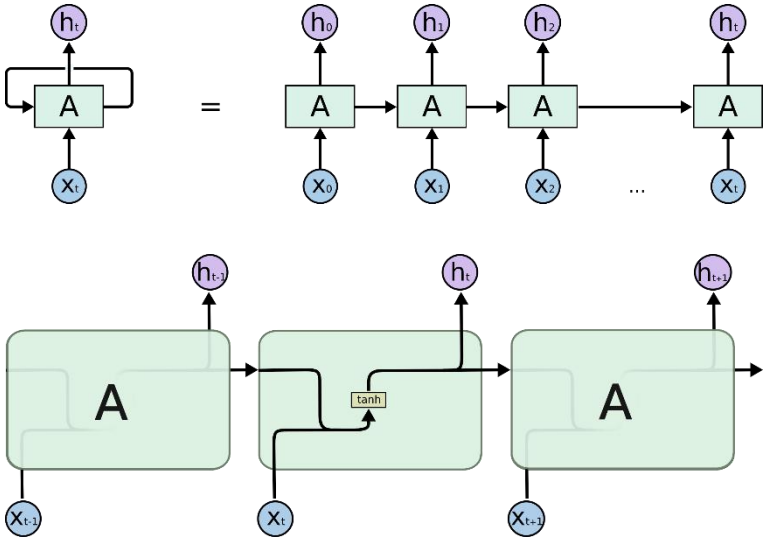


图 2：RNN 模型原理图

RNN 模型看似简单，但也有一些严重的缺点：

①过拟合。在目前广泛应用的端到端 RNN 模型中，RNN 对上下文相关性的拟合较强，而这可能会使得 RNN 比 DNN 更容易出现过拟合问题。

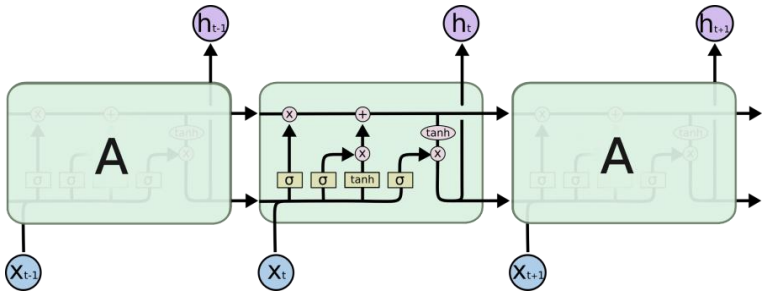
②传统 RNN 在实际中很难处理长期依赖的问题，即难以处理较长的文本。比如对于两个句子，两句话的差别在于主语和谓语动词的单复数，但由于句子比较长，RNN 无法处理这种问题。而这是因为当序列较长时，序列后部的梯度很难反向传播到前面的序列。

考虑到需要建立分类模型的留言较长，同时为避免过拟合，我们选择使用 RNN 模型的变种——LSTM 模型。

- **LSTM 模型**

长短期记忆网络（Long Short-Term Memory，LSTM），是一种时间递归神经网络，适用于处理和预测时间序列中间隔和延迟相对较长的重要事件。相比于传统 RNN 模型只有一个简单的结构（tanh 层），LSTM 的模块中有四个交互的层。同时，LSTM 在算法中加入了一个判断信息是否有用的“细胞”，LSTM 模型通过输入门、遗忘门和输出门三扇门来保护

和控制细胞。当一个信息进入到模块中，通过规则来判断信息是否有用，符合规则的信息被



留下，不符合的信息则会在遗忘门被遗弃。

图 3：LSTM 模型原理图

任一时间步 t 的输入门(i_t)和遗忘门(x_t)输入的变量如下：

（1）输入变量(x_t)

（2）上一个时间步 t－1 的输入向量(h_{t-1})

（3）偏置(b)作为输入，并通过激活函数得到响应值。

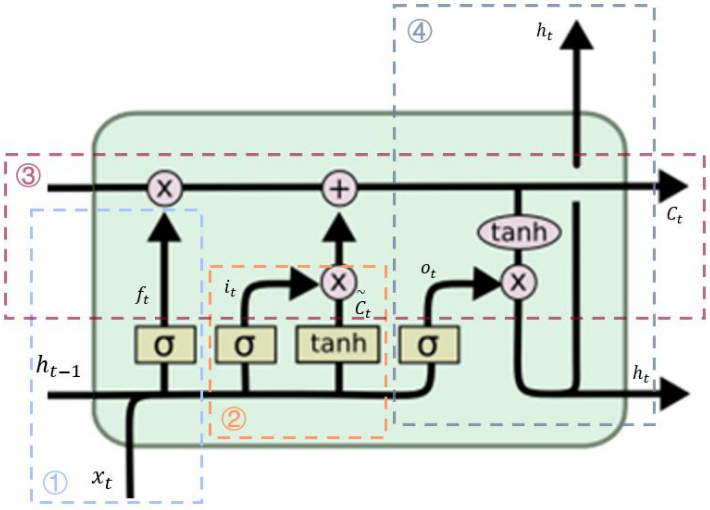


图 4：LSTM 模型单步中四层交互神经网络层

信息输入细胞中，第一步要决定我们从细胞状态中丢弃什么信息，这一步通过遗忘门完成[①]，其公式为：

$$f_t = \delta(W_f \cdot [h_{t-1,}x_t] + b_f)$$

下一步是确定什么样的新信息被存放在细胞状态中[②]。这里包括两个部分，第一部分是通过输入门决定我们将要更新什么值。然后一个 tanh 层将会创建一个新的候选值向量$\tilde{C}_t$加

入到状态中。其公式如下：

$$i_t = \sigma(W_i[h_{t-1},x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1},x_t] + b_C)$$

接下来更新旧细胞状态[③]。将旧状态与f_t相乘，从而丢弃掉我们确定丢弃的信息，然后加上$i_t \times \tilde{C}_t$，进而形成新的候选值，根据我们决定更新每个状态的程度进行变化。从而，C_t的公式为：

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

最后，我们通过输出门来确定细胞状态的输出部分[④]。把细胞状态通过 tanh 处理，并将其与输出门的输出相乘从而使得输出值为我们确定的输出部分。

$$o_t = \sigma(W_O[h_{t-1},x_t] + b_O)$$

$$h_t = o_t * tanh(C_t)$$

4.2 数据分析

分别对附件一.xlsx 及附件二.xlsx 进行基本数据分析。附件一.xlsx 中共有一级分类 15 个，二级分类 102 个，三级分类共 517 个，其中各一级分类下三级分类的数量如下所示：

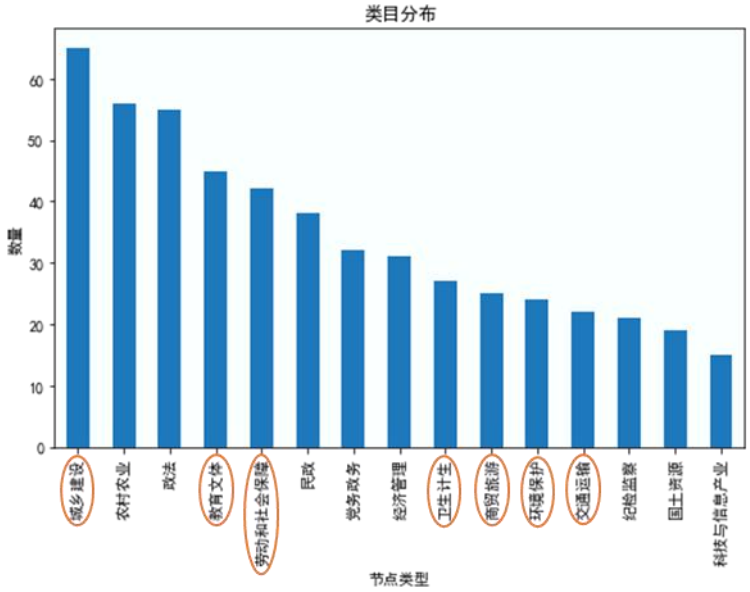
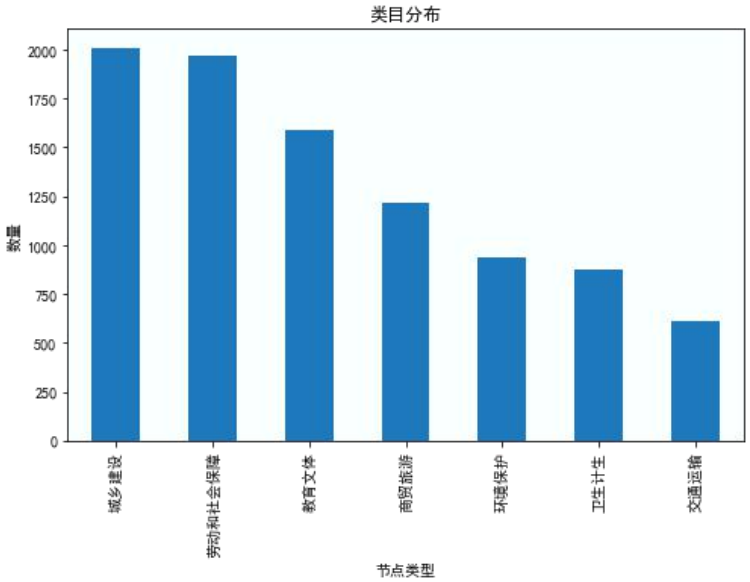


图 5：附件一中三级分类数据统计结果

附件二.xlsx 中共有留言 9210 条（各列数据均不存在空值），其中包含 7 种不同的一级标签（已在上图中用橙色圈标注），留言在各标签中的分布如下所示：

图 6：附件二中各分类下留言数据统计结果（见下页）



经过预处理后，对留言详情进行进一步统计分析，所有留言总词语数为：3272573，平均每条留言有 355.33 个词。因此，设定每条留言最大长度为 355，将留言词组序列化并填充字符。序列化后，共有 82758 个不相同的词语。

4.3 实验设置

我们将经过预处理（分词、去停用词、序列化、填充字符等）后的附件 2.xlsx 数据划分为训练集和测试集，放入 LSTM 模型的嵌入层进行训练。

模型共分为 4 层：嵌入层(Embedding)、SpatialDropout1D 层、LSTM 层和输出层。嵌入层(Embedding)使用长度为 100 的向量来表示留言详情；；LSTM 层包含 100 个记忆单元；输出层为包含 7 个分类的全连接层。设置训练周期为 5，batch-size 为 64。

4.4 实验结果和模型评估

- **损失函数趋势图和准确率趋势图**

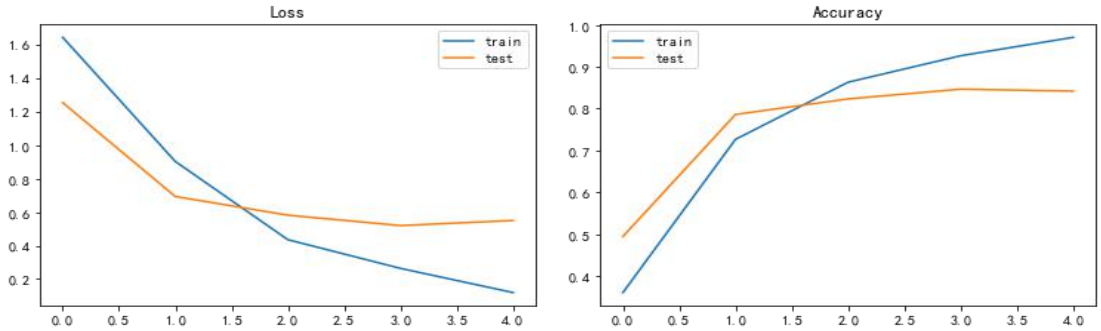


图 7：附件一中三级分类数据统计结果

由图可知，在该次实验中，随着训练周期的增加，模型在训练集的损失逐渐减小，而在测试集的损失呈现先减小后略微增加的趋势；准确率对比损失函数，呈现相反的趋势，即在这次模型训练中出现了轻微的过拟合现象。

考虑到 LSTM 模型的随机性，我们通过反复训练模型发现：大部分情况下模型不出现过拟合现象或较为轻微，仅偶尔出现过拟合或欠拟合的现象占。由此我们认为，该模型的设置是合理的，偶尔出现的过拟合现象是模型随机性导致的。

- **模型评估**

在该次实验中，模型在测试集上应用的混淆矩阵及评估结果如下所示：

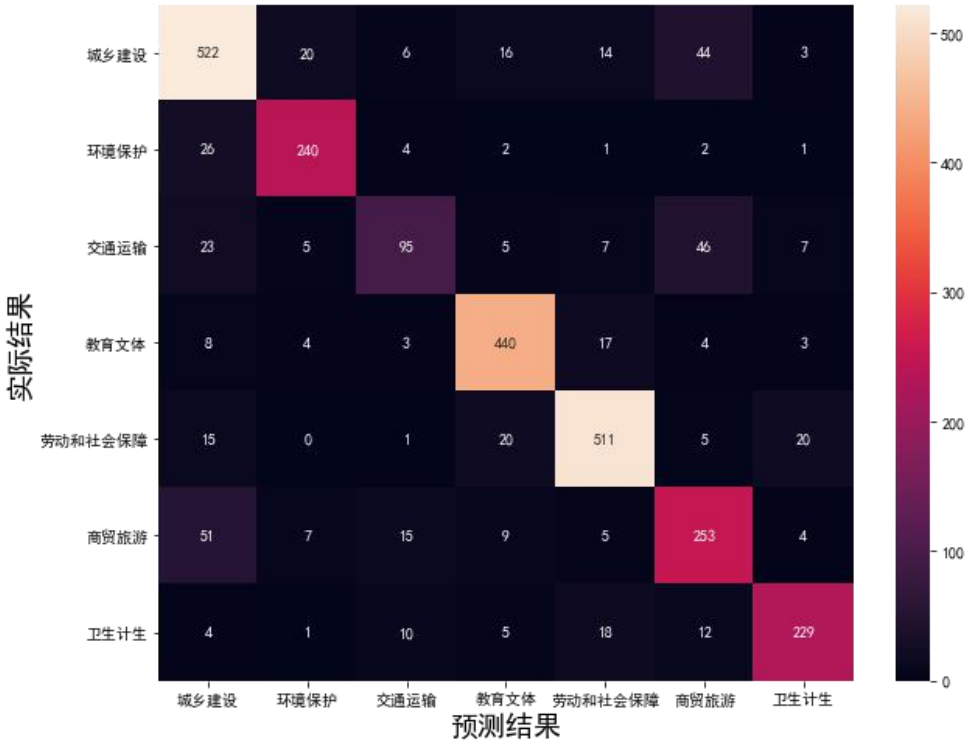


图 8：测试集混淆矩阵结果

	precision	recall	f1-score	support
城乡建设	0.80	0.84	0.82	625
环境保护	0.87	0.87	0.87	276
交通运输	0.71	0.51	0.59	188
教育文体	0.89	0.92	0.90	429
劳动和社会保障	0.89	0.89	0.89	572
商贸旅游	0.69	0.74	0.71	344
卫生计生	0.86	0.82	0.84	279

Accuracy			0.83	2763
Macro-averaging	0.82	0.80	0.80	2763
Weight-averaging	0.83	0.83	0.83	2763

表 3：测试集评估结果

模型准确率 $Accuracy \approx 0.8288$，宏平均 $Macro-averaging \approx 0.80$，加权平均 $\text{Weight}-\text{averaging} \approx 0.83$。多次实验的结果稍有区别，变化幅度均稳定在$\pm 0.01$以内，可见模型整体来看准确性较高。

但交通运输类和商贸旅游类的留言在模型的分类情况明显差于其他类别。结合数据分析结果推测，交通运输类的 $F1-score$ 较低大概率是由于该类的留言样例较少，训练集样例不足；而导致商贸旅游类的 $F1-score$ 较低的线索并不明确，可能是该分类下覆盖范围广，特征词不明确。

- **自定义预测评估**

在模型评估的基础上，我们结合附件一的三级分类标签自行编写了一些简短的留言，并进行分类，与模型预测的分类结果对比如下所示：

留言	人工分类	模型分类
我是来 A 市打工的，跟着工程干了一年了，老板说开发商资金不足，发不出工资。应该向哪里反应才能要到工资？（农民工权益）	劳动和社会保障	劳动和社会保障
家里孩子原本的独生子女证丢失了，请问怎么重办呢？需要带什么证件去重办？（计划生育服务管理）	卫生计生	卫生计生
每天上下班期间,某某道路到某某路段人流车流极多。路口红绿灯又坏了，造成极大的安全隐患。请求尽快维修整改（安全隐患）	城乡建设	商贸旅游
你好我是 xx 学校的学生，我们学校违背教育规则强制学生暑假补课且收费。（乱收费）	教育文体	教育文体
我昨天坐的网约车乱收费，不按正常价格收费，不同意就把车门锁上不给下车（网约车问题）	交通运输	商贸旅游

城中村改建什么时候才能完成啊，已经搞了好几年了（棚户区和城中村改造）	城乡建设	城乡建设
我们医院老是有人来医闹，今天还搞伤了一个医生，医护人员的权益什么时候能得到保障啊（医护人员权益）	卫生计生	卫生计生

*留言后的括号内为编写时对应的三级指标，蓝色的为分类错误的留言。

表 4：自定义预测评估结果

共编写 7 条留言，其中大部分留言都分类正确，未分类正确的两条留言，均被错误分类至商贸旅游类别，与模型评估部分体现出的结果一致。

五、热点问题挖掘

5.1 热度评价体系

先对附件三.xlsx 中数据进行基本统计分析，结果如下：共有留言 4326 条，留言时间、留言详情及反对数、点赞数列中均不存在空值。

结合上述数据情况及热度评价指标体系的相关研究,我们从事件作用力和群众作用力两个方面建立留言的热度评价体系。

①事件作用力

事件的作用力指某地点/人群问题造成的不良影响，由问题持续时间及问题的损失、危害程度来组成。其中，问题持续时间以同类留言的留言时间范围占所有留言的时间范围的比例来衡量，即对持续时间进行标准化；问题的损失、危害程度为辅助指标，通过对同类留言的数量多问题依据其程度进行赋值（得分为 0-1，0 表示没有损失和危害，1 表示损失和危害程度极大）。

②群众作用力

群众作用力由留言量及群众的情绪强度组成。其中，留言量以同类留言的数量占总留言数的比例衡量，即对留言量进行标准化；群众的情绪强度包括激动情绪及愤怒情绪，以同类留言的总点赞数与总反对数之和除于最大总点赞数与总反对数之和标准化的结果进行量化

尽管存在较大的主观性，但由于该评价体系较为简单、指标数量较少，我们采用层次分

析法（AHP），在相关研究的基础上建立对比矩阵对各指标赋权，结果如下：

一级热度指标	二级热度指标	Weight
事件的作用力	问题持续时间	0.0783
	问题的损失、危害程度	0.0783
群众作用力	问题留言量	0.5383
	点赞数+反对数	0.3051

表 5：热度评价指标及其对应权重

5.2 文本聚类模型

- **word2vec**

为了对留言文本进行聚类，我们首先需要将留言进行预处理（分词、去停用词），利用 word2vec 将留言转化为词向量，即将自然语言转换成计算机能够理解的向量。

Word2Vec 是由 Google 的 Mikolov 等人提出的一个词向量计算模型，作者的目标是利用海量的文档数据，从中学习高质量的词向量。该词向量在语义和句法上都有很好地表现，已经广泛应用于自然语言处理的各种任务中。

在本次实验中，我们利用 gensim 库中封装好的 word2vec 进行训练。

- **Single-Pass 算法**

Single-Pass 算法又称单通道法或单遍法，依次输入文本，以增量的方式进行动态聚类。初始时将第一篇文本作为一个话题模型，将每次输入的文本与已有话题模型进行匹配，如果与某话题模型匹配，就将文本归入该话题，若该文本和所有已有的话题模型的相似度度量均小于设定阈值，则将该文本表示为一个新的话题模型。

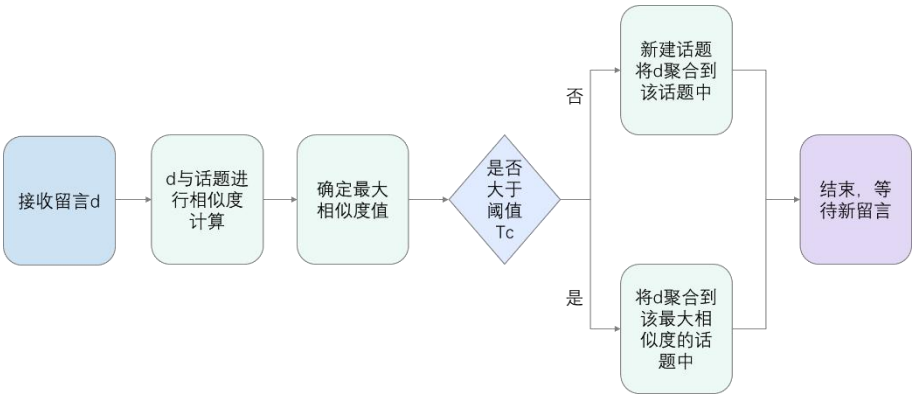
Single-Pass 算法聚类过程如下：

①接收一条留言向量 d；

②将 d 逐一与已有话题中各留言进行相似度计算，并取最大者作为与该话题的相似度。

③在所有话题间选出与 d 相似度最大的话题，若此最大相似度大于预设的相似阈值 TC，d 所对应的留言被分配至该话题模型的文本类簇；若此最大相似度小于于预设的相似阈值

TC，则创建新话题，同时此留言归至新创建的话题模型的文本类簇。



④本次聚类结束，等待下一篇文本的输入。

在本次实验中，我们利用余弦函数计算相似度，即

$$similarity = cos\theta = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

六、答复意见评价模型

6.1 答复评价指标

答复意见的质量在一定程度上反映出相关部门的工作能力和工作效率,对于答复意见的质量进行评价可以更直观地评判出工作能力。基于这个前提，我们将建立一套基于相关性、完整性、有效性、时效性和完整性的答复意见评价方案，通过对答复意见进行评价从而可以更全面地了解部门的工作效益。

①相关性

相关性指的是留言和对应的答复意见两者在内容上的关联程度。留言与答复的相关性强弱可以反映出答复意见是否准确地回应了群众留言的问题。通过构造留言及答复意见的余弦相似度来体现两者的相关度。具体步骤与文本聚类模型中相似度计算一致。

②完整性

完整性指的是答复意见是否完整的回答了群众留言的所有问题。很多情况下，一条留言中提出了多个问题，而答复不一定完整的回答了所有的问题。因此完整性是答复意见评价中十分重要的指标。通过提取留言中的 10 个关键词，以其在答复意见中重现数量的占比来反映。

②有效性指的是对于留言提及的问题，答复意见中是否存在有效的解决问题的方法，即

答复意见内容能否切实解决群众的问题。为了体现答复意见的有效与否，我们在评价过程中对每条答复意见进行 0-1 的赋值，其中“0”表示无效，“1”表示有效。

③时效性指的是相关部门对于留言的答复的速度。它显示着相关部门响应的效率，是否迅速地解决群众问题。我们通过构造 $T=\frac{T_1-T_0}{T_0}$来体现答复的时效性。（T_0表示留言时间，T_1表示答复时间）

6.2 答复意见评价模型

- **熵权法**

在信息论中，熵是对不确定性的一种度量。不确定性越大，熵就越大，包含的信息量越大；不确定性越小，熵就越小，包含的信息量就越小。

根据熵的特性，可以通过计算熵值来判断一个事件的随机性及无序程度，也可以用熵值来判断某个指标的离散程度，指标的离散程度越大，该指标对综合评价的影响（权重）越大。比如样本数据在某指标下取值都相等，则该指标对总体评价的影响为 0，权值为 0。

- **实现步骤**

①计算每条留言其对应的时间相对值X_1、有效值X_2、相关度X_3和完整度X_4。

②对这四个指标X_1、X_2、X_3、X_4数据进行标准化，其中$X_i=\{x_1,x_2.....x_n\}$。则各指标数据标准化后的值为Y_1，Y_2，Y_3，Y_4，那么。$Y_{ij}=\frac{X_{ij}-min(X_i)}{max(X_i)-min(X_i)}$

③求各个指标的信息熵。一组数据的信息熵$E_j=-ln(n)^{-1}\sum_{i=1}^{n}p_{ij}\,ln\,p_{ij}$，其中$p_{ij}=Y_{ij}/\sum_{i=1}^{n}Y_{ij}$。如果$p_{ij}=0$，则定义$\lim_{p_{ij}\to 0}p_{ij}\,ln\,p_{ij}=0$

④确定指标权重。计算出这四个指标的信息熵E_1、E_2、E_3、E_4，通过信息熵计算出各个指标的权重$W_i=\frac{1-E_i}{k-\sum E_i}$

⑤进而得到评价模型

参考文献

[1]梁昌明,李冬强.基于新浪热门平台的微博热度评价指标体系实证研究[J].情报学报,2015,34(12):1278-1283.

[2]张一文,齐佳音,方滨兴,李欲晓.非常规突发事件网络舆情热度评价体系构建——基于 BP 神经网络算法(英文)[J].中国通信,2011,8

[3]殷风景. 面向网络舆情监控的热点话题发现技术研究[D].国防科学技术大学,2010.