基于自然语言处理的“智慧政务”

摘要

本文旨在解决智慧政务系统中由于用户留言数据量增大所带来的人工处理工作量大的问题，希望通过机器学习和数据挖掘的方法对用户不同的留言数据进行处理，并建立一个可靠的计算机模型来对留言问题给出普适化的留言方案，减少了人工操作的工作量。

针对问题一，我们基于 bert 模型和 word2vec 模型给出了两套分类的方案，引入了线性加权融合法对两种模型给出的结果进行权重取舍，最终得到我们认为可行的分类结果，在充足的训练数据前提下，模型的预测正确率达到 93.6%。

针对问题二，我们使用静态权重结合了基于 CRF 的地点信息提取和 tf-idf 模型的内容相似度计算，得到留言的相似度数据，用 single-pass 聚类方法得到热点问题分类，最后，我们参考魔方秀和 reddit 热度计算算法，给出了我们的热点问题评价模型并排序得出热点问题表。

针对问题三，我们从相关性，完整性和可解释性分别制作了三个独立的评分模型在保持各指标评分的平均值和方差的情况下，得到每条回复的三个维度评分，并结合维度权重计算出其最终评分，从结果导向来看，我们的模型基本上达到了人工评价的满意水准。

关键词： 机器学习 bert 词嵌入 分类聚类 评价模型

Abstract

The purpose of this paper is to solve the Wisdom Government System due to the user a message data to increase the problem of manual processing workload big, want to the user by using the method of machine learning and data mining different message data processing, and establish a reliable computer model to generalize message is provided for message problem solution, reduce the workload of manual operation.

Aiming at task one, we proposed two classification schemes based on Bert model and word2vec model, and introduced the linear weighted fusion method to make a weight choice of the results given by the two models. Finally, we obtained the classification result we thought feasible. Under the premise of sufficient training data, the prediction accuracy of the model reached 93.6%.

Aiming at task two, we use the static weight is a combination of location information extraction based on CRF and tf-idf similarity calculation model, get the message the similarity of data, use single - pass clustering method to get hot question classification, finally, we refer to the Mofun-English and reddit heat calculation algorithm, gives us the hot issues of evaluation model and sorted hot issue list.

Aiming at task three, we from correlation, integrity and interpretability respectively produced three separate score model in keeping the score of every index under the condition of the mean and variance, get reply each of the three dimensions of the score, and connecting with the dimension weight to calculate the final score, from the point of result orientation, our model basically achieved the satisfaction level of artificial evaluation.

Keywords : Machine learning; Bert; Word embedding; Classification and Clustering; Evaluation model

目录

1、问题描述

1.1、背景描述

近年来，随着互联网技术的发达，人们在互联网投诉建议的途径也越来越多，随着投诉留言数据量的增大，众多留言数据的分类整理便成了一个待解决的问题。由于大数据和机器学习技术的不断更新普及，相对于传统的人工识别的方法，管理人员希望通过建立计算机模型来解决留言的整理问题，因此，题目中给出了三个问题和数据：问题一是留言的划分整理，通过留言的分类之后再将问题传达到相应的职能部门手中，数据中给出了整理出的留言数据和分类结果，希望利用这些数据训练出一个准确率高的分类模型。问题二是热点问题挖掘问题，热点问题的处理效率是评价管理部门的标准之一，处理热点问题的流程中，对热点问题的查找和快速反应是关键，这就需要依靠留言检测人员对热点的嗅觉来完成，现在管理人员希望通过附件中给出的留言时间，内容等数据，用机器学习的方法帮助人们整理出热点问题的表格。问题三是针对相关部门最终给出的答复，从多个角度建立对相关部门答复的评价模型，并尝试对附件中给出的答复数据进行评分。

1.2、问题定位

通过对题目的讨论分析可知，题目中涉及的问题归属于自然语言处理领域，需要用到预处理，词嵌入，分类聚类，评价模型等技术。问题一是对留言的分类问题，通过已有数据的有监督模型训练，得到对新文本的分类预测模型，主要处理流程为：数据预处理—词嵌入—分类—评价模型等步骤。问题二是文本的聚类问题，通过对留言数据的分析整理，生成聚类结果并根据评价模型排名。问题三是一个相对开放的题目，需要定义两个文本之间的关系评价模型，从相关性，完整性，可解释性等方面出发，制定一套答复评价方案。

2 、问题一

问题一是群众留言分类，目前，大部分电子政务系统还是依靠人工根据经验处理，存在工作量大、效率低，且差错率高等问题。 我们希望通过计算机人工智能自动筛选分类，一边缓解政务工作人员人力不足,及时采集、存储、归类相关政务信息,推动政务服务质量提升。简单来说，这是一个有监督的 NLP 的分类问题，基于给定已经分好类的评论数据，对未知分类的数据自动分类。

2.1、模型分析和建模思路

对文本分类的方法总的归为两个大类，传统机器学习（如随机森林 RF，svm，分类，knn 模型）和深度学习（LSTM，快速文本，卷积神经网络，循环卷积神经网络）。在 2018Bert 模型横空出世，横扫 NLP 领域，打破多个 NLP 记录。学术研究需要跟紧时代技术发展，在问题一中我们主要使用 bert 模型实现一级分类，同时研究 word2vec 词向量和多种神经模型，对比各个分类效果，将其中取得较好结果的模型整合。

2.2、Bert 模型介绍

Bert 模型是针对语言理解的深度双向 Transformers 模型的预训练[1]，基本变压器编码原理如下图：

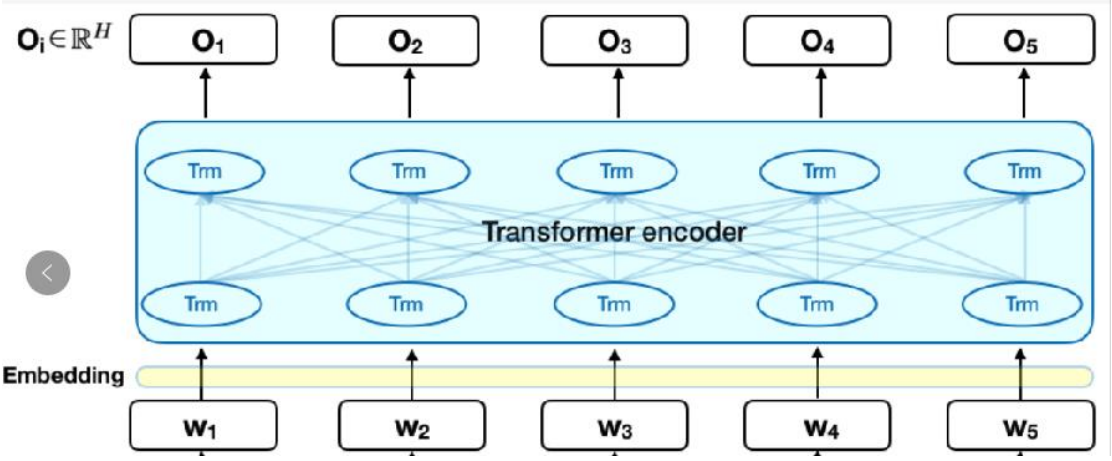


图 1: 变压器编码图

2.2.1、Bert 的输入表征

Token Embedding：词特征（词向量）的嵌入，针对中文，目前只支持字特征嵌入[2]；

Segment Embedding：词的句子级特征嵌入，针对双句子输入任务，做句子 A，B 嵌入，针对单句子任务，只做句子 A 嵌入；

Position Embedding：词的位置特征，针对中文，目前最大长度为 512；

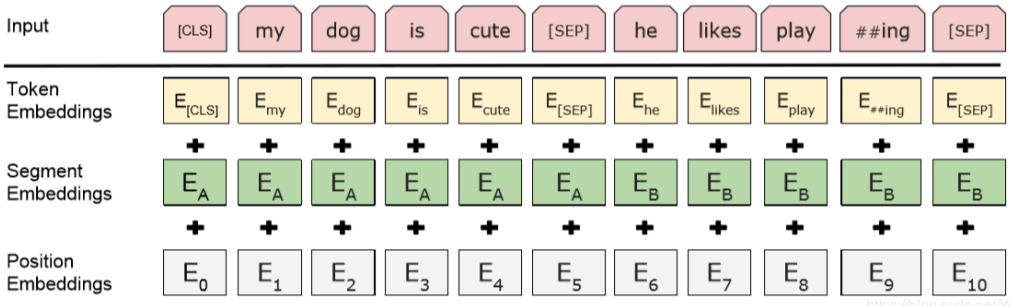


图 2：输入表征图

2.2.2、bert 模型的预训练

Bert 的预训练包含两项任务：语言模型和句子对关系判定。

双向语言模型：随机将输入中 15%的词遮蔽起来，通过其他词预测被遮盖的词（这就是典型的语言模型），通过迭代训练，可以学习到词的上下文特征、句法特征等，保证了特征提取的全面性，这对于任何一项 NLP 任务都是尤为重要[3]。

句子对判定：输入句子 A 和句子 B，判断句子 B 是否是句子 A 的下一句，通过迭代训练，可以学习到句子间的关系，这对于文本匹配类任务显得尤为重要。

2.2.3、Bert 的使用方法

可以将 Bert 看做一个文本编码器，可以应用在各类 NLP 上下游任务网络构建中作为文本嵌入层使用。如我们下文需要使用的文本分类任务和文本匹配任务操作步骤为：

（1）加载预训练 BERT 模型;

（2）取输出字向量：embedding = bert_model.get_sequence_output();

（3）根据具体需求使用，可以结合不同的神经网络使用。

2.3 Bert 模型训练和测试过程

2.3.1、数据预处理

由 2.2.1 Bert 的输入表征中我们知道 Bert 具有能够自动提取输入文本的特征，因此我们不需对文本进行去去停用词，提取关键词等常规预处理方法。根据我们的观察和模型需要，需要进行下列几点预处理操作：

（1）Excel 表中数据头部和尾部存在大量的空格和换行，将其去掉便成为留言内容和留言标题的文本。

（2）将提取的训练数据顺序打乱，原顺序是同一个类别的数据都聚在同一块，容易对训练模型过拟合，严重影响模型结果。

（3）部分两条留言内容极为相似的情况

2.3.2 数据集的构建

数据集分为训练数据，验证数据和测试数据。将全部数据随机取出 70%作为训练数据，同时训练数据打乱顺序作为验证数据，剩下的 30%数据作为测试。

2.3.3 阶段一：示例数据运行

使用示例数据运行的结果正确率只有 85.876%，这个结果不算很差，但还不能满足我们的要求。在对训练过程和数据层面分析原因后是存在两种情况：

（1）数据量少过：所有示例数只有 496 个数据，七个分类平分到各个分类的数据就更少。

（2）各个数据之间数量不平衡：最多的城乡建设有 106 个数据，最少的环境保护只有 48 个，

为了验证是由于部分数据量少影响模型特点不明显的问题，另外将数据量最多的

城乡建设（106 个）和劳动和社会保障（105 个）全部数据，同样取其 70%为训练数据，30%为测试数据，将模型的分类结果和正确分类结果对比正确率达到 92.1875%，部分运行结果如下：

正确值	计算出的验	正确率
劳动和社会	劳动和社会	92.19%
城乡建设	城乡建设	
劳动和社会	劳动和社会保障	
劳动和社会	劳动和社会保障	
城乡建设	城乡建设	

图 3：预测结果展示

由于数据个数还较少，我们等到全部数据的时候再判断数据数量的不平衡是否影响了模型的正确率。

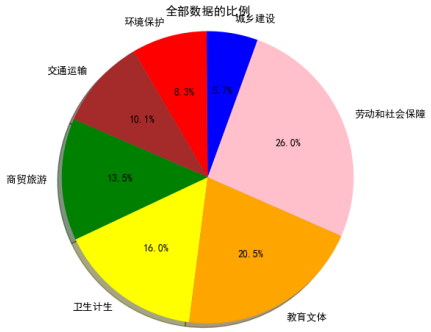
2.3.4 阶段二：全部数据运行

同样的划分方法对模型分出，训练数据，验证数据和测试数据。使用全部数据运行的结果正确率达到 92.5054%，F-score 值为 0.923802 并将结果打印在 Excel，内容如下图：

	A	B	C	D
1	正确值	计算出的验	正确率	F-score
2	城乡建设	城乡建设	0.925054	0.923802
3	商贸旅游	商贸旅游		
4	城乡建设	城乡建设		
5	环境保护	城乡建设		
6	教育文体	教育文体		

图 4：预测结果展示

为了判断数据数量的不平衡是否影响结果的正确率，我们分别做出全部数据，测试数据，和计算出的验证数据中的各个一级分类所占总数的比例圆饼图如下：



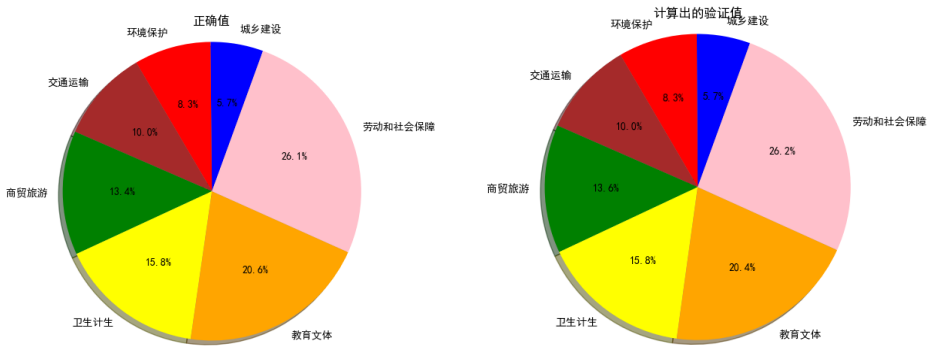


图 5：分类占比分析

由上图可看出，每个一级分类的标签所占的比例都差不多，可见模型验证的时候不会因为数量的多或少而影响模型的判断，所以数据量的不平均不会影响模型的正确率。

2.4、基于 word2vec 的分类器模型

2.4.1、模型思路

Word2vec 是一个较为常用的词嵌入模型，主要用途有 skip-gram 预测上下文和 cbow 预测输入[4]，所以由于其词袋模型的特性，我们打算利用 cbow 预测模型中训练过程的过程数据，即就是在结果预测过程中对各结果节点权重来作为词语之间的相似度参数，通过计算两篇留言中每一个词语的相似度值来作为两篇留言的相似度数据，最后通过有监督的分类器模型来生成最终的分类结果，具体流程如下：

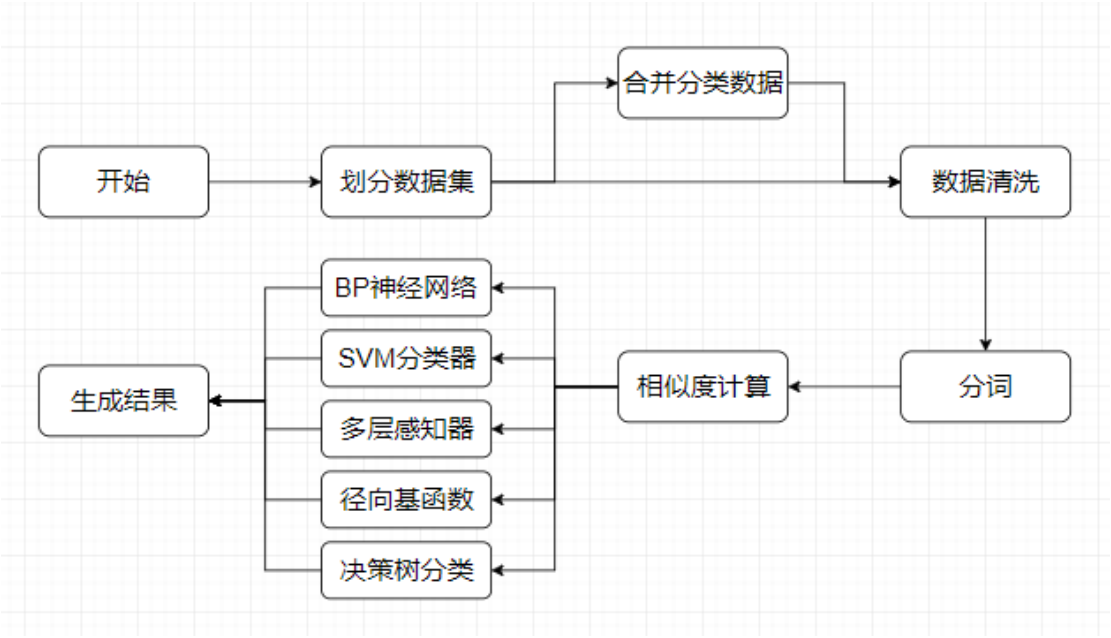


图 6：基于 word2vec 的分类流程

2.4.2、jieba 分词处理

在分词处理之前，我们需要打乱数据集顺序并划分训练数据集和测试数据集，在训练阶段，我们的训练集和测试集按大小比例为 7：3 划分，划分完训练集之后，对应数据集中的七个一级分类结果，把训练集中数据按分类标签归结为七个类别，每个类别对应的是此类别中的留言数据集，对每个类别中的留言合并之后，相对应的七个自定义语料库。

在分词阶段，我们对上一阶段生成的七大语料库和测试集中的留言数据进行分词处理，分词处理主要分为三个步骤进行：

（1）数据清洗，过滤掉留言内容中出现的空格，换行符等数据

（2）内容拼接，由于给定的数据集中有“留言标题”和“留言内容”两种数据，我们认为，留言内容和留言标题信息的重要程度不一，所以需要对两者的内容进行权重赋值之后再拼接成一个合适可行的分词内容，经过参考《参与式感知下环境评价指标权重确定方法》[5]和多次的权重参数调整之后，我们给出了我们认为合理的标题：内容权重占比：

$$\text{Title}:\text{Content}=0.14:0.86$$

（3）分词处理通过内容拼接得到最终需要分词的内容之后，我们使用了 python 下的 jieba.posseg 包进行内容分词，得到分词结果和词性标注数据。

（4）停用词处理，为了提高我们分词结果的质量，需要对保留下对分类结果影响较大的词性和对分词结果进行通用词处理，在词性分析方面，由于 jieba.posseg 会返回分词词性这一特点，我们对分词结果中的特殊词性进行过滤，具体过滤掉的词性列表参考代码：

```
stop_flag = ['x', 'c', 'u', 'd', 'p', 't', 'uj', 'f', 'r']  # 过滤掉特殊的词性
```

图 7：词性过滤表

在停用词方面，我们参考使用了百度大脑 AI Studio 数据集中的 stopword.txt 数据集[2],该停用词数据集词量大，停用词条具有普适性，不受领域词汇影响，可以过滤掉分词结果中区分度不大的数据。

2.4.3、基于 word2vec 的文本相似度计算

在划分训练数据集和测试数据集之后，我们可以通过计算测试集中的每一条留言相对于各大类别语料库中的相似度作为分类器的输入参数，具体的计算流程为：

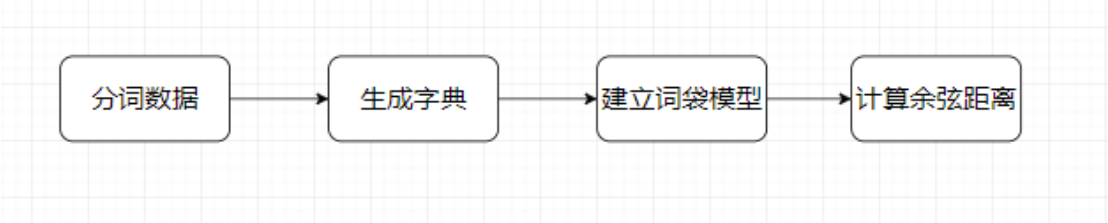


图 8：相似度计算流程

在实现技术上，我们使用了 python 的 genism 库作为工具来完成流程操作，在计算向量间的距离模型中，我们综合对比了欧氏距离、曼哈顿距离、切比雪夫距离、夹角余弦距离几种向量距离模型，从测试数据中结果的表现来看，夹角余弦距离是比较可靠的距离模型，所以我们最终选用了夹角余弦距离模型来计算向量距离。在相似度计算的过程中，我们需要计算出训练集各留言相对于七个一级分类的相似度数据，即数据量为 n*7 相似度数据和结果标签作为训练数据，此外，还需要计算测试数据集中的各个留言相对于七个一级分类的相似度数据但不需要结果

标签，生成训练集和测试集相似度文件之后，作为分类器的参数输入得到分类结果。

2.4.4 分类器模型测试与使用

在分类器模型方面，我们准备了多种常见的分类器模型作为分类标准，分别是归一化后的 bp 神经网络模型，svm 分类模型，径向基函数分类，决策树分类。统一使用同一次相似度计算得到的相似度数据，每个分类器各做了三次随机划分训练集得到的结果取平均数，得到的各分类器的测试结果对比如下：

	BP 神经网络	SVM	径向基函数	多层感知器	决策树
测试正确率	0.8023	0.7814	0.8762	0.8537	0.6521

表 1：分类器性能对比表

所以，由测试结果的平均正确率可知，在对分类数据的有监督分类中，多层感知器和径向基函数的表现较为良好，正确率均在 85%以上，因此，我们着重对比了两种模型在训练过程对数据得敏感度效果，绘制出了两种模型得 roc 曲线，roc 曲线对比如下：

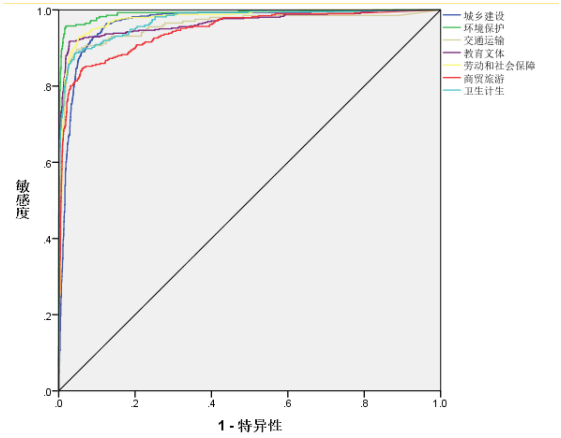


图 9：多层感知器 roc 曲线

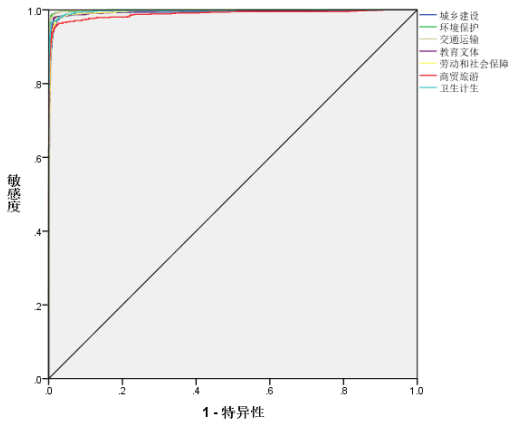


图 10：径向基函数 roc 曲线

从两个分类器的 roc 曲线上看，两者的 roc 曲线下部分面积都较高，较为向左上角倾斜，且没有出现大幅度震荡的情况，说明就模型的相似度计算结果而言，我们得到了较为理想的相似度结果数据，不同分类之间的差异性较为容易判断。对比两个 roc 曲线来看，显然径向基函数的 roc 曲线较为理想，曲线下部分面积相

对于多层感知器的 roc 曲线更大，且各分类标签之间的差异波动较小所以，我们可以得出结论，在该问题中使用径向基函数分类器的预测结果要优于多层感知器。

2.5、线性加权融合处理

如上面所述，我们模型正确率已经达成高于 92%，虽然可以说是较高的正确率，但在评论数量巨大的基数下，仍然还出现较大的判断错误，我们希望能将模型正确率尽量多提高。我们模型运行的测试结果生成 test_results.tsv 文件，里面是每个测试句子对于各个分类的概率，如下图示：

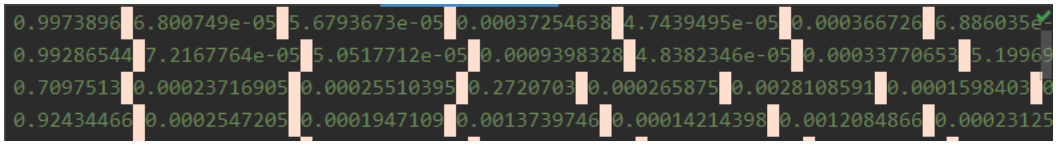


图 11：分类概率示例

我们取每个数据概率最高的那个分类作为该文本的测试结果，如截图中数据的结果都是取分类标签的第一个作为对应的结果。在查看数据时候发现，存在一些最高概率只有不到 0.65 的数据，如在这说明模型对于这部分句子的判断是较为的模糊和不够精确的，如果能针对这部分数据矫正，对于模型的进一步提高具有很大的帮助。在算力丰富的今天，我们想到能不能将算力来换取正确率，我们将训练的数据打乱顺序，用于训练后得到不同的几个模型（在这次我们建立十个），再将这些模型对应每个句子的测试结果取众数，用概率论的知识不难知道整个模型的准确率得到很大的提高。运行结果显示这个方法取得了不错的结果，此时模型的正确率达到 93.2%。

在 bert 和 word2vec 的模型融合上，我们采用了线性加权融合法对两个模型的结果数据进行融合处理，根据训练阶段两个模型的分类表现确定其权重值参数，最终模型的正确率达到 93.6%。

正确值	计算出的验	正确率
城乡建设	城乡建设	0.936278
商贸旅游	商贸旅游	
城乡建设	城乡建设	
环境保护	城乡建设	
环境保护	城乡建设	
教育文体	教育文体	

图 12：预测结果图

2.6、模型评价

我们模型经过测试后正确率达到 93%以上，这个结果可以满足大多数的政务需求，使用 GPU 计算整个模型训练时间不到一个小时，算力损耗低，效率高，正确率高。能够有效降低政务工作人员的负担，体现智慧政务的要求。

3、问题二

3.1、模型介绍

问题二是对网友的留言进行归类，提取出某一时间段内特定地点或特定人群的热点问题，帮助有关部门进行有针对的处理，提高服务效率。这一部分需要用到命名实体识别、文本聚类、文本摘要提取等技术，大体流程图如下：

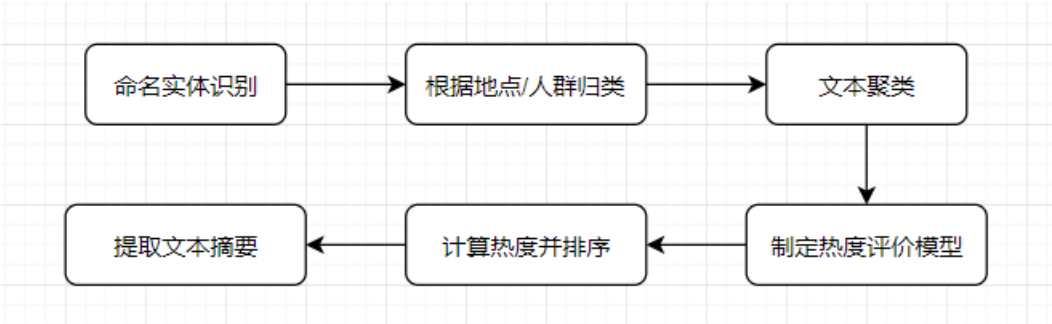


图 13：热点问题归类流程图

为提取出地点或人群，我们采用了基于 crf 优化的命名实体识别，在根据地点或人群信息把满足同一地点或同一人群的留言进行归类，考虑到同一地点或同一人群可能会同时出现多个不同的问题，我们在进行归类后针对同一地点类或人群类中的留言进行文本聚类。

文本聚类采用的算法是 Single-pass clustering，它是一种简洁且高效的文本聚类算法。由于留言数据没有类别相关的信息，所以我们不能采用有监督的方式进行聚类，而 Single-pass clustering 的优点是可以进行无监督聚类，不需要指定类目数量，可以通过设定相似度阈值来限定聚类数量。不仅如此，Single-pass clustering 是一种增量聚类算法，每个留言只需流式计算一次，效率较之其他

的聚类算法有很好的提升。

在得到聚类数据之后，我们需要对得出聚类的热点问题进行热度评价，根据热度对问题进行排序。考虑到留言数据中包含了时间、点赞数和反对数信息，所以在计算热度时我们加入了这些影响指标，同时我们的热度计算还是以留言数为主，并给出了我们的热度计算公式。

完成上述步骤后，再根据留言信息生成问题的文本摘要作为问题描述。这一部分主要是通过 TF-IDF 算法提取出各个留言的关键字，通过关键词找出包含分值最高的簇的句子，将他们整合形成问题的摘要[6]。采用 TF-IDF 是考虑到热点问题会存在多条留言，而多条留言则会频繁出现相关热点的关键词，通过词频统计的方式将关键词提取出来会达到比较好的效果。

3.2、基于 single-pass 的文本聚类算法

3.2.1、模型介绍

Single-pass 是一种实现较为简单实用的聚类方法，相对于 k-means 和其他的一些层次聚类方法，single-pass 在新闻领域问题上的表现要优于其他分类聚类算法[7]，所以我们参考了 single-pass 聚类的思想，结合相似度模型，得出我们的初步聚类结果，将初步聚类的结果作为评价模型的输入参数，具体的实现流程如下：

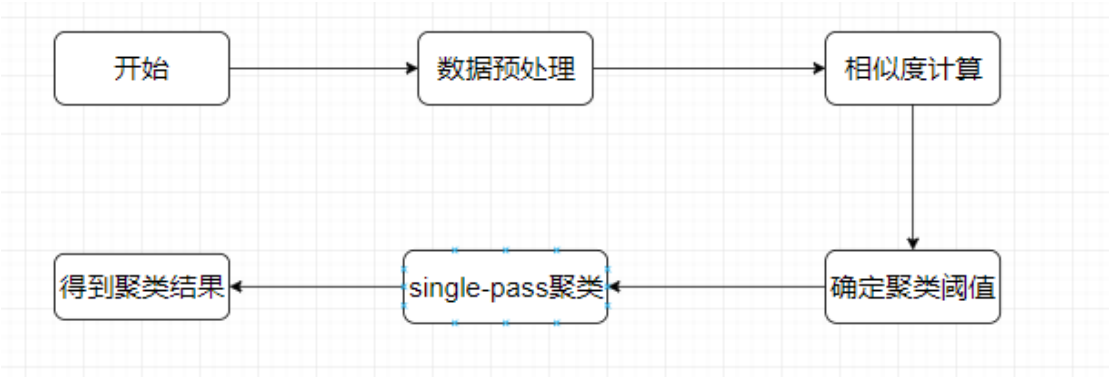


图 14：初聚类流程图

3.2.2、数据预处理

我们抽取了 10%的留言样本作为观察样本，进行人工观察，并试图找出样本

中可能出现的数据格式问题，经过我们的观察得出结论，数据中需要进行预处理的地方主要有：

（1）留言中制表符，换行符，空格等内容可能会影响到分词结果。

（2）可能会出现两条留言内容极为相似的情况。

（3）留言内容中一般含有礼仪规范用语，可能会稀释相似度结果。

针对以上几种数据中出现的问题，我们给出了对应的处理策略：

（1）利用正则表达式过滤掉敏感字符。

（2）对留言极为相似的情况，我们认为这种情况出现的原因是数据库存储异常或者留言的时候表单重复提交，即两条留言表述的内容一致，时间相近且留言作者是同一人的情况，所以我们对这类的留言数据进行了合并，随机保留任意一条，且合并点赞量和反对量，具体做法是：对所有的留言进行两两相似度计算，当某次相似度结果大于 0.95 时，我们就认为是重复留言，进行合并操作。

（3）对于包含礼仪用语的留言，在计算相似度时，除了引用百度 AI Studio 的停用词文件以外，我们在其基础上将样本数据中出现的礼仪用词数据添加到停用词文件中，分词阶段后会过滤掉礼仪用词，具体分词实现参考 2.4.2。

3.2.3、基于 CRF 优化的地点命名实体识别

在“如何判断两条留言表述的是同一问题”的问题上，我们主要通过两点来确认我们的结果，地点提取和内容识别，即就是发生同一地点的同关键词的留言我们可以认定为同一类留言，所以，我们对每条留言的信息都基于 jieba 分词和 CRF 优化做了地点命名实体识别[8]，在 jieba 词性标注分词的基础上，我们发现分词的效果并不理想，具体表现在于 jieba 分词在类似词语‘a5 区’等地点词的识别上无法识别周全，所以我们参考了 CRF 实体识别的思路，建立了自定义的匹配规则库，提升了省，区，县，小区等后缀词的匹配几率，最终达到了我们满意的效果，取部分识别内容展示如下：

图 15：地点提取效果示例图

3.2.4、内容相似度计算

在内容相似度计算上，我们考虑了地点和留言内容两大因素，对提取出的地点实体相似度计算方面，由于 tf-idf 的词袋模型特性，我们的地点实体词条可以得到充分发挥，所以最后我们选用了 tf-idf 模型来计算地点相似度。在内容相似度上我们继续沿用了 word2vec 的相似度计算思路，在此基础上做了修改，使得其适应问题二中涉及的相似度计算，修改的内容主要有：我们放弃了问题一中的中心化相似度计算思想，对问题二中的每一条留言数据看作是一个节点，生成各含有节点特征的字典数据，则可以得到 n 个语料库模型，得到各留言的语料库模型之后，再使用留言拼接之后的内容计算内容相对于另一节点模型的相似度数据，从而得到所有留言之间的相似度二维数组。随后，我们定义了一条函数来对地点相似度和内容相似度进行加权统计，我们希望内容相似度权重能略高于地点相似度，但是需要消除数据值大小差异带来的权重值影响，所以最终我们给出的函数如下：

$$\text{weight} = \frac{\text{avg(text)}}{\text{avg(place)}}$$

$$\text{similar} = \text{place} * 0.3 * \text{weight} + \text{text} * \frac{0.7}{\text{weight}}$$

$text$：*内容相似度*　　$place$：*地点相似度*

3.2.5、确定聚类阈值

基于 single-pass 的文本聚类是一个较为依赖于主观性的聚类模型，即聚类的效果受输入参数的值和输入顺序影响较大，其在参数上的体现在于聚类阈值 theta 的确定上，相似度计算之后大于 theta 值的两个节点会归于同一类。因此，为了得到效果相对较好的聚类阈值，我们对相似度计算得到的结果进行了可视化分析，由于相似度矩阵的数据维度较大（n*n，n>4000），我们采用随机抽取的方法取出10%的数据进行可视化分析，得出的相似度曲面图如下：

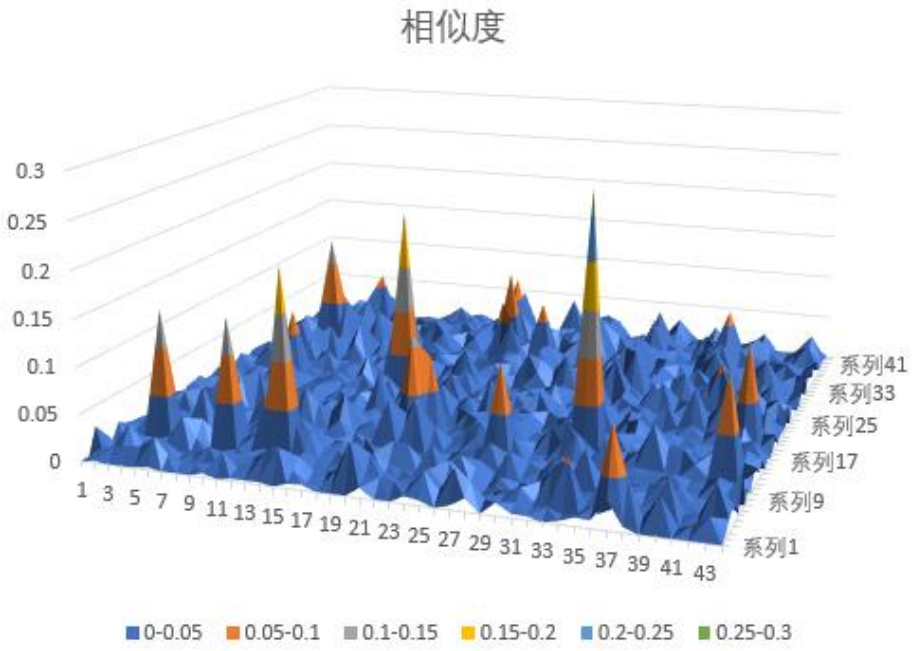


图 16：相似度分布图

该图中，z 轴表示两个节点的相似度数据，x，y 轴各表示一个节点，很明显从相似度曲面图上来看，属于同一分类下的两条留言的相似度数据较高，即在曲面图上的曲面凸包即为我们认为的属于同一话题下的两条留言数据，因此，我们采用了横向分割的方式，将数值大于 theta 两条留言归结为同一类留言，从经验总结的角度给出了我们认为的合理的 theta 值：

$$\mathrm{Theta} = 0.10$$

3.2.6、single-pass 聚类

在计算出相似度二维数组和确定 theta 之后，我们开始参考 single-pass 的思

路进行了话题聚类，并对聚类出来的话题根据留言总数进行初步排名，具体流程如下：

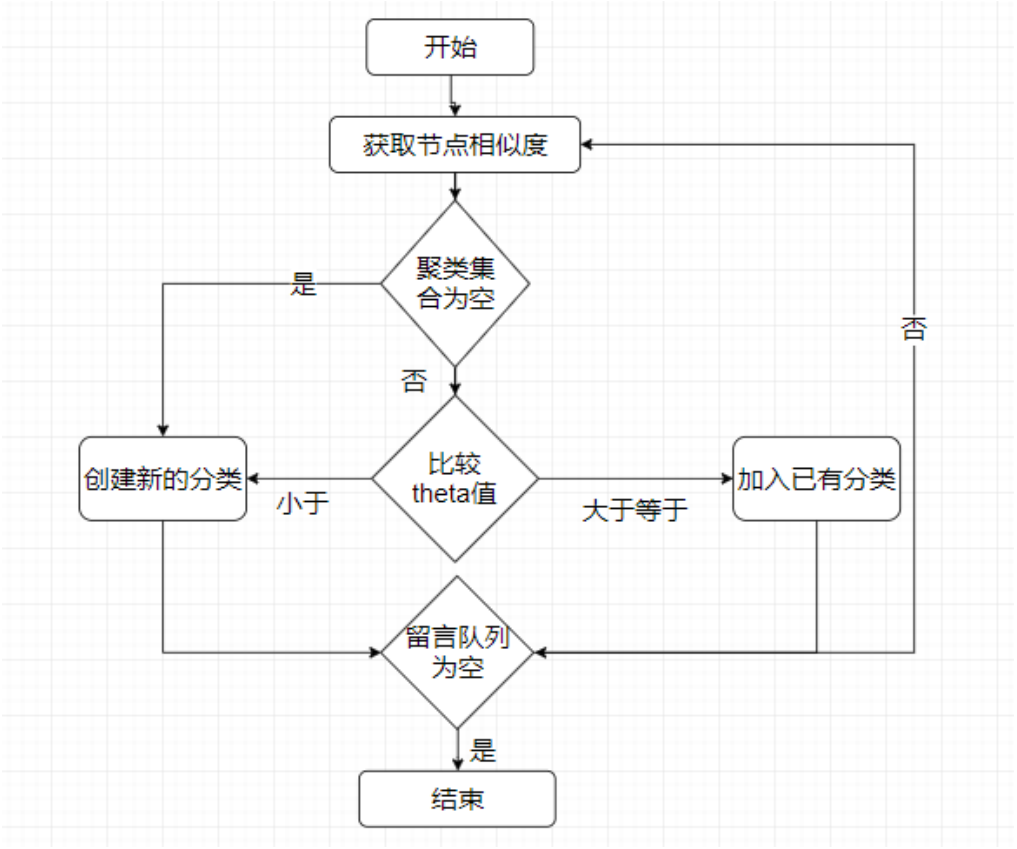


图 17：single-pass 原理流程图

大致算法处理过程如下：

（1）以第一条留言为种子，建立一个主题

（2）基于词向量模型将留言文本向量化，其中词向量模型替换成了我们自己训练得出的模型

（3）将留言文本与已有的所有话题均做相似度计算，采用了余弦距离来计算相似度

（4）找出与该留言文本具有最大相似度的已有主题

（5）若相似度值大于阈值 theta，则将文本加入到该类别之中，跳转到步骤 7

（6）若相似度值小于阈值 theta，则表明该留言不属于现有的所有主题，需就该留言新建一个主题类别，将当前留言归为该类

（7）聚类结束，导入下一条留言进行聚类

3.3 评价模型

3.3.1 评价指标

（1）留言数

考虑到系统主要是为政府查找出群众集中反映的某一热点问题，秉承着公平公正、平等可信的原则，防止控评等现象发生，我们选择群众的留言数作为主要热度评判指标，当多个留言都涉及到同一个问题时，说明这个问题影响很大、需要着重关注优先解决，而且由于存在多个群众反映，可信度高，因此它的热度会较之同一时段的其他问题会偏高。

（2）点赞数和反对数

点赞数和反对数也是影响热度评价的因素，但是他们在评价模型的权重较低，主要是考虑到可能会存在留言者拥有多个支持者或攻击者，从而导致名人效应，造成留言的点赞数或反对数虚高，出现某一不存在或无关紧要的问题是热点问题的假象。因此，我们降低了点赞数和反对数的权重占比。

（3）时间

在实际生活中，时效性是资讯热度的重要指标，新出现的问题往往是待解决的问题，时间久远的问题往往已经被解决。参照各大咨询网站，信息热度都会随时间的拉长而降低。因此，我们在评价模型中加入了时间的影响，热度会随问题时间的拉长而衰减。由于问题不可能一提出就得到解决，需要政府调查核实，因此问题的时间跨度往往较长，所以我们采用天数作为时间差的单位。为避免异常值，我们选用 3/4 位置的留言时间作为热点问题时间。

3.3.2、点赞量数据处理

在整理分析数据集时我们发现，绝大部分留言的点赞量都是处于 100 以下，只有极少数留言的点赞量超过了 100：

	留言编号	留言用户	留言主题	留言时间	留言详情	反对数	点赞数
2	208636	A00077171	汇金路五矿万境K9县存在一	2019/8/19 11:34:04	狗咬人，请问有人对	0	2097
3	223297	A00087522	反映A市金毛湾配套入学的问	2019/4/11 21:02:44	纳入配套入学，A3	5	1762
4	220711	A00031682	请书记关注A市A4区58车贷案	2019/2/21 18:45:14	案情消息总是失望，	0	821
5	217032	A00056543	A市58车贷特大集资诈骗案保	2019/2/25 9:58:37	股东、苏纳弟弟苏	0	790
6	194343	A000106161	A市58车贷案警官应跟进关注	2019/3/1 22:12:30	侦并没有跟进市领	0	733
7	263672	A00041448	小区距长赣高铁最近只有30米	2019/9/5 13:06:55	复到我如下问题：1	0	669
8	193091	A00097965	绿物业丽发新城强行断业主	2019/6/19 23:28:27	提供地摊上买的收据	0	242
9	284571	A00074795	省尽快外迁京港澳高速城区	2019/1/10 15:01:26	、长浏高速出口，这	0	80
10	200667	A00079480	要把和包支付作为任务而不让	2019/1/16 17:01:25	层工作者也不理解，	0	78
11	262052	A00072424	月亮岛路沿线架设110kv高压	2019/3/26 14:33:47	上电力线路，应采	0	78
12	226723	A00040222	一大道全线快速化改造何时	2019/9/15 15:31:19	改造，打通机场北通	0	66
13	281898	A00096623	号云时代多栋房子现裂缝，质	2019/2/25 15:17:38	检站处理，陷入了与	5	55
14	272089	A00061602	A6区月亮岛路110kv高压线的	2019/4/9 17:10:01	地省体操学校、西地	2	55
15	239595	A00057814	回东六路恒天九五工厂地块，	2019/11/8 15:48:07	地区，这里潜力巨大	0	44

图 18：原始数据点赞数异常值

上图是我们在全部数据的 excel 表中对点赞量进行降序排列得出的结果，可以看到留言 1 到 7 的点赞数目很大，各条留言之间点赞量相差极大，而且这部分留言在全部数据中占比不足 0.05%，除去这几条留言，其他留言的点赞量都是处于 100 以下。

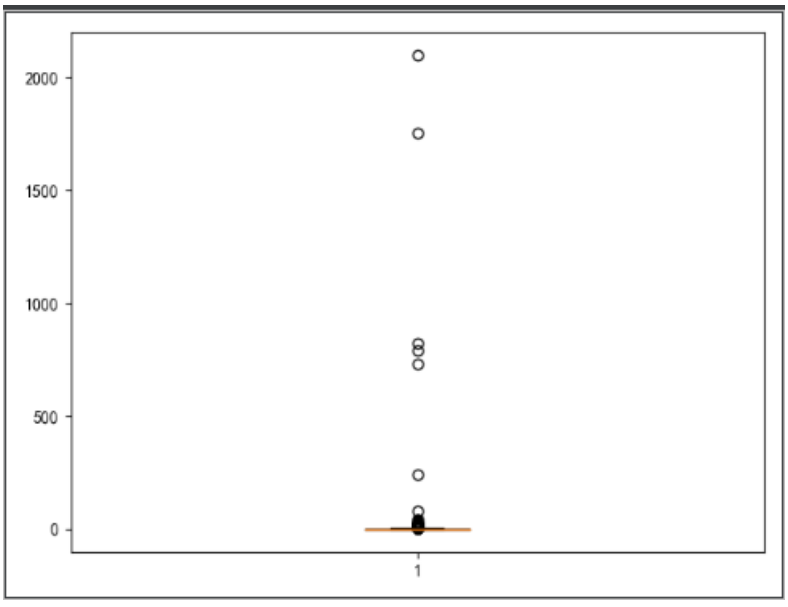


图 19：点赞数箱线图

通过聚类得出这几类问题的留言量，我们发现这些问题的留言量和点赞数并不成正比，因此我们判定这些超过 100 的点赞量是异常数据，由于在评价模型中加入了点赞量和反对量的影响，这些异常数据会对最终结果造成很大的影响，因此我们该异常数据进行了处理，利用下列函数将异常点赞量适当的缩小，但同时保留他们原始点赞量较多的特征：

$$F(x) = \ln(x) * 20$$

x：*异常值*，$F(x)$：*替换值*

3.3.3 热度计算公式

$$\text{Flag} = \frac{\text{like} - \text{oppose}}{|\text{like} - \text{oppose}|}$$

$$\text{LikeOppose} = |\text{like} - \text{oppose}|^{a}$$

$$\text{DateNum} = \text{nowTime} - \frac{3}{4} * \text{commentTime}$$

$$\text{Score} = \frac{(\text{Flag} * \text{LikeOppose} + \text{commentNum}) * \text{b}}{(\frac{\text{DateNum}}{\text{c}})^{d}}$$

$Flag$：*符号变量*，$like$：*点赞数*，$nowTime$：*当前时间*

$commentTime$：*留言时间*，$commentNum$：*留言数*

a，b，c，d：*模型参数*

计算模型中综合考虑了留言数（commentNum）、点赞数(like)、反对数(oppose)和时间（Date_num）等指标，同时给他们设定了指定的参数(0<a<1, b, c>1, d>1)，这些参数可根据实际情况变动。点赞数和反对数通过加入指数减低权重，时间也通过指数来控制对评分的影响，时间差越大热度消减的速度越快。此评价公式参考了 Reddit 热度计算算法[9](Score = $\log_{10}$z+y*t/4500)和英语魔方秀网站的热度计算算法(Score = (总赞数*0.7+总评论数*0.3)*1000/(发布时间距离当前时间的小时差+2)^1.2)。

3.3.4、参数设置

通过分析现有数据，多次实验，我们对热度计算公式中相关的参数进行了以下设定：

$$\text{a} = 0.5\,,\text{b} = 1000\,,\text{c} = 10\,,\text{d} = 1.1$$

3.4、模型优化

3.4.1 词向量模型替换（word2vec-bert）

由于文本聚类等操作需要将文本进行向量化，因此影响该模型的准确率的一个主要因素便是词向量的选择。一开始我们采用的是 word2vec 对文本进行预训练得出相应的句向量，但是将其应用于文本聚类的效果并不佳。之后我们改用 Bert 模型来生成句向量，主要原因是 Bert 的预训练数据 Bert_base chinese_L-12_H-768_A-12 是谷歌使用巨大的训练的语料，花费很多的时间和目前已经得到证明有效的 deep bi-directional transformer 模型。训练得出的部分结果如下：

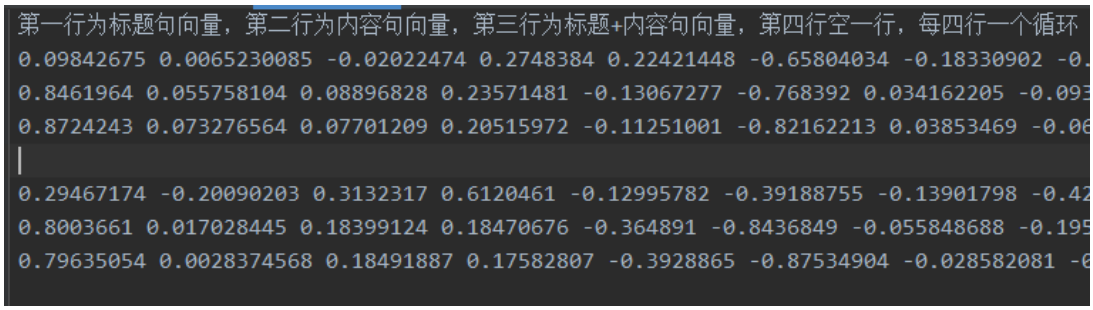


图 20：bert 词向量示例图

通过将 Bert 模型训练得出的句向量替换原来的 word2vec，留言聚类的效果得到了很大的提升，并且我们发现：将标题+内容词向量加入训练可以明显发现分类效果更佳，模型得到的分类留言相似度更高。

3.4.2、theta 值优化

在 3.2.6 中我们解释了 theta 值的重要性并给出了我们的确定方案，主要是通过可视化相似度数据来认为确定 theta 值的大小，但是这种方法不能很好的适应多种相似度值的情况，例如在更换了相似度计算算法之后，相似度矩阵数据的平均值和方差都会随之改变，导致出现原有的 theta 值效果一般的情况，所以在后续的模型迭代中，我们希望能够给出一种普遍适用的 theta 取值方案：我们认为在相似度计算过程中，假如随机抽取两种表述内容不一致的言论时，生成的相似度结果会偏小，甚至接近于 0，而两篇相似的留言相似度则会明显区别于不一致内容的相似度，即两者的分化程度偏大，所以我们可以认为在足够数据量的情况

下，相似留言的阈值接近于异常值，这一观点可以从相似度可视化图中的曲面凸起中得到验证，所以，我们在对比了几种异常值计算方法之后，选用了比较合适可行的箱型图计算方法作为我们的动态 theta 取值解决方案，计算 theta 值函数为：

$$theta = 上四分位数 + (上四分位数 - 下四分位数) * 1.5$$

在 3.2.6 的相同情况下，使用箱型图异常值检测出的 theta 值为：theta=0.1438，对比我们由人工确定的 theta 值差异不大，所以我们可以认为，这是一种有效可行的动态阈值方案。

3.5、结果分析

采用以上模型对留言进行聚类，再通过热度评价模型计算热度并进行排序，最终得出排名前 5 的热点问题：

热度排名	问题 ID	热度指数	时间范围	地点/人群	问题描述
1	33	2834	2019/07/03 至 2020/01/26	A 市 A2 区丽发新城小区	丽发新城小区违建搅拌站，彻夜施工扰民污染环境
2	32	2574	2019/07/02 至 2019/09/01	A 市伊景园滨河苑	广铁集团违法捆绑销售车位
3	27	2036	2019/01/02 至 2019/12/25	A 市公积金影响人群	A 市公积金政策的疑问及建议
4	356	1984	2019/03/02 至 2019/12/15	麻将馆	麻将馆扰民严重
5	244	1926	2019/12/15 至 2020/01/03	A 市 A1 区老地方美食广场	A1 区老地方美食广场没有消防证却能正常营业，仍虚假招商骗取商户上千万

表 2：热点问题表

统计现有数据，我们发现评分排名前 5 的热点问题普遍留言量和点赞量都很多，而且比较接近当前时间，这也与我们定义的评价模型相符。通过命名实体识别得出的特定地点或特定人群也比较符合热点问题留言的信息。不过由于留言中的特定地点或人群是虚构的，命名实体识别无法准确的提取出全部的地点和人群，加上聚类算法无法做到百分百的准确等多种原因，造成一类热点问题中还是会存在一些信息不匹配的留言。

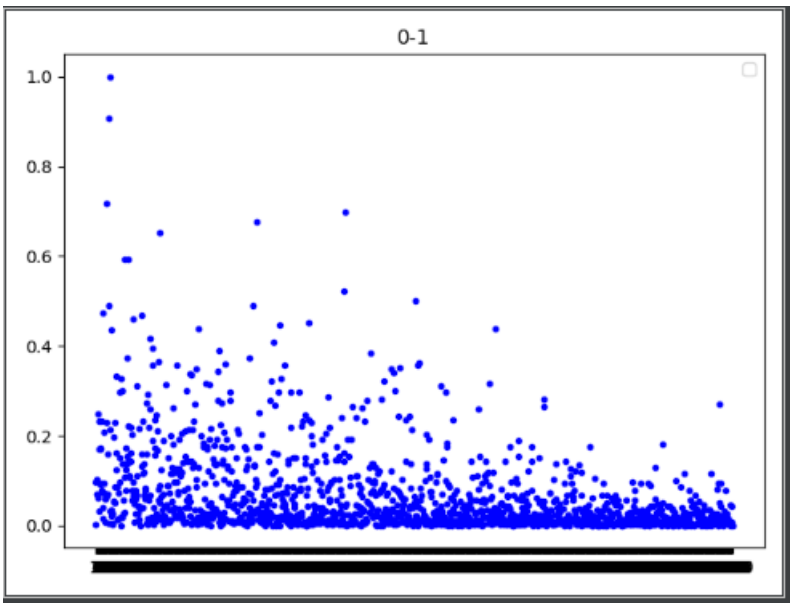


图 21：热度值 0-1 归一化的结果图

从图中可以看出，聚类得出的热点问题种类比较多，这主要是由于大部分留言的相关性太低，无法聚成一个类，导致很多类都是由一条或几条留言组成。而且大部分问题热度很低，这其中的原因有：该热点问题留言量、点赞量太少，问题距离当前时间太久远等。但也有一部分评分很高，这些问题就是排名靠前的热点问题。

4、问题三

4.1、问题分析

问题三是针对相关部门的留言答复进行质量评价，影响因素包括相关性、完整性和可解释性。相关性用于检测答复意见是否符合问题述求，完整性用于判定答复意见是否符合某种规范，可解释性用于查找答复意见中内容的相关解释。考

虑到答复意见主要是针对问题进行回复，所以在评价模型中我们主要以相关性为主，完整性和可解释性的权重会略低。通过为每类指标建立相应的打分体系，利用层次分析相结合的权重赋值法，对每个答复留言进行比较、分析和评分。秉持差异性和鼓励性相结合的原则，我们还会对初次的评分结果进行优化，使最终的评分更加人性化，形成最终的答复意见质量评价模型。

4.2、相关性分析

相关性分析是计算评论和回复的关联度，简单讲就是判断回复是否是答非所问，回复内容与评论的点对应时，这个回复的相关性得分就越高。计算句子相似性我们使用 Bert_base 预训练模型加 Lcqmc 数据集训练完成。Laqmc 数据是哈工大转为实现文本相似度预测生成口语化描述的数据集，包含训练、验证和测试集，训练集包括 24 万口语化描述的中文句子对，标签为 1 或 0 分别表示语义相似和不相似。

经过 GPU 训练了三个小时的训练模型用于测试后，得到如下的相似度预测，部分数据截图如下：

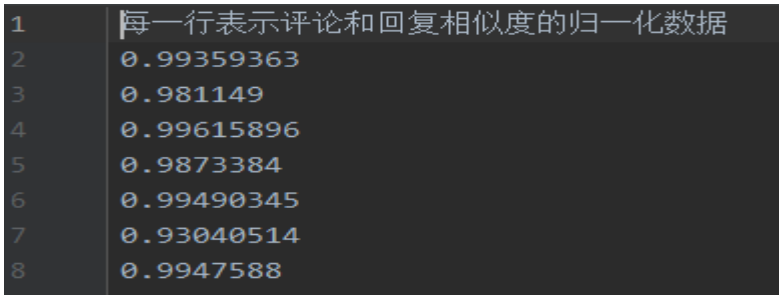


图 22：相似度结果图

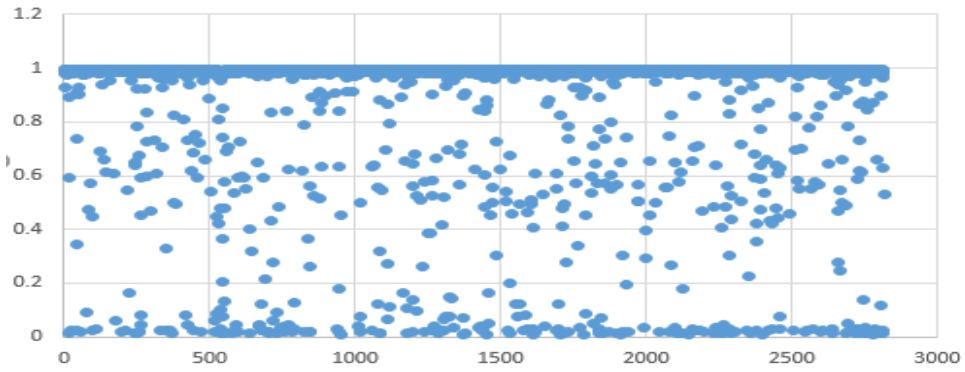


图 23：相关性评分散点图

通过对结果和回复内容的人工对比，基本上是符合相关性要求，但从图的结果明显可以看出相似度呈现两极分化的现象，通过我们对数据分析，找出原因如下：

（1）我们模型相关性的分数只与回复的讨论话题是否与评论一致想关系

（2）大部分的回复都答到问题的点上，自然取得的分数就高

（3）存在一些很敷衍的回复，如“该情况已经交到相关部门处理”，这种回复评分很低。

相关性得分是整个回复的基础得分，这样的数据越能体现出不同评论的优劣差异。

4.3、完整性分析

题目中完整性指的是判断答复是否满足某种规范，通过分析现有数据，我们发现答复主要由以下几个部分组成：问候语、留言接收情况、留言回复、感谢语和回复时间。因此，我们根据现有答复数据制定了留言答复规范，通过正则匹配将数据集中的留言与制定的规范一一比较，得出答复的完整性得分。留言答复规格大致如下：

xxx：你好（相关问候语）！xxxxxx 留言已收悉（留言接收情况相关词语）。xxxxxx 函复如下（答复相关词语）：xxxxxx。感谢（相关感谢语）xxxxxx。回复日期。

此外，我们给每一部分分配了不同的权重，分别是 15，10，50，15，10。通过将每一部分的得分相加得出每一条留言答复的初评分。

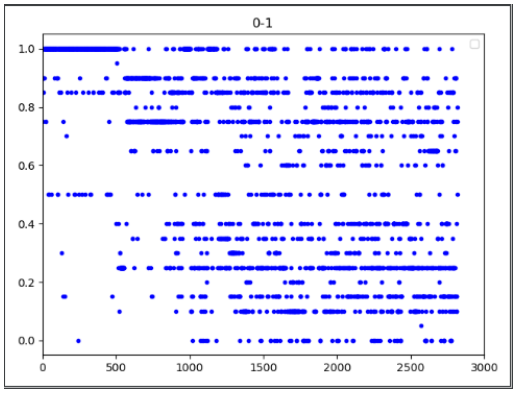


图 24：零规范化分布图

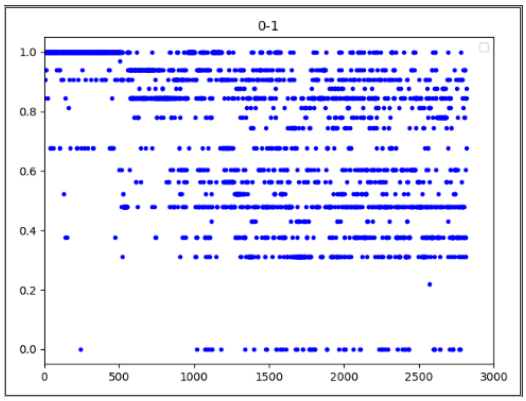


图 25：优化后分布图

从零规划分布图可看出得分情况较为均匀，但有将近一半的数据处于低评分区域，而且与高评分存在较大的分差。考虑到在实际生活中人们更关注的是问题

答复是否满足问题述求，而较少去关注留言答复的规格。所以我们引入了一条自定义函数对评分进行优化：

$$\mathrm{F(x)} = (log_{10}^{(x+1)} * 50 + x)/2$$

x：*初评分*

优化后的评分分布如上图所示，大部分留言回复的得分都处于可观的区域，同时保留了得分之间的差异性，达到了我们的预期效果。

4.4、可解释性分析

在分析一个答复可解释性效果的时候，我们认为一个可解释性良好的答复应该具备答复缘由，引用文件，文件出处等，所以，我们打算从法律文件，政府公示等文件格式入手，统计答复内容中涉及到的法律词条，政府文件等规划出了一个基于统计的可解释性分析模型，其中涉及的评分模型我们以统计词条结果作为各答复可解释性的划分依据，引入自定义函数将初评分规划映射到期望区间和达到控制方差的效果。

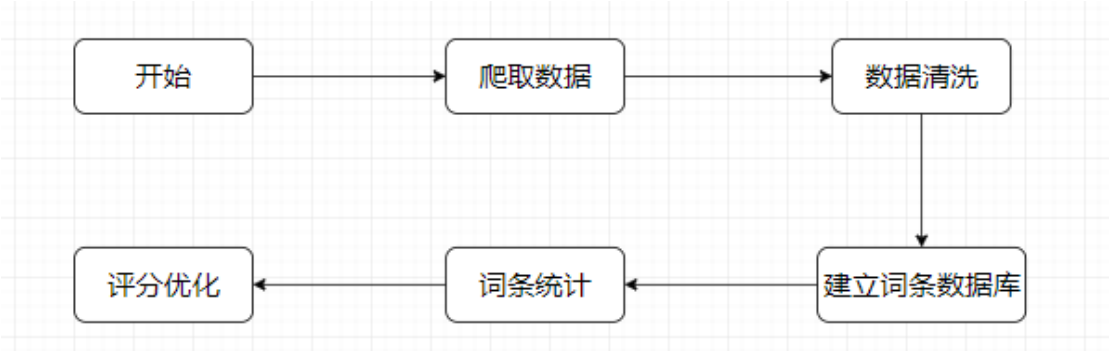


图 26：可解释性评分模型流程图

（1）建立此词条数据库，我们找到了中华人民共和国司法部法律法规数据库网站[10]作为我们的词条数据库来源，并以‘小区’，‘纠纷’，‘举报’，‘管理方法’等为搜索关键词，使用 python 爬虫对这些关键词批量搜索，将搜索结果进行清理掉停用词，低价值词条之后建立我们可解释性的词条数据库。

（2）词条统计，对于词条数据库，我们通过将答复结果进行分词之后得到的词条数据与词条数据库中的词条进行比对，使用 word2vec 模型对两者的相关性打分，最终得到每条留言的初评分数据。

（3）评分优化，为了可解释性与评价答复的其他指标进行加权统计，我们对留言的初评分数据做了零规范化处理，稀释了由初评分值差异带来的权重偏离影响，此外，在零规范化的基础上，我们希望我们的评分结果更加的人性化，在体现差异性的同时避免数据出现分化的情况，所以我们引入了一条自定义函数来对数据进行二次处理：

$$F(x) = \frac{\ln(100*x+1)}{4.61}s$$

公式中出现的分母 4.61 为 ln(100)的近似值，目的在于使得最终生成得数据可以落入（0，1）区间中，不会和零规范化的初衷冲突。我们做出了两张二维散点图来对比引入自定义函数前后的数据分布情况：

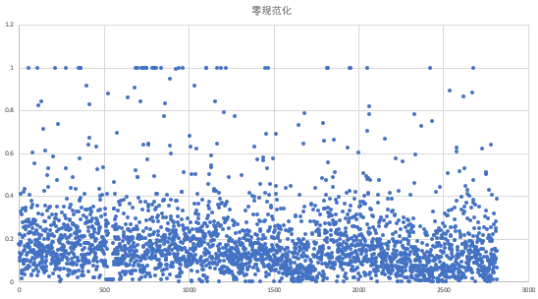


图 27：零规范化分布图

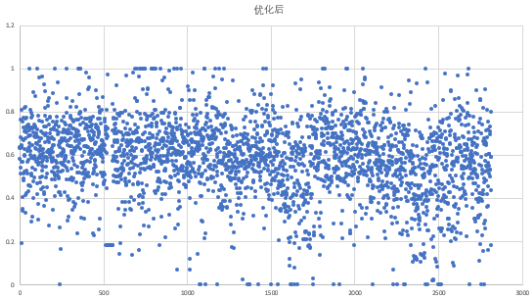


图 28：优化后分布图

从两张散点图中可以看出，经过评分优化前的分数值多分布于（0，0.2）区间内，个别值出现分化现象，显然与我们的评分理念不相符，函数优化后的数据分布情况较为均匀，点的分布区间较为可观，且由于该优化函数的连续性特性，点之间的差异性得以保留，在不改变差异性的同时优化了评分结果的分布。

4.5、结果分析

根据比例值 6:2:2，将相关性评分，完整性评分，可解释性评分整合为最终评分。部分评分结果如图：

通过对结果建立散点图，可以看出结果能够体现出不同的数据的差别，能够有效区分不同评分回复的优劣差异。进行部分人工审查，结果和人工对回复的评价大致符合，如一些“您好，你所反映的问题已转交相关单位调查处置。”类似较为敷衍的回复，模型测得的结果则很低：

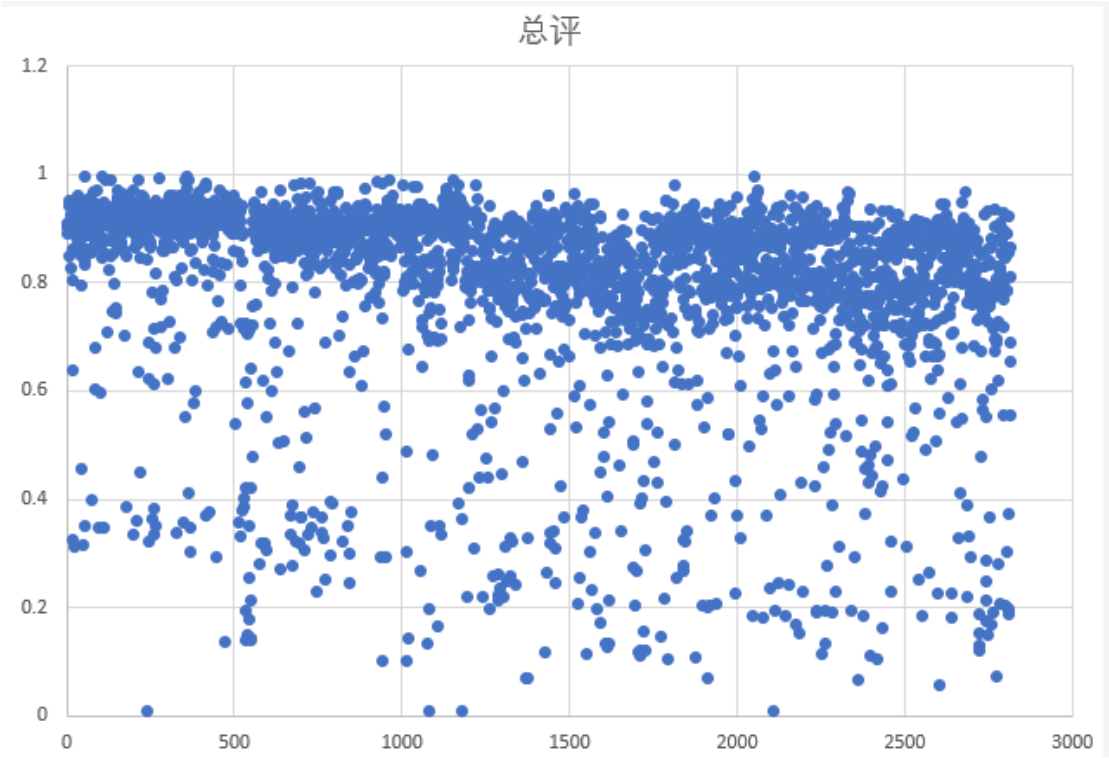


图 29：答复评分分布图

5、参考文献

[1]:arXiv,BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[2019-5-24] {jacobdevlin,mingweichang,kentonl,kristout}@google.com
[2] 奇点机智, NLP 必读 | 十分钟读懂谷歌 BERT 模型
https://www.jianshu.com/p/4dbdb5ab959b?from=singlemessage
[3] 简书，Bert 模型 tensorflow 源码解析（详解 transformer encoder 数据运算）
https://www.jianshu.com/p/2a3872148766
[4] 毛郁欣，邱智学，基于 Word2Vec 模型和 K-Means 算法的信息技术文档聚类研究，浙江工商大学管理工程与电子商务学院，2020-04-15
[5]母玉雪,张晓滨.参与式感知下环境评价指标权重确定方法[J/OL].西安工程大学学报:1-6[2020-05-05].http://kns.cnki.net/kcms/detail/61.1471.n.20200408.0905.012.html.
[6]阮一峰，TF-IDF 与余弦相似性的应用（三）：自动摘要
http://www.ruanyifeng.com/blog/2013/03/automatic_summarization.html
[7] 黄建一，李建江，王铮，方明哲，基于上下文相似度矩阵的 Single -Pass 短文本聚类，北京科技大学计算机与通信工程学院，2019-04-008
[8] 石春丹，秦岭，基于 BGRU-CRF 的中文命名实体识别方法，南京工业大学计算机科学与技术学院，2019-09-15
[9]阮一峰，基于用户投票的排名算法（二）：Reddit
http://www.ruanyifeng.com/blog/2012/03/ranking_algorithm_reddit.html
[10] 中华人民共和国司法部 司法部信息中心法律法规数据库 [2017] http://search.chinalaw.gov.cn/search2.html