“智慧政务”中的文本挖掘应用

摘要

随着社交方式的多样化，政府获取社情民意的文本数据量增加，随之而来的是留言分类和热点整理问题，因此我们建立了基于 NLP 技术的智慧型政务系统。

针对问题一，本文建立了一个基于卷积神经网络的中文文本多分类模型，首先将文本经过字符级嵌入形成的句子表示经过卷积层、池化层、全连接层，然后采用 softmax 函数产生属于每一个类别的概率。最后本文将赛题提供的数据进行切分，随机选取 70%的数据用于模型训练，20%的数据用于模型验证，10%的数据用于模型测试，结果显示在测试集上模型准确率、F-Score 分别达到了 0.86、0.85。但是由于 TextCNN 模型忽略了句子的上下文信息，导致模型没有取得很好的分类结果，因此我们对模型进行改进，建立了 BertCNN 模型，最终在测试集上取得了理想的效果，准确率和 F-Score 均达到了 0.94。

针对问题二，本文首先对数据进行数据清洗、分词及去停用词、长文本转化为短文本等数据预处理操作，然后分别采用基于 tf-idf 向量空间模型和卷积神经网络对句子进行特征表示，最后采用 k-means 算法对反应特定地点和特定人群问题的留言进行聚类，并采用 PCA 算法进行评估，结果显示两种特征表示模型均取得了较好的效果。在此基础上，本文根据反应特定问题的留言数量、点赞数及反对数和时间定义了合理热度评价指标，并且按问题要求格式给出了排名前五的热点问题。

针对问题三，本文分别对答复相关性、完整性、可解释性进行建模。针对答复的相关性，我们首先采用 tf-idf 模型提取出每个留言的 10 个关键词汇，然后综合起来建立词汇表，最后利用该词汇表对每个留言和对应的答复采用词袋模型（Bag-of-Words）进行编码，并采用余弦相似度计算得到答复的相关性结果为：很相关 35.76%、比较相关 46.88%、不相关 17.37%。针对答复的完整性，我们认为答复是由开头、主体、结尾三部分组成的，因此基于问题一提出的 TextCNN 模型对答复的三个部分分别训练三个不同的神经网络，并通过自行标注的数据完成对网络的训练，最后我们得到答复的完整性结果为：很完整 64.76%、比较完整 30.41%、不完整 4.82%。针对答复的可解释性，我们分别从答复的文本长度、答复和留言的相关性、信息量和因果关系四方面评价可解释性，最后得到答复可解释性的结果为：可解释 33.77%、较可解释 52.45%、不可解释 13.78%。最后我们综合相关性、完整性、可解释性给出了答复意见的质量。

关键词：卷积神经网络、BERT、tf-idf、LSA、simHash、k-means

Application of Text Mining in Smart Government Affairs

Abstract

With the diversification of social methods, the amount of text data obtained by the government for social conditions and public opinion has increased, and with it comes the problem of message classification and hotspot sorting, so we have established a smart government system based on NLP technology.

For question one, this paper establishes a Chinese text multi-classification model based on Convolutional Neural Network.First,sentence representation formed by text through character-level embedding pass through convolutional layer, pooling layer, fully connected layer,and then use softmax function to generate the probability of each category.Finally,this paper divides the data provided by the competition questions,and randomly select 70% of the data for model training,20% of the data for model validation, and 10% of the data for model testing.The results show that the model accuracy rate and F-score on the test set have reached 0.86 and 0.85.However,the TextCNN model ignores the context information of the sentence, and the model does not achieve good classification results.Therefore, we have established the BertCNN model, and achieved the desired effect on the test set:accuracy rate 0.94,F-score0.94.

For question two,this paper performs data preprocessing operations such as data cleaning, word segmentation and removal of stop words, and conversion of long text into short text.Then we use tf-idf vector space model and Convolutional Neural Network to characterize sentences.Finally we use k-means algorithm to cluster messages that reflect problems in specific locations and specific groups of people,and use the PCA algorithm for evaluation.The results show that both feature represent model has achieved good results.Based on the above,this paper defines reasonable evaluation indexes by the number of messages that respond to specific questions,the sum of likes and opposition and time,and then give the top five hot issues according to the format of the problem requirements.

For question three,this paper models the relevance, completeness, and explainability of responses.For the relevance,we first use the tf-idf model to extract 10 key words for each message,and synthesize a vocabulary.Finally,we use the vocabulary to encode each message and the corresponding responses by the Bag-of-Words model,and use the cosine similarity calculation to get the result of the responses relevance:very

relevant 35.76%, relatively relevant 46.88%, not relevant 17.37%.For the completeness,we believe that the responses consists of three parts: the beginning, the body, and the end.Therefore,based on the TextCNN model proposed in question one,we have trained three different neural networks for the three components of the responses,and complete the training of the network through the self-marked data. Finally we get the result of the responses completeness:very complete 64.76%, relatively complete 30.41%, incomplete 4.82%.For the explainability,we evaluate the explainability in terms of the length of the text of the response, the relevance between the response and the message, the amount of information, and causality. Finally,the results of explainable of response:explainable 33.77%, more explainable 52.45%, unexplainable 13.78% .Finally, we give the quality of the responses based on relevance, completeness and explainability.

Keywords: CNN, BERT, tf-idf, LSA, simHash, k-means

目录

一、问题重述

1.1 问题背景

近年来，政府部门开设了众多网络问政平台，在提高政府的管理水平和施政效率的同时，也带来了庞大的文本数据量，这给主要依靠人工来进行留言划分和热点整理的部门带来了极大挑战。

目前，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。

1.2 问题重述

问题一：根据附件 1 和附件 2 给出的标签和数据，建立关于留言内容的一级标签分类模型，并使用 F-Score、查准率、查全率对分类方法进行评价。

问题二：根据附件 3 将某一时段内反映特定地点或特定人群问题的留言进行归类，同时根据留言文本及特征，定义合理的热度评价指标，并按问题要求的格式给出相应的评价结果。

问题三：分析附件 4 相关职能部门对各类留言的答复意见，从答复的相关性、完整性、可解释性等角度出发，对答复意见的质量给出合理的评价指标。

二、问题分析

2.1 问题一的分析

要建立关于留言的一级标签分类模型，首先我们需要将文本内容转换为向量，然后建立多分类模型将留言分类。

为了能得到更好的分类效果，我们直接采用字符级编码将文本向量化，然后通过卷积神经网络提取主要特征，并通过 softmax 函数计算概率，最后通过反向传播算法优化模型参数。

2.2 问题二的分析

问题要求我们对所有的留言进行归类，首先我们需要对文本进行特征表示，然后选取聚类算法进行聚类，并定义合理的热度评价指标，最后按问题要求格式给出结果。

因为本问题是一个无监督问题，无法准确的评估模型效果，因此我们分别采用基于 tf-idf 的向量空间模型和基于卷积神经网络对文本进行特征表示，然后采用 k-means 算法进行聚类，并采用 PCA 降维显示聚类效果。最后根据合理的热度评价指标，给出热点问题的排名以及详细留言信息。

2.3 问题三的分析

问题要求从答复的相关性、完整性、可解释性等角度对留言回复的质量进行评价，对这些指标进行量化描述。

对于相关性，首先构建热点词汇表，然后对答复和留言进行特征表示，最后采用余弦相似度计算相关性。

对于完整性，可以从回复内容的结构完整性的角度出发，通过分析回复数据，将留言的回复是分为开头、主体、结尾三部分。对数据处理完成后，通过对应的模型对问题进行求解，得到回复内容的完整性。

对于可解释性，可以从是否具有依据或是否具有因果关系这两方面进行判断。通过对回复数据的分析，从留言与答复的相关性、回复文本长度、回复所含信息量以及是否含有因果关系等四个方面去评价回复的可解释性。

三、符号说明

符号类型	符号	符号说明
问题一	L	交叉熵—多分类问题的损失函数
	N	样本数量
	M	分类的类别数量
	y_{ic}	指示变量（0 或 1）
	p_{ic}	观测样本i属于类别c的预测概率
	M_j	卷积神经网络第 j 个特征面
	b	additive bias(偏执参数)
	f	relu 激活函数
问题二	$tf_{t,d}$	词语 t 对于文档 d 的词频
	$n_{t,d}$	词语 t 在文档 d 中的出现次数
	k	词汇表所有词语
	idf_t	某一特定词语 t 的逆文档频率
	N	所有文档的总数
	df_t	包含词语 t 的文档数目
	$w_{t,d}$	词语 t 在文档 d 中的重要程度
	τ	原始特征的维度
	$\phi(i)$	第 i 个原始特征的词频数值
	$\bar{\phi}(i)$	哈希后第 j 个特征的词频数值
	d_w	每一个词语的 embedding 维度
	s	句子的长度

	C	留言数量
	E	点赞数和反对数之和
	T	时间长短
问题三	x_i, y_i	留言和回复向量化之后的向量
	$\cos\theta$	余弦相似度
	f_1, f_2, f_3	三个网络输出的正标签的概率
	F	答复的完整性
	$H(X)$	信息熵—答复的信息浓度
	X	每一个答复
	x	答复分词之后的每一个词语
	q_1, q_2, q_3, q_4	四个类别上的分类概率
	Q	因果关系强弱
	S	答复的可解释性
	E	答复意见的质量
	C	答复与留言的相关性

四、模型建立及求解

4.1 问题一的模型建立及求解

针对问题一，因为问题需要我们对群众留言按照一定的划分体系进行分类，我们能将其看作一个文本多标签分类问题。对于多标签文本分类问题的求解，能将多标签文本分类问题划分为数个二分类文本分类问题，即对各个标签构建相应的模型，选取概率最高的标签作为预测分类结果。在传统机器学习方法中，通过提取 tf-id 或者词袋特征，然后可以使用 SVM、Naive Bayes 等模型进行训练。在深度学习领域中，TextCNN 模型[1]是基于卷积神经网络的文本分类模型，其在文本分类上有不错的表现。本文正是基于 TextCNN 模型构建相应留言内容的一级标签分类模型。

首先，我们在字符级上对文本进行编码，然后将形成的句子表示经由卷积神经网络的卷积层、池化层提取出文本的主要特征表示，然后经过全连接层，并通过 softmax 函数产生属于每一个类别的概率，模型采用反向传播算法进行训练，

并引入了 dropout 机制抑制过拟合现象。

本文首先介绍 TextCNN 的基本原理。

4.1.1 TextCNN 基本结构

TextCNN 是基于卷积神经网络，为解决文本分类问题而构建的模型。其利用多个 feature map 提取句子中的关键信息以获得更好的局部相关性，能很好的解决文本多分类问题。

TextCNN 基本结构为嵌入层、卷积层、池化层、全连接层以及输出层，其结构如图 1-1 所示：

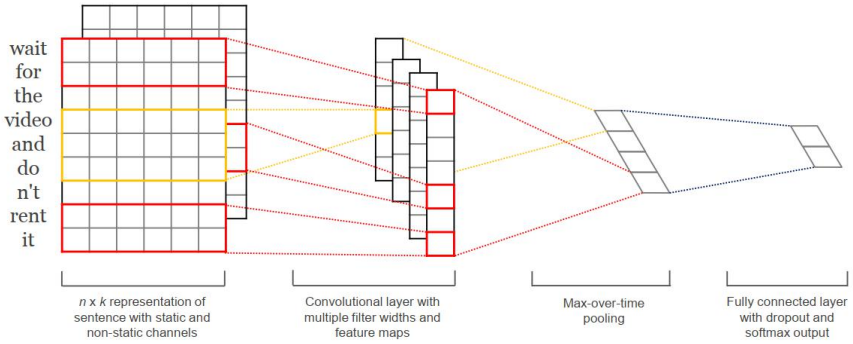

图 1-1　TextCNN 基本结构[1]

（1）输入层

数据在输入层前需对数据进行预处理，以符合输入层的输入需要。利用字符级嵌入，将每个字映射成 n 维向量，构建完向量后，将其拼接形成二维矩阵作为最初输入。

（2）卷积层

卷积层是卷积神经网络进行特征提取的核心模块，利用卷积层能够将数据在某一特征进行增强，减少噪声的影响。其利用梯度下降的方法，通过最小化损失函数对节点权重进行优化，提升模型对数据的分类能力。

卷积层通过卷积操作能够提取输入数据的不同特征。其第 l 层第 j 个特征面计算公式为：

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{1-1}$$

M_j 代表第 j 个特征面，b 代表 additive bias(偏执参数)，f 是 relu 激活函数

在卷积层后，我们添加 relu 函数进行激活，以提升模型的表达力。Relu 函数公式为

$$f(x) = \max(0, x) \qquad 0 < x \tag{1-2}$$

Relu 函数图像如图 1-2：

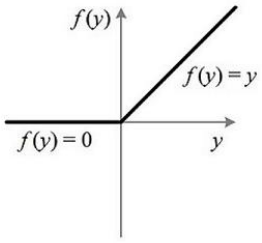

图 1-2　Relu 函数图

Relu 函数为分段线性函数，所有负值为 0，正值不变，因此具有单侧抑制的特点，ReLU 实现稀疏后的模型能够更好地挖掘相关特征，拟合训练数据。而且因为 Relu 非负区间的梯度为常数，因此不存在梯度消失的问题，使模型的收敛速度维持在一个稳定状态。

（3）池化层

池化层的作用是对数据进行降维，用更高层次的特征来表示数据，从而降低信息的冗余，提升模型的尺度不变性、旋转不变性以及防止过拟合。在我们的模型中，我们选择使用了最大池化方法，取局部接受域中值的最大值。

（4）全连接层

通过卷积以及池化，我们将原始数据映射到了隐层特征空间，通过全连接层我们能够将学习所得的特征映射到样本标记空间。在全连接层中，每一个节点都与上一层的所有节点相连，从而对上一层的特征进行综合，整合具有类别区分性的局部信息。但是由于模型参数过多，容易发生过拟合情况，故在全连接层后添加了 dropout 层。同时为提升卷积神经网络的性能，我们对每个神经元进行 relu 激活。最后为了输出分类结果，我们采用 Softmax 逻辑回归的方式进行分类。

（5）模型学习

最后对于分类任务，需要选定合适的损失函数来量化模型对样本数据的拟合程度，以及模型预测的分类结果和实际值，并以此来判断模型的好坏。交叉熵对于多分类问题，是一个合适的损失函数。

交叉熵函数

$$L=\frac{1}{N}\sum L_i=\frac{1}{N}\sum_i(-\sum_{c=1}^{M}y_{ic}\log(p_{ic})) \qquad (1\text{-}3)$$

M 类别的数量，y_{ic} 指示变量（0 或 1）,如果该类别和样本 i 的类别相同就是 1，否则是 0，p_{ic} 对于观测样本 i 属于类别 c 的预测概率。

当模型拟合能力越强，预测结果与实际结果越接近，则交叉熵的数值越小，反之则模型的交叉熵越大。

4.1.2 算法求解结果

（1）TextCNN 模型训练

本次实验中，我们采用 tensorflow 深度学习框架来构建 TextCNN 模型。

在数据划分中，随机截取其中 70%作为训练集，20%作为验证集，10%作为测试集。同时通过计算各单词在训练集出现的频数，我们选择最频繁出现的 5000 个单词构成词汇表。

在嵌入层中，利用我们构建的词汇表，将数据集内的各个词都映射成词向量，并将各个词向量按行整合作为卷积层的输入。

在卷积层中，利用 tensorflow 中的 layers.conv1d()构建了含有 256 个尺寸为 5 的卷积核的一维卷积层 conv，其只在纵向上滑动。在卷积层后，添加池化层 gmp，采用最大池化操作，对领域的特征值取最大操作，保留最大特征，减少模型参数数量，避免模型出现过拟合现象。

在全连接层中，定义神经元数为 128 全连接层 fc1，并在其后添加一层 dropout 层，定义 dropout 的保留概率为 0.5。其能降低网络的耦合度，使模型更多的学习重要的特征。最后，用 relu 进行激活并将结果输入输出层。

在输出层中，利用 softmax 函数进行结果的分类判断，并将判定结果进行输出。

此外，采用 Adam 算法调整训练过程中的学习率，其能通过计算梯度的一阶矩估计和二阶矩估计而为不同的参数设计独立的自适应性学习率，这样能避免学习率对模型结果带来不良影响。

我们利用 tensorflow 的 tensorboard 模块，输出实验中构建的 Textcnn 模型的结构，如图 1-3 所示：

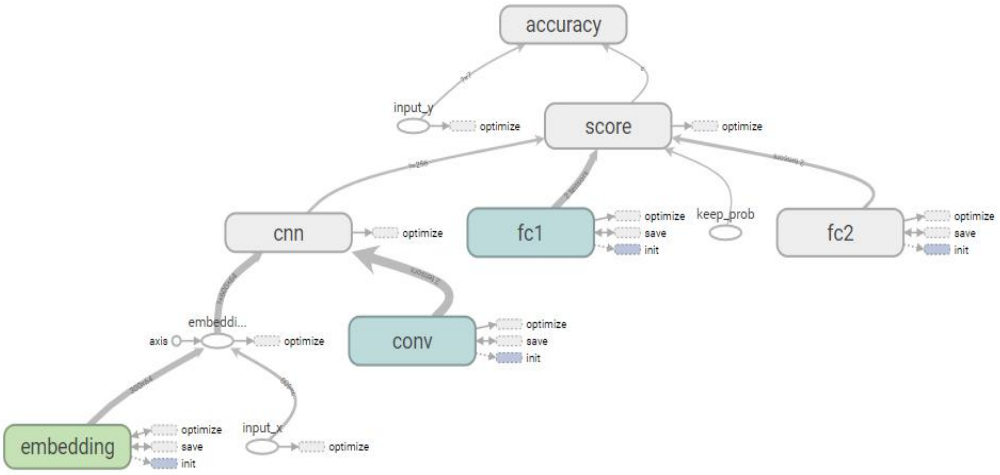

图 1-3　模型训练流程图

（2）测试结果

我们对样本预定进行 100 个回合训练，每个回合迭代 64 次，每迭代五次记录准确度以及损失函数情况，同时记录卷积层、全连接层的权重值以及偏执量的更新情况。

利用 tensorboard 中的 scalars 绘制平滑度为 0.8 的验证集准确度以及训练过程的损失函数变化情况，如图 1-4、图 1-5 所示：

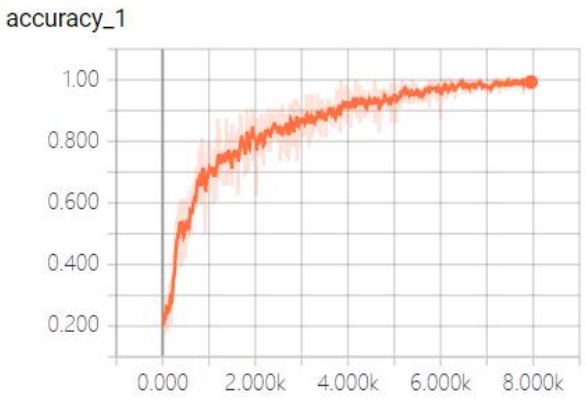

图 1-4　准确度变化情况

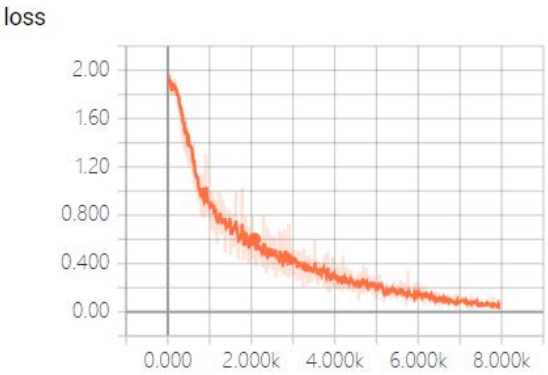

图 1-5　损失函数变化情况

利用 tensorboard 中的 histograms 绘制卷积层以及全连接层权与偏置量变化情况，如图 1-6、图 1-7、图 1-8 所示：

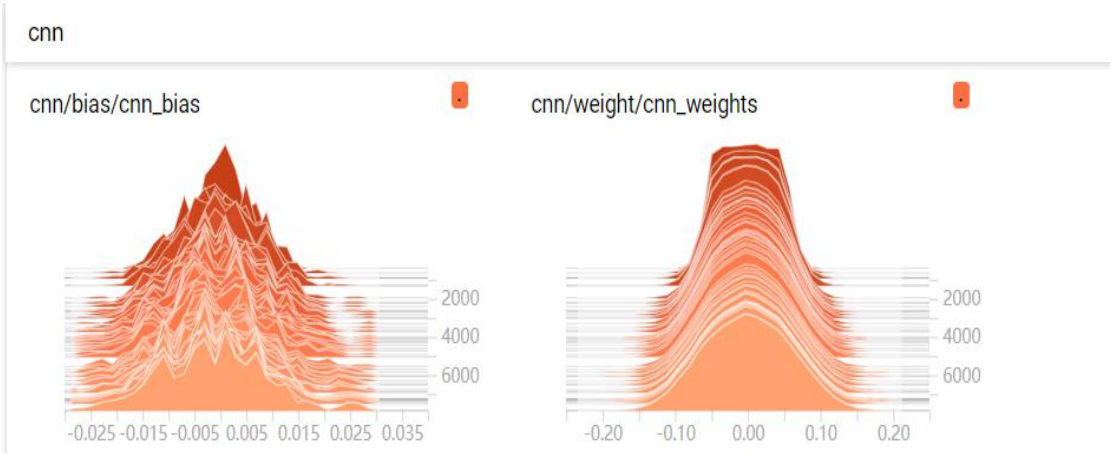

图 1-6　卷积层权重与偏置

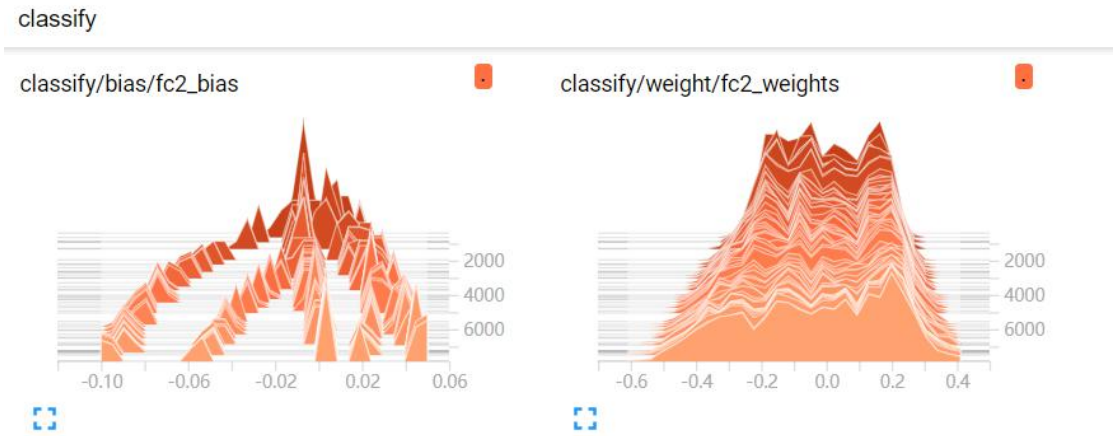

图 1-7　全连接层权重与偏置

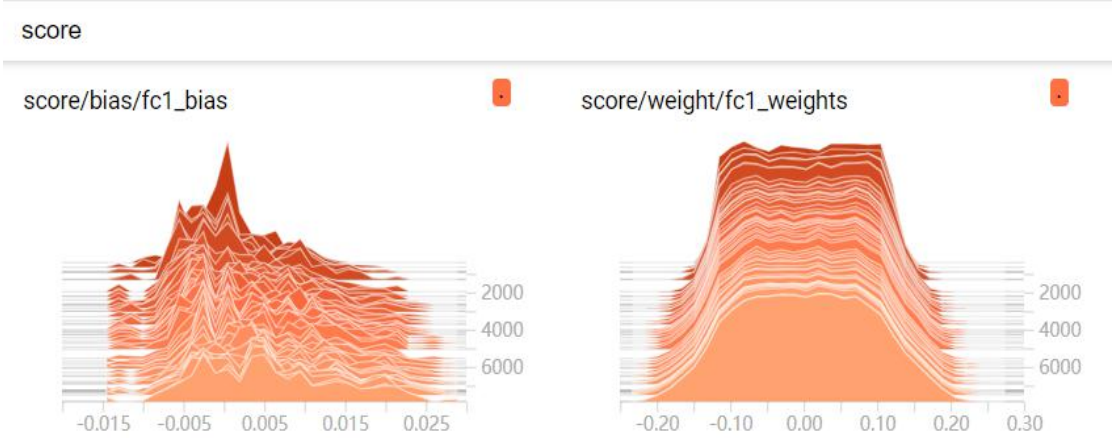

图 1-8　softmax 层分类权重与偏置

利用训练完成的模型对我们的测试集进行测试，将查全率、查准率以及 F-Score 进行了输出，并以各标签的结果的加权平均作为最后的模型判断结果，结果如表 1-1 所示：

表 1-1　一级标签分类指标测试结果

	precision	recall	F1-score
城乡建设	0.81	0.82	0.81
环境保护	0.86	0.89	0.88
交通运输	0.83	0.68	0.75
教育文体	0.93	0.96	0.94
劳动和社会保障	0.91	0.93	0.92
商贸旅游	0.76	0.74	0.75
卫生计生	0.90	0.86	0.88
模型结果	0.86	0.84	0.85

模型在测试集数据的检验结果有 85.71%的准确度，模型的 F-Score 达到了 0.85，模型预测效果好。

4.1.3 模型改进

虽然 TextCNN 取得了较好的效果，但是其中的卷积、池化操作丢失了文本序列中的词汇的顺序、位置信息，因此难以捕获文本序列中的否定、反义等语义信息，达不到理想的效果。

在自然语言处理任务中，预训练的语言模型已经被证明可以提升各种任务的效果，在本问题中，word embedding 及模型的所有参数都是基于该问题的数据训练得出的，因此，为了进一步提高模型的效果，本文引入预训练的语言模型 Bert[2]。

Bert 是由 Google 团队提出的，其可以根据具体的下游任务进行参数微调。本问题使用了 Bert-CNN 模型[3]，将 Bert 看作编码器，将留言映射成文档向量，在其上叠加更加复杂的卷积神经网络。

4.1.3.1 BERT-CNN 模型结构

本文基于 BERT 预训练语言模型，提取其最顶层的 transformer 的输出作为留言的文档向量，与卷积神经网络进行联合训练。

（1）BERT 层

BERT 的英文全称是 Bidirectional Encoder Representations from Transformers 采用双向 Transformer 编码器[4]，利用多头注意力机制融合了字左右的上下文信息。

BERT 模型的详细结构见图 1-9，由多层双向 Transformer 构成，每个 Transformer 利用多头自注意力机制 Multi-Head Attention 建立词与词之间的联系。谷歌开源了两种不同规模的 BERT 模型，分别为 BERT-Base 和 BERT-Large，考虑到计算资源，我们使用了 BERT-Base-Chinese 模型。

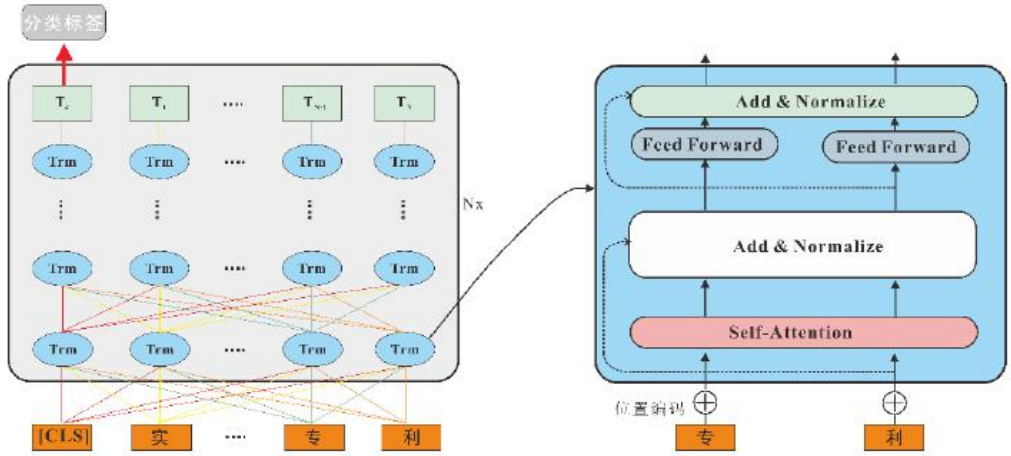

图 1-9　BERT 模型的详细结构[3]

（2）BERT-CNN 结构

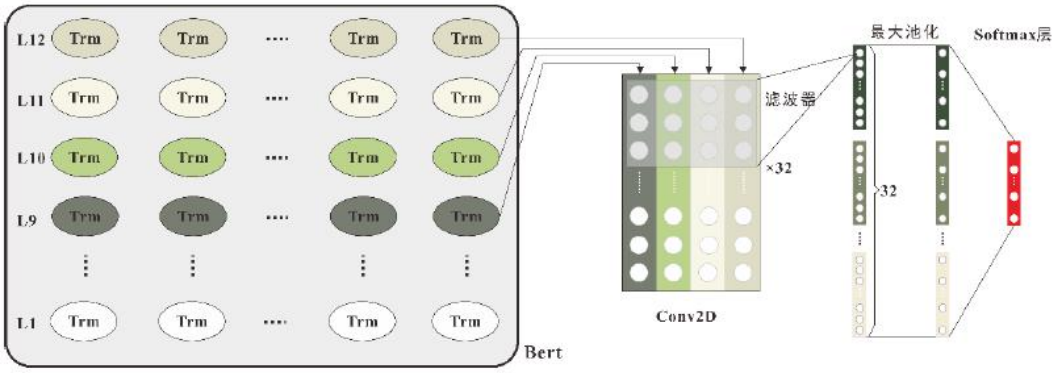

图 1-10　BERT-CNN 模型总体结构[3]

BERT-CNN 采用 BERT 中的最后一个 Transformer 层的输出作为下游 CNN 模型的输入，模型总体结构见图 1-10。

4.1.3.2 模型训练

本问题采用 pytorch 深度学习框架，GPU 显卡型号为 GeForce GTX TITAN X，共 256GB 内存。

BERT-CNN 模型使用 BERT-Base-Chines 模型，学习率为 2e-5，batch_size 为 64，最大文本长度 L 为 128，Epoch 为 60，优化器为 Adam。BERT-CNN 中的 CNN 滤波器个数为 200，激活函数为 ReLU，训练模型时采用 gradient accumulation 方法。

本文使用 tensorboardX 库对训练过程可视化，训练过程中生成的训练集和验证集的 F-Score、训练集 loss 及训练集和验证集的准确率见图 1-11、图 1-12、图 1-13：

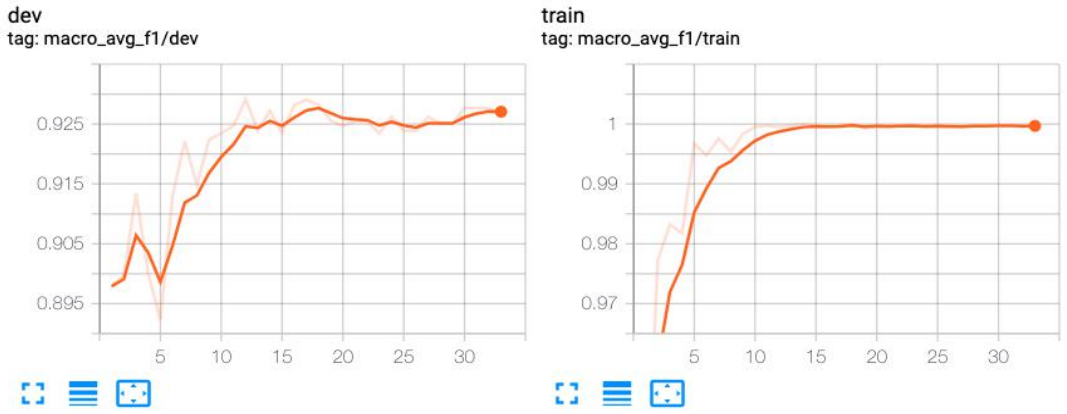

图 1-11　训练集 F-Score 与验证集 F-Score

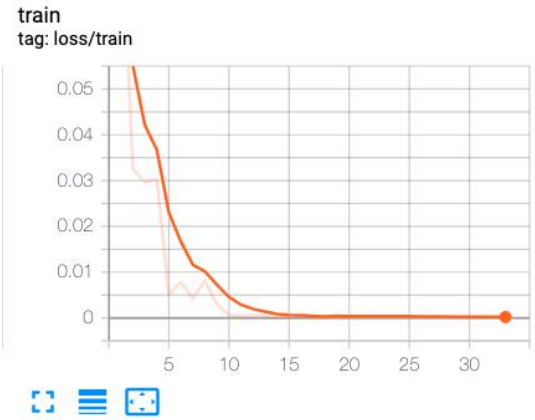

图 1-12　训练集 loss

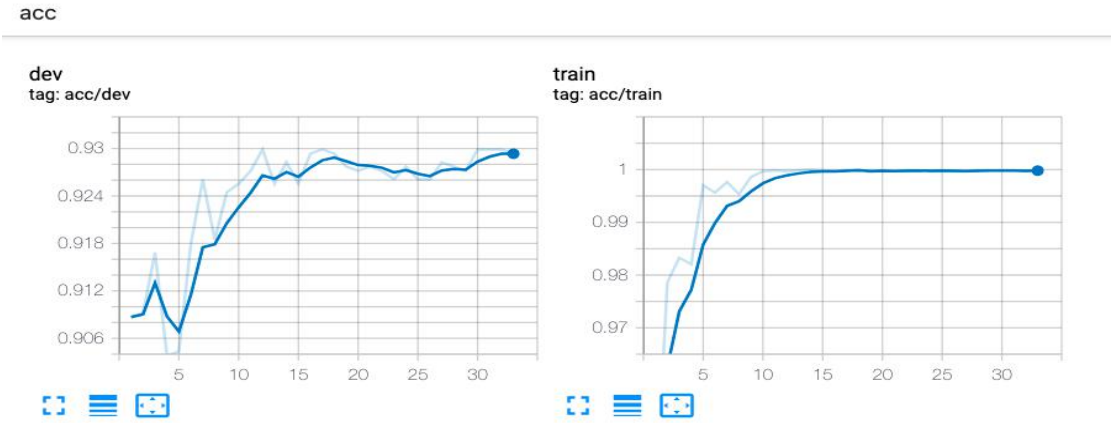

图 1-13　训练集准确率与验证集准确率

最后，在测试集上的结果见表 1-2，其中平均准确率和平均 F-Score 均达到了 0.94。

表 1-2　一级标签分类指标测试结果

	precision	recall	F1-score
城乡建设	0.93	0.92	0.92
环境保护	0.94	0.97	0.95
交通运输	0.95	0.92	0.93
教育文体	0.93	0.96	0.94
劳动和社会保障	0.96	0.95	0.95
商贸旅游	0.93	0.91	0.92
卫生计生	0.91	0.94	0.93
模型结果	0.94	0.94	0.93
加权结果	0.94	0.94	0.94

4.2 问题二的模型建立及求解

针对问题二，因为问题要求我们将某一时间段内反应特定地点或特定人群问题的留言进行归类，这显然是个聚类问题，可以认为需要将相似文本归成一类，那么我们首先需要对文本进行处理，将一个个文本表示成高维空间中不同的点，然后我们再根据不同这些点去计算他们之间的距离，最后将相距较近的点聚成一个簇，这样我们就可以聚类结果定义合理的热度评价指标求出排名前五的热度问题了。

在文本数字化之时，可以采用 One Hot 编码、Bag of Words 编码或者 tf-idf 编码将留言编码成高维向量，也可以采用深度学习技术将文本转化为低纬的绸密向量，考虑到本题数据并不多，因此我们分别使用两种方式去进行特征表示，进而比较两种算法的效果。

因此，我们首先对原始数据进行数据清洗、分词、停用词过滤等预处理操作，然后进行特征的提取，即将文档转化为特征向量，最后利用 k-means 算法对文本进行聚类，同时利用 PCA 和 SVD 将数据降至二维，显示算法聚类效果，然后我们基于类别大小、点赞数和反对数及时间长短定义了热度评价指标，对事件进行排序，并给出了计算结果。具体计算流程见图 2-1

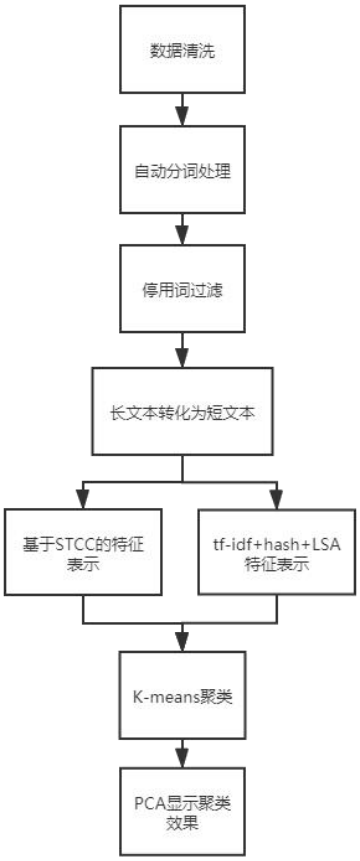

图 2-1　留言聚类算法流程图

4.2.1 数据预处理

本节主要详细介绍了数据清洗、分词、停用词过滤等预处理操作。

4.2.1.1 数据清洗

数据清洗一方面是为了解决数据质量问题，另一方面为了是让数据更适合做挖掘。本文主要是利用正则表达式将群众留言中出现的各种除中文、英文、数字外的符号替换为逗号，并去除重复的标点符号。

4.2.1.2 自动分词及去停用词

本文采用开源的 jieba 分词工具对句子进行切分,然后去除在中文停用词表、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[5]出现的词语.

4.2.1.3 长文本转化短文本

通过观察赛题提供的数据，可以发现附件中“留言详情”的文本质量不是很高，它不能简明扼要的表示留言的主要问题，为了更好的利用赛题提供的数据，我们将这一列的文本数据进行预处理，抽取出主要的地点、任务等关键词或句子作为留言的主要代表。考虑到文本的质量、长度等因素，采用文本摘要技术可能

不会取得很好的效果，因此本文采用关键词抽取技术，将抽取的关键词和“留言主题”结合起来用于后续计算任务。

本文采用基于 tf-idf 的关键词抽取技术。

（1）tf-idf 的主要思想

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）基本原理

在一份给定的文档里，词频(tf)指的是某一个给定的词语在该文档中出现的频率，这个数字需要对词数进行归一化，以防止偏长的文档。词语 t 对于文档 d 的词频定义为

$$tf_{t,d} = \frac{n_{t,d}}{\sum_k n_{k,d}} \tag{2-1}$$

$n_{t,d}$ 是该词在文档 d 中的出现次数，k 指词汇表所有词语。

逆文档频率（idf）是一个词语普遍重要性的度量。某一特定词语的 t，idf 定义为

$$idf_t = \log(\frac{1+N}{1+df_t}) + 1 \tag{2-2}$$

N 指的是所有文档的总数，df_t 指的是包含词语 t 的文档数目。

将 tf 和 idf 结合起来，定义 tf-idf 为

$$w_{t,d} = (1 + \log tf_{t,d}) \times idf_{t,d} \tag{2-3}$$

d 代表对应的文档.

（3）关键词抽取

对每一个“留言详情”，我们计算分词之后的每个词语的 tf-idf，选出每篇文档的 tf-idf 排名前 10 个词语做为本篇文档的关键词，将抽取的关键词和“留言主题”结合起来用于后续计算。

4.2.2 特征的提取

本文分别采用基于 tf-idf+Hash+LSA 技术[6]和深度学习技术进行特征提取.

4.2.2.1 基于 tf-idf+Hash+LSA 技术

我们首先将分词之后的句子经过 tf-idf 向量空间模型初步表示，考虑到词汇表巨大，因此我们首先通过 Hash 初步降维，然后我们采用 LSA 去提取句子的潜在语义，最后形成的向量即可用于后续聚类任务。

（1）初步特征表示

Bag-of-Words（词袋模型）是表示句子的一种方式，它可以通过为每个单词分配一个唯一的编码来完成句子表示。我们所看到的任何文档都可以被编码为一个固定长度的矢量，其长度为文档中全部已知单词的词汇量，矢量中每个位置的值代表编码文档中每个单词的出现个数。

考虑到像“the”这样的词会出现很多次，但却没有太大意义。因此，本文采用 tf-idf 模型去计算矢量中每一个位置的值，假设某个位置表示的是单词 t，那么使用公式（2-3）计算这个位置的矢量值。

假设经过上述流程之后句子被表示成为$(v_1,v_2,v_3,v_4,.....,v_{m-1},v_m)$

为了后边更好的特征提取，我们采用了 L2 Norm 标准化，即

$$v_i'=\frac{v_i}{\sqrt{(v_1^2+v_2^2+v_3^2+v_4^2+v_5^2+.....+v_m^2)}},i=1,...,m \tag{2-4}$$

（2）哈希降维

在本问题中，由于特征的维度对应分词词汇表的大小，所以面临着维度爆炸的问题，因此需要进行降维，我们采用最常用的文本降维方法：Hash Trick。

具体的方法是，对应任意一个特征名，我们会用 MurmurHash3-32bit 函数找到对应哈希表的位置，然后将该特征名对应的词频统计值累加到该哈希表位置，为了解决原始特征的词频累加导致的特征值突然变大的问题，我们采用了另外一个符号哈希函数，即：

$$\xi:N\to\pm1 \tag{2-5}$$

$$\bar{\phi}(j)=\sum_{i\in\tau:h(i)=j}\xi(i)\phi(i) \tag{2-6}$$

其中τ是原始特征的维度，$\phi(i)$指的是第 i 个原始特征的词频数值，$\bar{\phi}(j)$指的是哈希后第 j 个特征的词频数值。

（3）LSA 挖掘潜在语义

通过潜在语义分析，LSA 模型可以用来挖掘文本中的语义信息，把原文本特征空间降维到一个低维语义空间，减轻一词多义和一义多词问题，本文采用 SVD 去实现 LSA。

假设矩阵A是一个$m\times n$的矩阵，那么定义矩阵A的 SVD 为：

$$A=U\Sigma V^T \tag{2-7}$$

其中U是一个$m\times m$的矩阵，Σ是一个$m\times n$的矩阵，除了主对角线上的元素以外全为 0，主对角线上的每个元素都称为奇异值，V是一个$n\times n$的矩阵。

SVD 算法求解流程：

1）将 A^T 和 A 做矩阵乘法，则会得到一个 $n\times n$ 的一个方阵 A^TA，对其进行特征分解得到的特征值与特征向量：

$$(A^TA)v_i=\lambda_i v_i \tag{2-8}$$

将 A^TA 的所有特征向量张成一个 $n\times n$ 的矩阵 V，这就是 SVD 公式里面的 V 矩阵,一般将 V 中的每个特征向量叫做 A 的右奇异向量。

2）将 A 和 A^T 做矩阵乘法，则会得到一个 $m\times m$ 的一个方阵 AA^T，进行特征分解，得到的特征值和特征向量满足：

$$(AA^T)v_i=\lambda_i v_i \tag{2-9}$$

将 AA^T 的所有特征向量张成一个 $m\times m$ 的矩阵 U，这就是 SVD 公式里面的 U 矩阵,一般将 U 中的每个特征向量叫做 A 的左奇异向量。

3)由于 Σ 除了对角线上是奇异值其他位置都是 0,那只需要求出每个奇异值 σ 就可以了。

$$A=U\Sigma V^T \Rightarrow AV=U\Sigma V^TV \Rightarrow AV=U\Sigma \Rightarrow Av_i=\sigma_i u_i \Rightarrow \sigma_i=Av_i/u_i \tag{2-10}$$

可以求出每个奇异值，进而求出奇异值矩阵 Σ。

4）在奇异值矩阵中将奇异值从大到小排列，只保留前 k 个奇异值，其他的全部置 0，找出对应的 V 矩阵之后，使用 V 矩阵代替 A 矩阵进行分析。

基于上述 tf-idf 文本向量化、hash 降维、LSA 潜在语义分析三个步骤提取的向量即可用于后续的聚类任务。

4.2.2.2 基于 STCC 的特征表示

STCC[7]最初是用于短文本聚类任务的模型，考虑到本题留言的特殊性，因此该模型可以较好的适应本问题。该模型首先通过 simHash 和拉普拉斯降维将文本关键特征编码成二进制代码，然后将基于 word embedding 得到的表示经过 CNN 得到句子的深度特征表示，并使用基于 CNN 的输出和预训练好的二进制代码构成的损失函数去调整网络参数。我们将分别介绍 simHash、拉普拉斯降维、模型结构、模型训练四部分。

（1）SimHash

simHash 本身属于一种 locality sensitive hash（局部敏感哈希）算法，它产生的 hash 签名在一定程度上可以表征原内容的相似度。

SimHash 算法流程：

输入：文档集，输出二进制长度 N

1）分词

使用 jiaba 分词工具分词，不人工引入额外词汇。

2）hash

对文档集出现的每个词语，通过 hash 函数计算该词语的 hash 值，hash 值为二进制数 01 组成的 N-bit 签名。

hash 具体计算方式如下：

（a）如果词语为空字符串，则随机产生一串 N 位的二进制数

（b）如果词语不为空，设词语为 str,则：

(i)令 $x = ASCII(str[0]) \times 2^7, m = 10000007$

(ii)对词语的每个字符 ch，$x = (x * m)^{ch} \ \& \ (2^N - 1)$ （2-11）

(iii)输出 x 作为词语的 N-bit 签名

3）idf 加权

For sentence ∈ 语料集：

（a）初始化一个 N 维的向量，初始值全为 0

（b）对句子中的每一个词语，在 hash 值的基础上，给所有特征向量进行加权

$$W = Hash * weight \quad (2\text{-}12)$$

遇到 1 则 hash 值和权值正相乘，遇到 0 则 hash 值和权值负相乘。

（c）将各个特征向量的加权结果累加，变成只有一个序列串。

4）对每个句子，对于 n-bit 签名的累加结果，如果大于 0 则置 1，否则置 0，从而得到该语句的 simhash 值。

输出：每个句子得到一个 N 位的二进制签名。

（2）拉普拉斯特征映射

拉普拉斯特征映射是一种从局部的角度去构建数据之间关系的降维算法。具体的说，拉普拉斯特征映射是一种基于图的降维算法，它希望相互间有关系的点（在图中相连的点）在降维后的空间中尽可能的靠近，从而在降维后仍能保持原有的数据结构。

拉普拉斯特征映射通过构建邻接矩阵为 W 的图来重构数据流形的局部结构特征。算法主要思想是，如果两个数据实例 i,j 很相似，那么 i,j 在降维后目标子空间中应该尽量接近，设数据实例的数目为 n，目标子空间即最终的降维目标的维度为 m，定义 $n \times m$ 大小的矩阵 ，其中每一个行向量 y_i^T 是数据实例 i 在目标 m

维子空间中的向量表示（即降维后的数据实例 i）。算法的目的是让相似的数据样例 i, j 在降维后的目标子空间里仍旧尽量接近，拉普拉斯特征映射优化的目标函数：

$$\min \sum_{i,j} \left\| y_i - y_j \right\|^2 W_{ij} \tag{2-13}$$

拉普拉斯特征映射算法流程：

1）构建图

将所有的点构建成一个图

2）确定权重

确定点与点之间的权重大小，选用热核函数来确定，如果点 i, j 相连，那么它们关系的权重设定为：

$$W_{ij} = e^{-\frac{\left\| x_i - x_j \right\|^2}{t}} \tag{2-14}$$

3）特征映射

计算拉普拉斯矩阵 L 的特征向量与特征值，W 为(2)构建的权重矩阵，D 为对角矩阵，$D_{i,i} = \sum_i W_{ij}, L = W - D$

$$Ly = \lambda Dy \tag{2-15}$$

使用最小的 m 个非零特征值对应的特征向量作为降维后的结果输出。

（3）模型结构：

考虑到前边已经说明了卷积神经网络的基本原理，我们直接介绍如何将卷积神经网络用于抽取文本的特征表示，本问题的 CNN 结构如图 2-2：

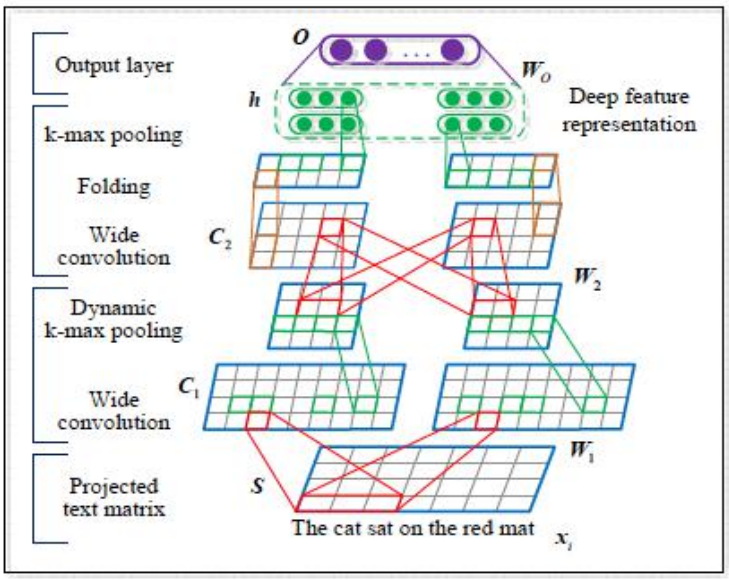

图 2-2　本问题使用的 CNN 结构图[7]

本文中，我们采用了一个两层的卷积神经网络，这个网络对一个输入句子进行特征提取，最后输出这个句子的特征表示。

定义卷积神经网络的输入为

$$\{x_i : x_i \in R^{d\times 1}\}, i = 1,2,...,n$$

n 表示语料集中的数量，d 是句子经过 word embedding 之后的特征维度，即

$$d = d_w \times s \tag{2-16}$$

d_w 指的是每一个词语的 embedding 维度，s 指的是句子的长度。

卷积神经网络的实质就是定义了一个特征映射

$$f(\cdot): R^{d\times 1} \to R^{r\times 1}, d >> r \tag{2-17}$$

将初始的文本特征表示转化为一个 r 维的特征表示，从中抽取主要特征，去除与任务无关的噪音。

模型总结构如图 2-3：

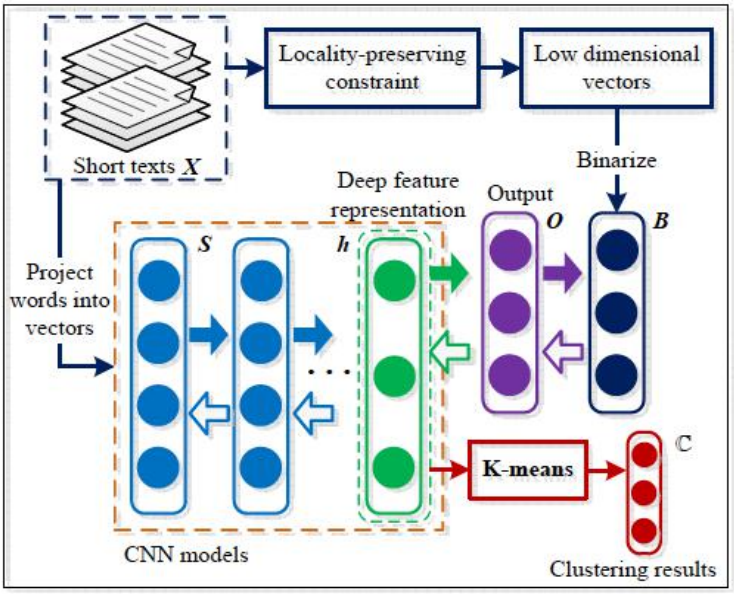

图 2-3　本问题模型总结构图[7]

（i）模型首先将给定的文本利用 simHash 算法将不同的文本转化为特定维度的向量，这个向量可以代表文本的主要特征，因为考虑到 simHash 算法直接将文本转化低维向量时表达能力不够，因此本文采用了拉普拉斯降维的方法，进一步将文本的特征表示转化为低维向量，该向量对应图 2-2 中的 B。

（ii）另一方面，模型根据输入文本，首先通过预训练好的 word embedding，将每个文本表示成一个向量，然后经过卷积、池化层之后得到深度特征表示向量 h，为了去适应预训练好的 B,模型采用了一个全连接层和 sigmoid 函数去对 h 做了映射：

$$O = W_0 h \tag{2-18}$$

$$P_i = \frac{\exp(O_i)}{1+\exp(O_i)}, i = 1,2,...,r' \tag{2-19}$$

（5）模型学习；

模型全部需要训练的参数定义为

$$\theta = \{W, W_o\}$$

其中W表示卷积神经网络的参数，最后采用了最大化似然函数的思想，定义

$$\tau(\theta) = \sum_{i=1}^{n} \log(b_i \mid x_i, \theta) \tag{2-20}$$

我们采用了批训练的反向传播算法去训练模型，并且为了解决过拟合问题，引入了 dropout 机制。

4.2.3 K-means 聚类

k-means 算法，属于无监督的聚类算法，对于给定的样本集，按照样本之间的距离大小，将样本集划分为 k 个簇。让簇内的点尽量紧密的连在一起，元素差异性尽可能小；让簇间的距离尽量的大，元素差异性尽可能大。

假设簇划分为$(C_1, C_2, ...C_k)$，则算法的目标是最小化平方误差 E：

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|_2^2 \tag{2-21}$$

其中，μ_i是簇C_i的均值向量，或称为质心，表达式为：

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{2-22}$$

k-means 算法流程：

输入样本集 $D=\{x_1,x_2,...x_m\}$，聚类的簇树 k，最大迭代次数 N

输出是簇划分 $C=\{C_1,C_2,...C_k\}$

1）从数据集 D 中随机选择 k 个样本作为初始的 k 个质心向量：$\{\mu_1,\mu_2,...,\mu_k\}$

2）对于 $n=1,2...,N$

a）将簇划分 C 初始化为 $C_t=\varnothing t=1,2...k$

b）对于 $i=1,2...m$，计算样本 x_i 和各个质心向量 $\mu_j(j=1,2,...k)$ 的距离：

$$d_{ij}=\left\|x_i-\mu_j\right\|_2^2 \tag{2-23}$$

将 x_i 标记最小的为 d_{ij} 所对应的类别 λ_i。此时更新：

$$C_{\lambda_i}=C_{\lambda_i}\cup\{x_i\} \tag{2-24}$$

c）对于 $j=1,2...k$，对 C_j 中所有的样本点重新计算新的质心

$$\mu_j=\frac{1}{\left|C_j\right|}\sum_{x\in C_j}x \tag{2-25}$$

d）如果所有的 k 个质心向量都没有发生变化，转至步骤 3）

3）输出簇划分 $C=\{C_1,C_2,...C_k\}$

对于 k-means 算法，首先考虑的是 k 值的选择，因为不存在先验知识，因此我们采用肘方法确定 K-Means 聚类的最佳 K 值。

k-means 是以最小化样本与质点平方误差作为目标函数，将每个簇的质点与簇内样本点的平方距离误差和称为畸变程度，其可以定义为

$$L=E \tag{2-26}$$

那么，对于一个簇，它的畸变程度越低，代表簇内成员越紧密，畸变程度越高，代表簇内结构越松散。 畸变程度会随着类别的增加而降低，但对于有一定区分度的数据，在达到某个临界点时畸变程度会得到极大改善，之后缓慢下降，这个临界点就可以考虑为聚类性能较好的点。

其次，在确定了 k 的个数后，需要选择 k 个初始化的质心。由于 k-means 算法是启发式方法，k 个初始化的质心的位置选择对最后的聚类结果和运行时间都有很大的影响，因此需要选择相互距离不是很近的 k 个质心。

4.2.4 热度评价

4.2.4.1 留言数量 C

直观上说，针对特定问题的留言数量直接反映了该问题的热度，反应该问题的留言数量越多，该问题就越热，留言数量越少，属于热点问题的可能性就越小。

4.2.4.2 点赞数和反对数 E

从人的心理学和行为学角度来说，一个人点赞一般表示他赞同该留言，说明可能他也有类似的问题，一个人反对这个留言，虽然有可能说明这个留言说的不正确，但是侧面反映了反对的人对这个问题的关注程度，因此，我们将点赞数和反对数相加一起作为影响热度的一个评价指标，总数与热度呈正相关。

4.2.4.3 时间长短 T

一般来说，一条留言只有在特定的时间发出，引起网友的共鸣，这样留言才会被更多的人关注，这条留言的热度也就越高。对于两个不同的问题，持续时间越长，留言数量 C 以及点赞数和反对数 E 越多，因此，时间长短需要作为一个抑制性指标，来控制指标 C 和指标 E 的增长。

4.2.4.3 热度指标量化

综合留言数、点赞数和反对数、时间长短，我们将问题的热度定义如下：

$$hot(d)=\frac{\alpha C+\beta E}{\gamma T} \tag{2-27}$$

α,β,γ 表示权重系数，d 表示经对留言聚类之后得到的特定问题。

4.2.5 PCA 呈现聚类效果

PCA 是指找出数据里最主要的方面，用数据里最主要的方面来代替原始数据。假如数据集是 n 维的，共有 m 个数据 $\left(x^{(1)},x^{(2)},...,x^{(m)}\right)$，需要将这 m 个数据的维度从 n 维降到 n' 维，让这 m 个 n' 维的数据集尽可能的代表原始数据集。数据从 n 维降到 n' 维肯定会有损失，但是需要让损失尽可能的小。

PCA 算法流程：

输入：n 维样本集 $D=\left(x^{(1)},x^{(2)},...,x^{(m)}\right)$，要降维到的维数 n'

输出：降维后的样本集 D'

1）对所有的样本进行中心化：

$$x^{(i)}=x^{(i)}-\frac{1}{m}\sum_{j=1}^{m}x^{(j)} \tag{2-28}$$

2）计算样本的协方差矩阵 XX^T

3）对矩阵 XX^T 进行特征值分解

4）取出最大的 n' 个特征值对应的特征向量 $(\omega_1,\omega_2,...,\omega_{n'})$，将所有的特征向量标准化后，组成特征向量矩阵 W

5）对样本集中的每一个样本 $x^{(i)}$，转化为新的样本

$$z^{(i)}=W^Tx^{(i)} \tag{2-29}$$

6）得到输出样本集 $D'=(z^{(1)},z^{(2)},...,z^{(m)})$

4.2.6 算法求解结果

（1）STCC 模型训练

本次实验中，我们采用 pytorch 深度学习框架，在嵌入层中，将数据集内的各个词都表示成词向量,词向量是根据搜狗新闻数据预训练出来的[8]。

在卷积层，我们利用 pytorch 中的 nn.Conv2d 构建了卷积层，其只在纵向上滑动。定义卷积层:卷积核的数目为 8，每个卷积核的尺寸为 3。在卷积层后，添加池化层，采用最大池化操作，对领域的特征值取最大操作。

然后通过 dropout 层，定义 dropout 的保留概率为 0.5，防止过拟合。

最后是全连接层，全连接层的神经元定义为(72,64)和(64,32)，并以 relu 和 sigmoid 进行激活,通过 sigmoid 的输出和预训练的二进制代码计算损失函数的值，并通过反向传播算法优化模型参数，Relu 的输出用于后续的聚类任务。

表 2-1　STCC 模型参数表

参数含义	参数名称	参数值
序列长度	Seq_length	20
卷积核数目	Num_filters	8
卷积核大小	Kernel_size	3
全连接层神经元	Hidden_dim	64，32
Droupot 保留比例	Dropout_keep_prob	0.5
学习率	Learning_rate	1e-3
批训练大小	Batch_size	100
总迭代次数	Num_epochs	8000

同时，我们利用 tensorboardX 库，输出实验中构建的 STCC 模型的结构、损失函数及网络参数的变化情况，如图 2-4、图 2-5、图 2-6、图 2-7、图 2-8 所示：

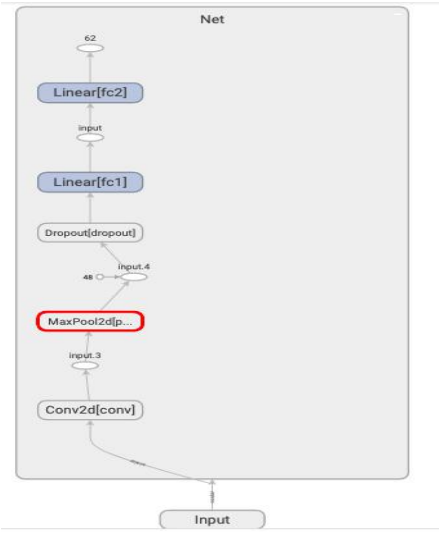

图 2-4　STCC 模型的结构图

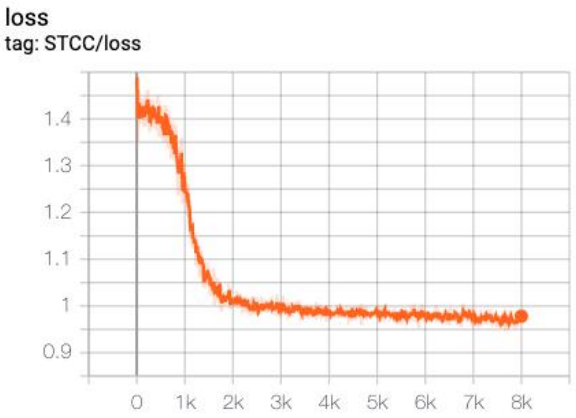

图 2-5　STCC 模型损失函数变化图

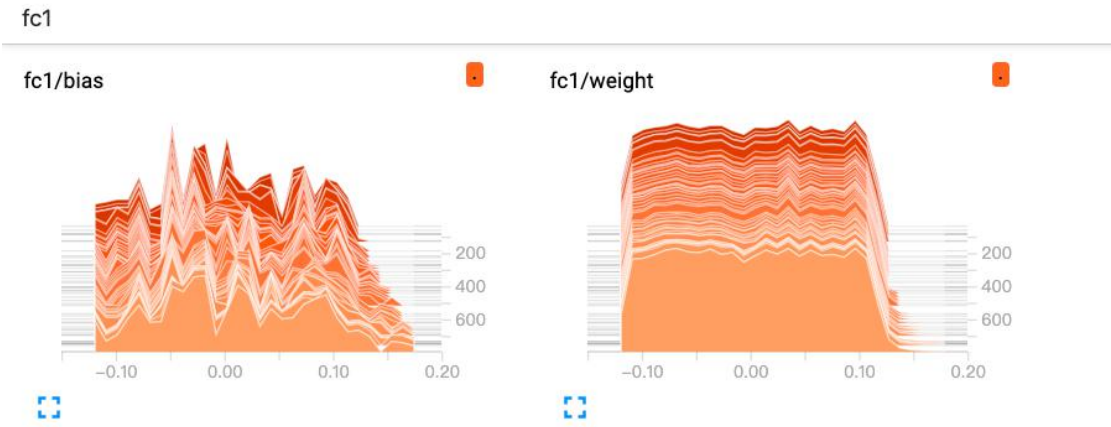

图 2-6　网络参数

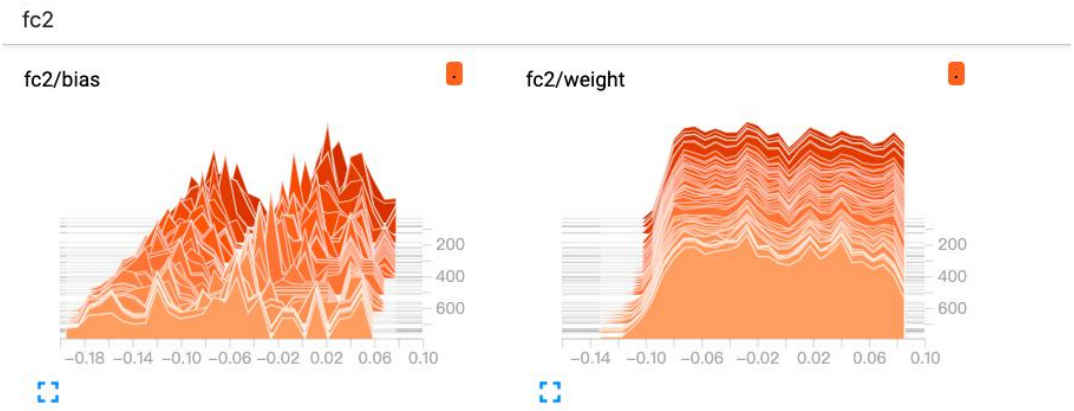

图 2-7　网络参数

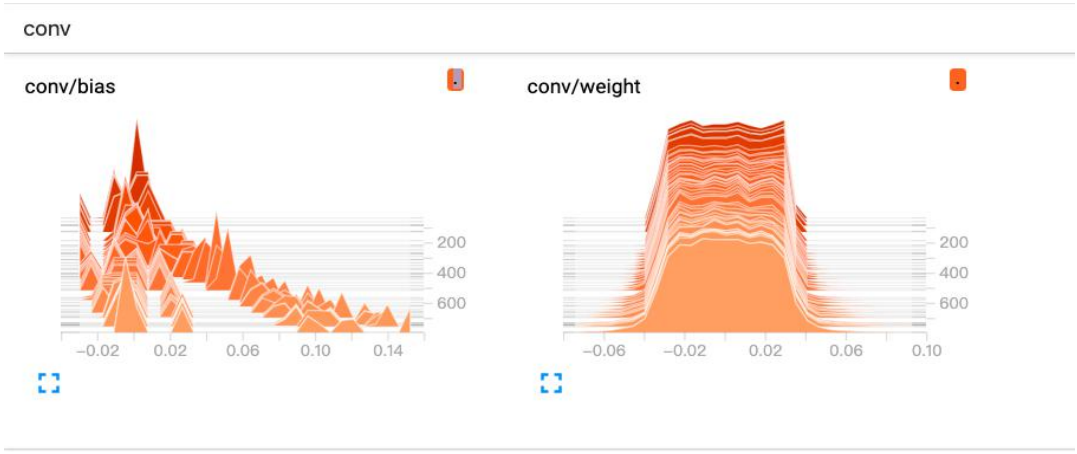

图 2-8　网络参数

（2）算法聚类效果分析

（i）基于 tf-idf+hash+LSA 聚类

我们首先通过 jieba 库进行分词，然后 tf-idf 向量空间模型对留言进行初步表示，然后通过 hash 函数将这个高维向量降成 4000 维，最后通过 LSA 将该向量降成 400 维，并通过肘部方法确定最佳类别数：

其中，模型的参数见表 2-2，肘部方法的误差变化图如图 2-9，WSCC 表示聚类平方误差：

表 2-2　tf-idf 向量空间模型参数表

参数含义	参数名称	参数值
去停用词	Stop_words	数据预处理阶段的用到的所有停用词
哈希降维数	n_features	4000
LSA 降维数	n_components	400
肘部方法的最大值	N_clusters	400
实际类别数-Kmeans	clusters	200

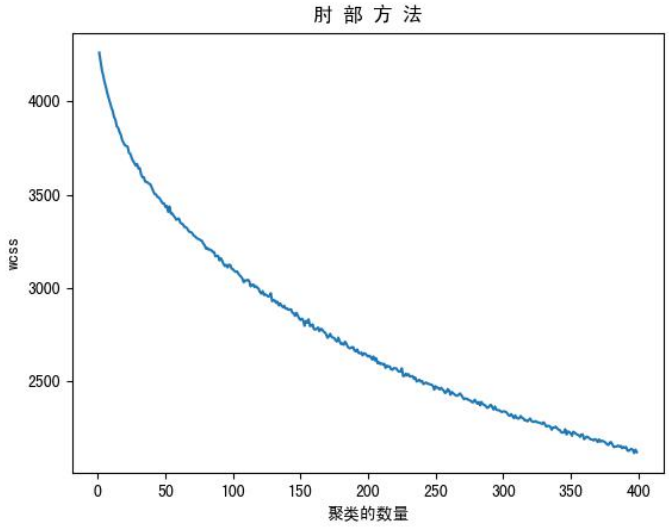

图 2-9　肘部方法的误差变化图

（ii）最后，我们选定聚类数量为 200 并设定$\alpha=\beta=25,\gamma=1$,通过 PCA 降维，求出了排名前五的热点问题，图 2-10、图 2-11 和图 2-12 分别显示了 tf-idf 模型求解的热度排名及通过 td-idf 和 STCC 对排名前八的热点问题的聚类结果：

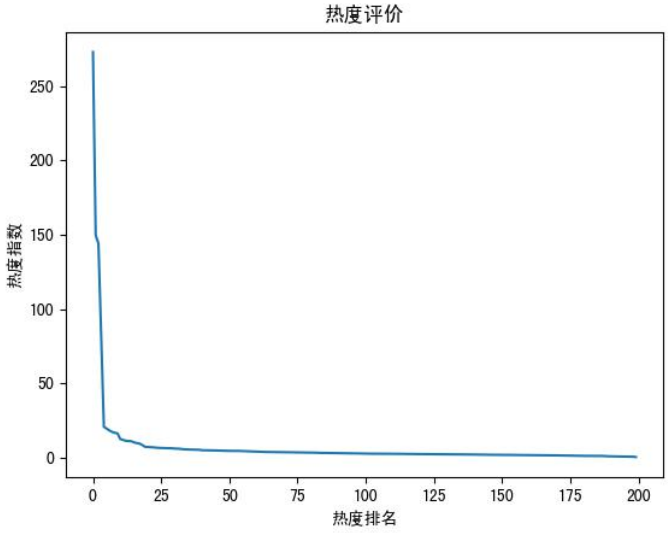

图 2-10　tf-idf 模型求解的热度排名

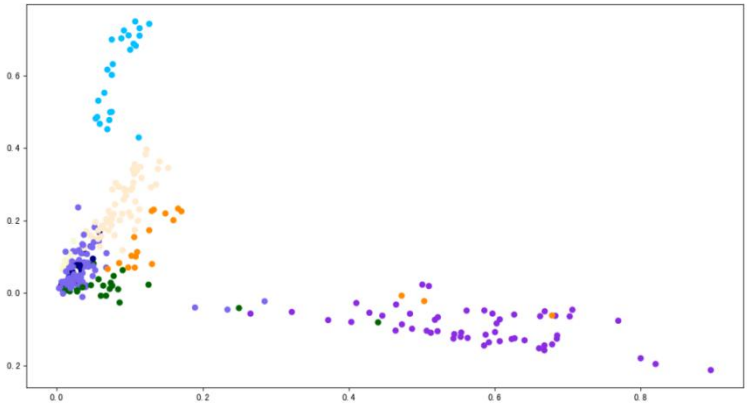

图 2-11　td-idf 对排名前八的热点问题的聚类结果图

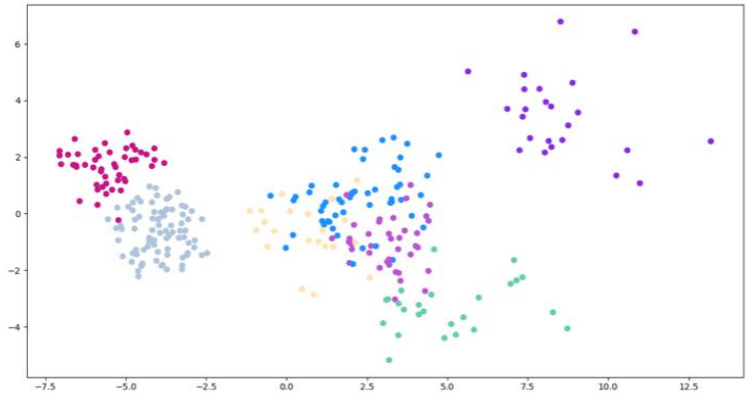

图 2-12　STCC 对排名前八的热点问题的聚类结果图

从图 2-10 可以看出，只有排名约前十的热点问题具有相对较高且明显的热度值，而其余热点问题的热度值极低且趋近于 0。图 2-11 和图 2-12 中的不同颜色代表聚类算法生成的不同类别，通过颜色分布情况可以看出，这两种算法都有很好的聚类效果。

（3）求解热点问题

根据（2）所述，我们给出了基于 tf-idf 模型求出的排名前五的热点问题的归类结果，见表 2-3：

表 2-3　基于 tf-idf 模型的排名前五的热点问题表

热度排名	问题 ID	热度指数	时间范围	地点/人群	问题描述
1	1	272.52	2019/1/11 至 2019/8/21	A 市 58 车贷案受害人	58 车贷案处理不周，希望相关部门妥善处理该案件
2	2	266.25	2019/3/26 至 2019/4/15	A 市 A6 区月亮岛路居民	架设 11 万伏高压线环评造假且存在安全隐患
3	3	176.5	2019/1/15 至 2019/11/11	A 市五矿万境 K9 县居民	居住环境存在许多安全隐患，业主权利无法得到保障
4	4	88	2019/7/7 至 2019/10/18	A 市部分小区的居民	遭到开发商捆绑销售车位并且存在违规涨价问题
5	5	33	2019/6/19 至 2020/1/9	A 市丽发新城小区居民	长期遭受噪音污染和粉尘污染，影响正常生活

4.3 问题三的模型建立及求解

4.3.1 相关性评价

留言和答复的相关性建模与相似性不同，我们需要提取留言和答复的主题，然后比较主题相关性。

本文中，对于每个留言，我们利用 TF-IDF 模型对留言中的词语进行排序，提取其中的排名前 N 的关键词。完成所有文章的关键词提取后，将各个文章的关键词聚集起来，构造出一个热点词的词汇表。构建的同时，将重复的热点词进行去重操作。

在完成热点词的词汇表构造后，我们根据这个词汇表通过 Bag-of-Words 模型将所有的留言和回复向量化。在得到了所有文本的热点词向量后，根据余弦相似度公式计算留言和回复之间的距离，其中余弦相似度定义为

$$\cos\theta = \frac{x_i \cdot y_i}{\|x_i\| \times \|y_i\|} \tag{3-1}$$

其中，x_i, y_i 分别表示留言和回复向量化之后的向量。

经过运算，得到留言和答复的余弦相似度结果如图 3-1：

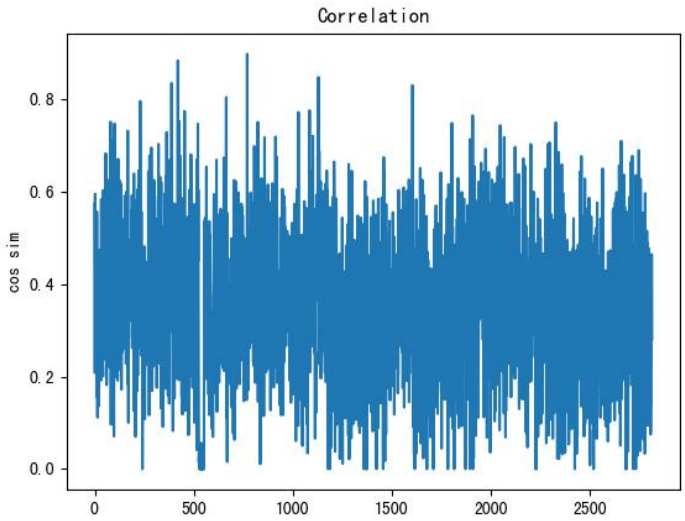

图 3-1　留言和答复的余弦相似度结果

根据图 3-1 可知，大部分相似度位于 0.2-0.6 之间，因此我们定义阈值边界分别为：0.3 和 0.65。最后我们根据阈值边界将文章相关性分为三类：很相关、比较相关、不相关。具体标准见表 3-1：

表 3-1　相关性标准

主题相关度	相关性
0-0.30	不相关
0.30-0.65	比较相关
0.65-1	很相关

根据表 3-1 的定义，可以得到文本相关性的结果如图 3-2：

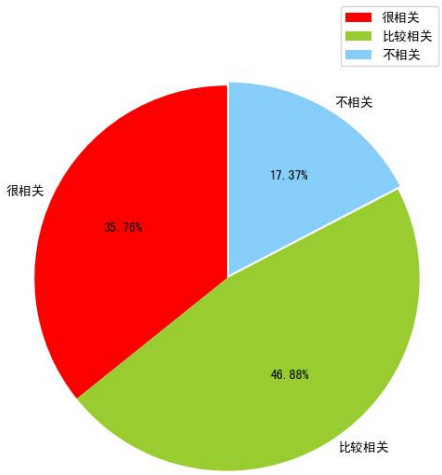

图 3-2　文本相关性的结果

4.3.2 完整性评价

完整性指的是回复内容的结构的完整性，通过分析回复数据，我们假定留言的回复是由开头、主体、结尾三部分组成的，文章开头会出现类似于“您好，你的留言已收到。”之类的语言；文章主体回复留言相关的问题；文章结尾会出现类似于“谢谢您的来信，2019 年 3 月 20 日”之类的话。

为了建模完整性，我们基于问题一的 TextCNN 训练三个不同的神经网络，网络一用于判断当前句子是否是用于开头的语言，网络二判断当前句子是否是文章主体的句子，网络三用于判断当前句子是否是用于结尾的语言。

假定模型训练好之后，设 f_1, f_2, f_3 分别代表三个网络输出的结果，我们定义回复的完整性为：

$$F = f_1 + f_2 + f_3 \tag{3-2}$$

其中，$f_1, f_2, f_3 \in \{0,1\}, F \in [0,3]$.

我们定义完整性为三类：很完整、比较完整、不完整。具体标准见表 3-2：

表 3-2　完整性标准

完整性分数	完整性
(2-3]	很完整
(1-2]	比较完整
[0-1]	不完整

我们根据手动标注的 300 条数据，划分了训练集和测试集，其中训练集的 loss、accuracy 和测试集的 loss、accuracy 分别如图 3-3、图 3-4 所示：

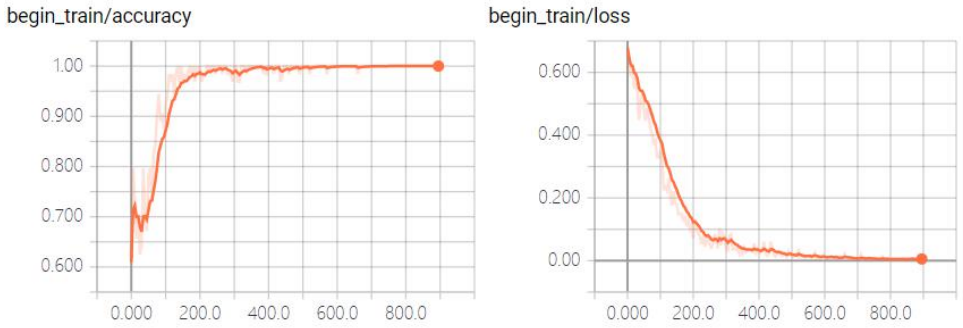

图 3-3　训练集结果

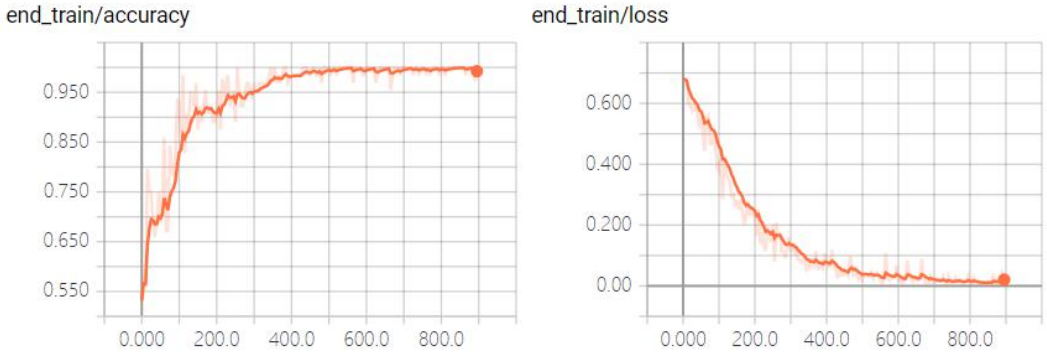

图 3-4　测试集结果

我们将训练好的模型用于全部数据的计算，最后得到答复的完整性结果如图 3-5 所示：

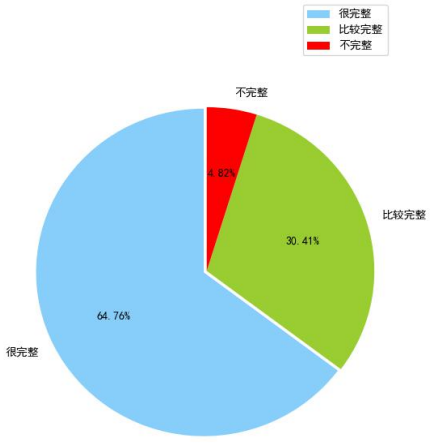

图 3-5　答复的完整性结果

4.3.3 可解释性评价

可解释性指的是回复是否具有依据或是否具有因果关系，其中可能会出现像“经过举行讨论会，我们作出如下答复”、“根据宪法，我们建议”、“经过实地考察，我们认为”之类的语言。

为了建模可解释性，本文分别从留言与答复的相关性、回复文本长度、回复所含信息量、是否含有“根据宪法”或“经过讨论会”之类的语言等四个方面去评价一个回复的可解释性程度。

（1）留言与答复的相关性 C

如果留言和答复的相关性不强，那么可能意味着出现了“答非所问”的情况，这种情况下应当认定答复是不具有可解释性的。

（2）回复文本长度 L

直观上来说，如果答复的文本长度很短，那么它有极大可能没有什么意义，可以认为答复敷衍，不具有可解释性。

（3）信息量 H

信息量指的是“信息浓度”， 答复的信息量也可以一定程度上代表可解释性的强弱，比如“感谢您的来信，我们会努力解决的。”和“您好，我们针对您提出的问题开了讨论会，并实地调研了”这两句话的信息量显然不一样，后者信息量更大并且更具有可解释性。

本文利用信息熵去衡量每个答复的信息浓度。信息熵是由香农提出的，用来衡量信息量的期望，其定义为

$$H(X) = -\sum_{x} P(x)\log P(x) \tag{3-3}$$

其中，X 指每一个答复，x 指答复分词之后的每一个词语。

经过运算，信息量结果如图 3-6：

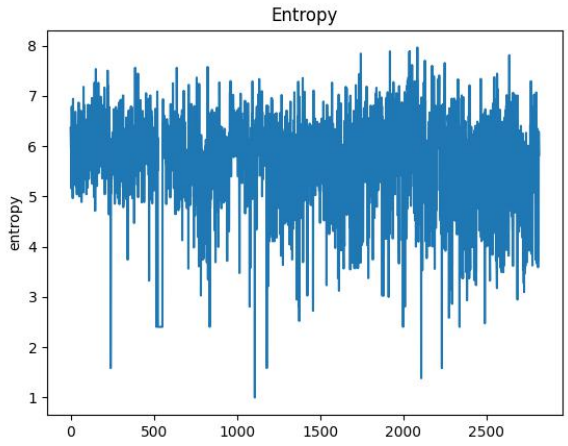

图 3-6　答复信息量

（4）因果关系 Q

这里的因果关系指的是前文提到的“开了讨论会”之类的答复，针对该问题，通过分析赛题数据，我们将答复分为四类：讨论会类、法律类、实地调研类、无可解释性类。

我们首先基于 tf-idf 模型对每个答复进行表示，然后通过 SVM（支持向量机）对答复进行分类，假定训练好的模型在四个类别上的分类概率分别为 q_1,q_2,q_3,q_4，则我们定义因果关系强弱 Q 为

$$Q = q_1 + q_2 + q_3 + q_4 \tag{3-4}$$

其中，$q_1,q_2,q_3,q_4,Q \in [0,1]$.

我们通过手动标注的 300 条数据完成了对模型的训练及测试，模型在测试集上达到了 88%的准确率。

最后，我们将训练好的模型用于全部答复数据的因果强弱计算，得出如图 3-7 所示结果：

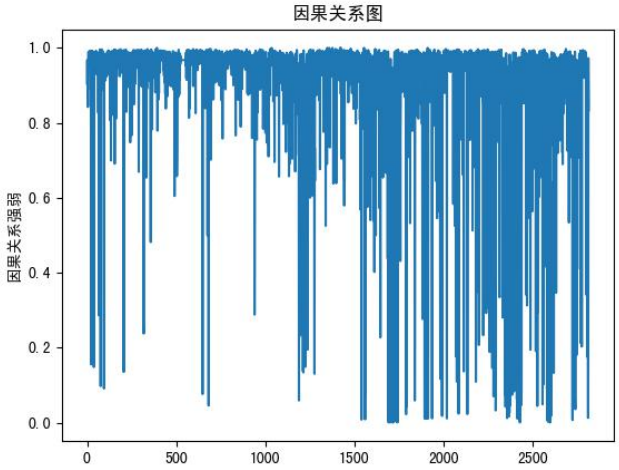

图 3-7　答复数据的因果强弱

（5）综合指标

经过（1）（2）（3）（4）的建模，我们分别从相关性、答复文本长度、信息熵、因果关系四个方面评价了答复可解释性，为了综合评价可解释性，我们定义答复的可解释性 S 为

$$S = \alpha C + \beta L + \gamma H + \delta Q \tag{3-5}$$

其中，$\alpha,\beta,\gamma,\delta$ 分别代表四个指标的权重。

本文首先对 C,L,H,Q 进行了归一化，并设定 $\alpha = 1,\beta = 1,\gamma = 1,\delta = 1$，得出了所有答复的可解释性程度如图 3-8 所示：

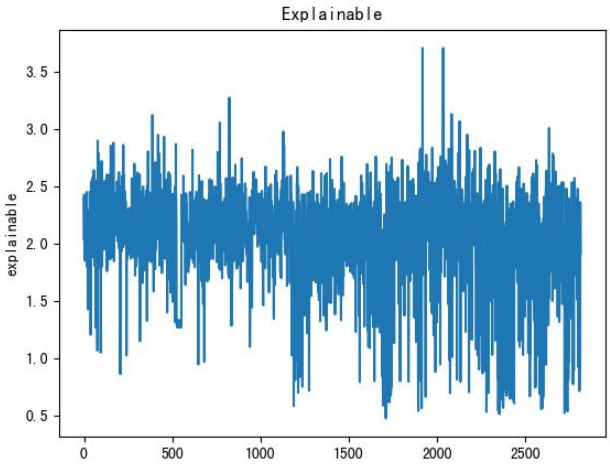

图 3-8　答复的可解释性程度

通过观察图 3-8 可知，大部分可解释性位于 1.0-2.8 之间，因此我们定义阈值边界分别为：1.5 和 2.2。最后根据阈值边界将答复的可解释性分为三类：可解释、比较可解释、不可解释。具体标准见表 3-3：

表 3-3　可解释性标准

可解释性分数	可解释性
(2.2-4]	可解释
(1.5-2.2]	较可解释
[0-1.5]	不可解释

根据表 3-3 的定义，可以得到可解释性结果如图 3-9 所示：

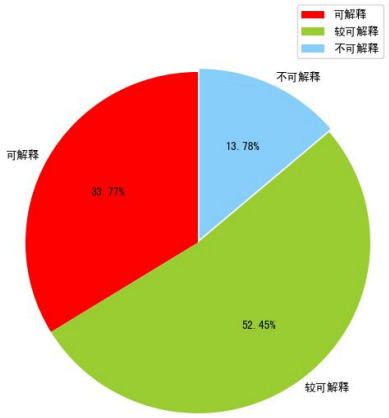

图 3-9　答复的可解释性结果

4.3.4 答复意见质量的综合评价

针对相关部门对留言问题的答复意见评价，我们从答复的相关性、完整性以及可解释性三个角度入手，分别使用了不同的诠释思路，建立不同的评价模型，

得到三个量化的评价指标，分别是：主题相关度$\cos\theta$、答复完整性F以及答复可解释性S。

为了综合这三个评价指标，我们定义答复意见的质量评价指标为：

$$E = i\cos\theta + jF + kS \tag{3-6}$$

其中，i, j, k分别代表三个评价指标的权重值。

设定$i = 1.5, j = 0.6, k = 0.9$，得出了所有答复意见的评价质量如图 3-10 所示：

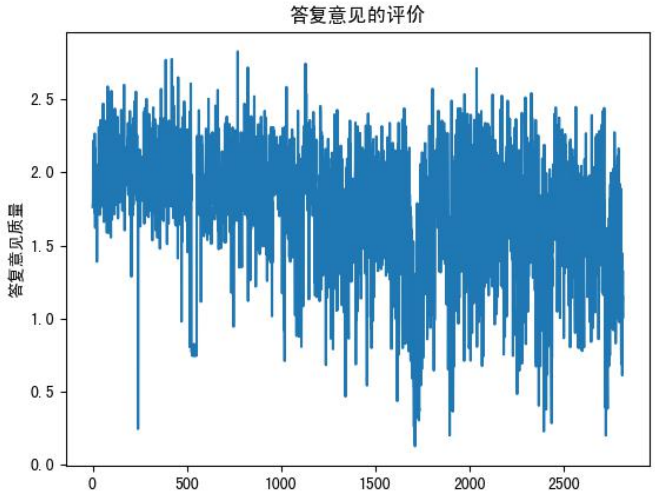

图 3-10　所有答复意见的质量

通过观察根据图可知，大部分答复意见的评价质量位于 1.0-2.5 之间，因此将阈值边界定义为：1.5 和 2.2。最后我们根据阈值边界将答复意见的质量分为三类：质量高、质量较高、质量低。具体标准见表 3-4：

表 3-4　答复意见质量标准

答复意见的质量分数	答复意见质量
(2.2-3]	质量高
(1.5-2.2]	质量较高
[0-1.5]	质量低

根据表 3-4 的定义，可以得到答复意见的质量结果如图 3-11 所示：

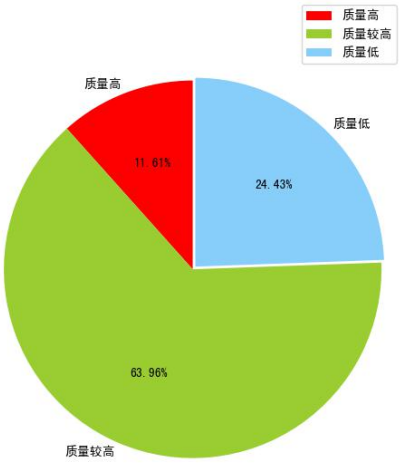

图 3-11　答复意见质量的结果

五、模型评价与推广

5.1 模型的优点

本文首先使用 TextCNN 模型解决了问题一的多分类问题，使用 TF-IDF 模型和 STCC 模型解决了问题二的文本聚类问题，使用 SVM 模型、主题相关度模型解决了问题三的评价问题。

（1）TextCNN 有共享卷积核，自动进行特征提取的优点，适合于解决文本多分类问题。但是 TextCNN 上下文关系识别能力的不足，因此我们在 TextCNN 模型的基础上，引入了 Bert 这一预训练模型，使整体模型能够捕捉更加长距离的依赖，提高了分类的准确性。

（2）由于本问题是一个无监督问题，很难评价聚类结果的质量，因此我们分别使用了两种模型对句子进行表示，将其用于后续的聚类问题，这样有利于从不同的角度分析模型的效果。

（3）SVM 模型可以高效的将训练样本转导至预报样本，简化文本的分类和回归，具有较好的“鲁棒性”。

5.2 模型的局限性

本文采用多种模型对给定的样本数据进行分类、聚类、评价，整体得到了较好的效果，但是由于问题给定的样本数据以及自行标注的数据较少，无法有效的训练模型，导致模型的效果未达到理想状态。

参考文献

[1] Yoon Kim.Convolutional Neural Networks for Sentence Classification. arXiv:1408.5882v2,2014

[2] Devlin J, Chang M W, Lee K, Et Al. Bert: Pre-Training of Deep Bidirectional Transformers For Language Understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[3]陆晓蕾,倪斌.基于预训练语言模型的BERT-CNN多层级专利分类研究. arXiv:1911.06241,2019

[4]Vaswani A, Shazeer N, Parmar N, Et Al. Attention Is All You Need[C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.

[5]中文停用词表：https://github.com/goto456/stopwords

[6]K-means 文本聚类：
https://www.kesci.com/home/project/5b6fb122a537e00010037dbd

[7]Jiaming Xu,Peng Wang,Guanhua Tian,Bo Xu,Jun Zhao,Fangyuan Wang,Hongwei Hao.Short Text Clustering via Convolutional Neural Networks[C]//Association for Computational Linguistics.2015:62-69

[8]预训练词向量：https://github.com/Embedding/Chinese-Word-Vectors