智慧政务中的文本挖掘

摘要

在这个信息时代，“智慧政务”的出现成了必然，它是大数据社会的产物。目前“智慧政务”的发展日趋成熟，已形成高效、敏捷、便民的新型政府服务模型。然而，群众表达民意便捷化的同时，政府收到的民意相关的文本数据量急剧攀升，相关部门依靠人工整理的传统模式已行不通，新型智慧政务系统需要依靠自然语言处理技术，才能有效地提高政府的管理水平和施政效率。本文将从留言划分、热点整理和答复评价等三个方面，提高“智慧政务”的效率，利于建设服务型政府，明确社会管理体制变革的重点任务和方向。

问题一的求解——基于传统机器学习与深度学习的文本分类模型。首先对附件2的留言详情进行数据清洗与分词去除停用词的预处理；其次，结合if-idf方法对分词结果进行特征向量化；再者，划分训练集与测试集，基于传统机器学习方法训练文本分类模型，分别使用网格搜索优化的随机梯度下降算法（即SGD算法）、逻辑斯蒂回归算法（即LR算法）和支持向量机算法（SVM算法）这三个算法，用测试集对模型的效果进行测试，三个模型的评价指标F1-score分别为0.91、0.90、0.90，可以看出这个文本分类模型的泛化能力相对较好。基于深度学习方法训练文本分类模型，分别使用卷积神经网络和循环神经网络，卷积神经网络中，对输入的留言关键词进行了两次卷积核大小为3、4、5的卷积操作后，使用ReLU激活函数，防止反向传播过程中的梯度弥散和梯度爆炸问题，然后经过最大池化，将池化的向量拼接起来，进行dropout处理，使训练效果最佳化。其中TextCNN、LSTM以及GPU三个模型的F1-score分别为0.662、 0.794、 0.826。从上面两种经典方法的测试效果，可以看出，对于此数据集，基于传统机器学习的文本分类模型效果优于深度学习模型。

针对问题二，本题先根据留言详情进行聚类，将描述同一件事的留言聚成一类，然后利用自己训练的NER模型识别留言主题的实体，将每类中的留言对实体进行归类，基本得到每类留言是关于某一群体在某一地点的事件，对每类留言使用热度评价指标得出热点排名，并反推回热点问题明细表。

问题三的求解——基于概率检索模型的算法BM25。首先对数据做预处理，并筛选含有代表性意义词性的词语。针对每条答复详情，对其相对应的留言详情进行相关性计算，其中，使用全部的留言关键词来BM25模型建立文档词库，通过IDF逆文档率来计算语素与文档D的相关性权重。BM25的得分计算公式为$Score(Q,d) = \sum_i^n Wi \cdot R(qi,d)$，根据此可算出每条答复详情的得分，接下来划分两个得分阈值15、10，分别将留言详情分为优、良、差三个部分，优的答复详情的相关性、完整性以及可解释性是比较好的。

关键词：TF-IDF ，SGD，LR ，SVM，TextCNN，LSTM，GPU，凝聚聚类 ，NER命名实体识别，BM25，相似度

Abstract

The solution of problem three:: text classification model based on traditional machine learning and deep learning. Firstly, the details of the message in Annex 2 are preprocessed by data cleaning and segmentation to remove the stop words; secondly, the segmentation results are vectorized by if IDF method; thirdly, the training set and test set are divided, the text classification model is trained based on traditional machine learning method, and the random gradient descent algorithm (SGD algorithm) and logistic regression algorithm optimized by grid search are used respectively Method (LR algorithm) and support vector machine algorithm (SVM algorithm) are three algorithms. The test set is used to test the effect of the model. The F1 score of the three models is 0.91, 0.90 and 0.90 respectively. It can be seen that the generalization ability of this text classification model is relatively good. Based on the deep learning method, the text classification model is trained, using convolution neural network and cyclic neural network respectively. In convolution neural network, the input message keywords are convoluted twice with convolution kernel size of 3, 4 and 5, and then the relu activation function is used to prevent the gradient dispersion and gradient explosion in the process of reverse propagation. After the maximum pooling, the pooled The training effect is optimized by combining vectors and dropout processing. The F1 scores of textcnn, LSTM and GPU are 0.662, 0.794 and 0.826 respectively. From the test results of the above two classical methods, we can see that for this dataset, the text classification model based on traditional machine learning is better than the deep learning model.

The solution of problem two: This topic first clusters the messages describing the same thing into a group according to the details of the messages, then uses our own trained ner model to identify the entities of the message subject, classifies the entities of each type of message, and basically gets each type of message is about an event of a group at a certain point, uses the heat evaluation index for each type of message to get the hot spot ranking, and reversely pushes back the hot spot List of problems.

The solution of problem three: algorithm BM25 based on probability retrieval model. First, preprocess the data and select the words with representative meaning. For each reply detail, the relevance calculation is carried out for the corresponding message details. Among them, the BM25 model is used to build the document thesaurus, and the relevance weight of morpheme and document D is calculated by the inverse document rate of IDF. The score calculation formula $Score(Q,d) = \sum_i^n Wi \cdot R(qi,d)$of BM25 is that the score of each reply detail can be calculated based on this. Next, it is divided into two score thresholds 15 and 10. The message details are divided into three parts: excellent, good and bad. The relevance, integrity and interpretability of the excellent reply details are relatively good.

Keywords: TF-IDF , SGD, LR , SVM, TextCNN, LSTM, GPU, Cluster, NER, BM25，Similarity

1 引言

文本分类是 NLP 中的常见的重要任务之一，它的主要功能就是将输入的文本以及文本的类别训练出一个模型，使之具有一定的泛化能力，能够对新文本进行较好地预测。它的应用很广泛，在很多领域发挥着重要作用，例如垃圾邮件过滤、舆情分析以及新闻分类等。

2 整体思路

本文先以流程图展示解题过程。

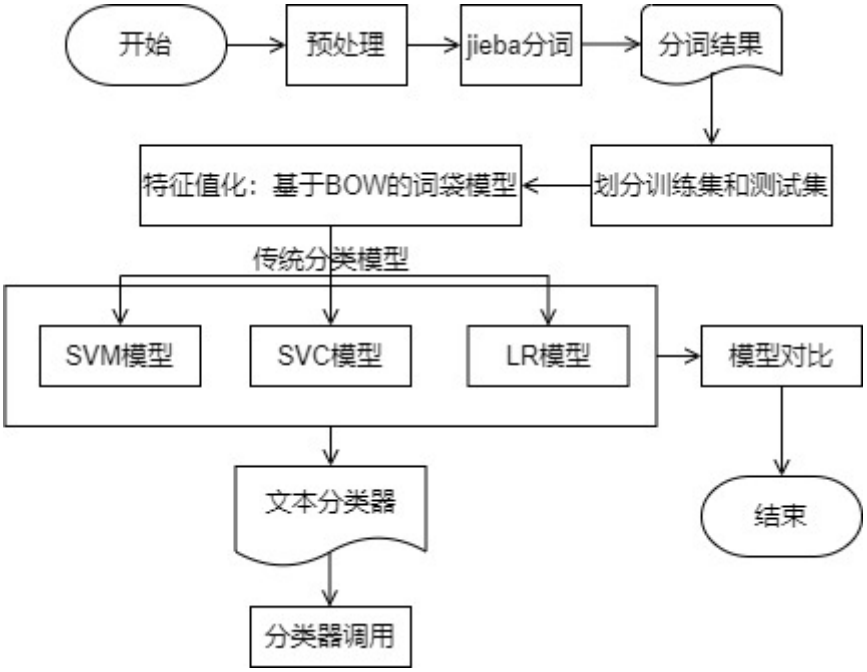

图 1 问题一的传统机器学习分类模型流程

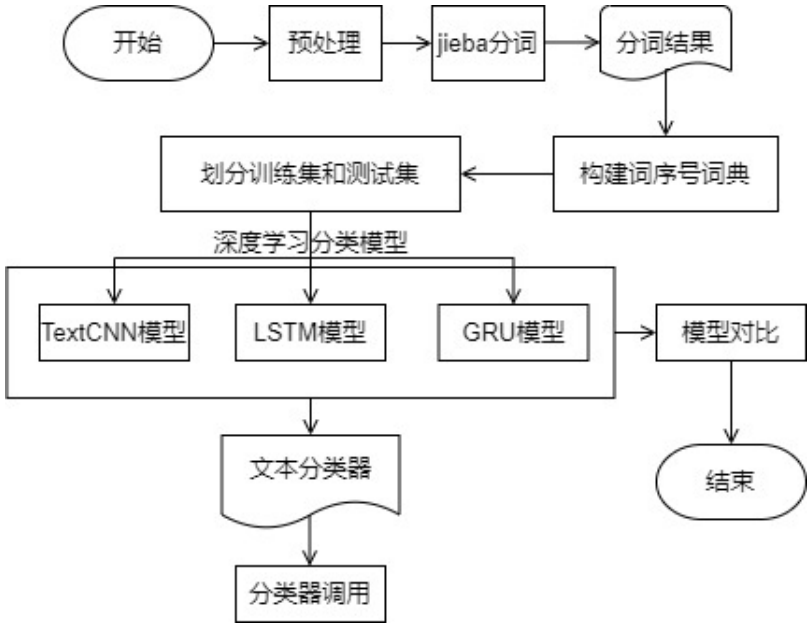

图 2 问题二的深度学习分类模型流程

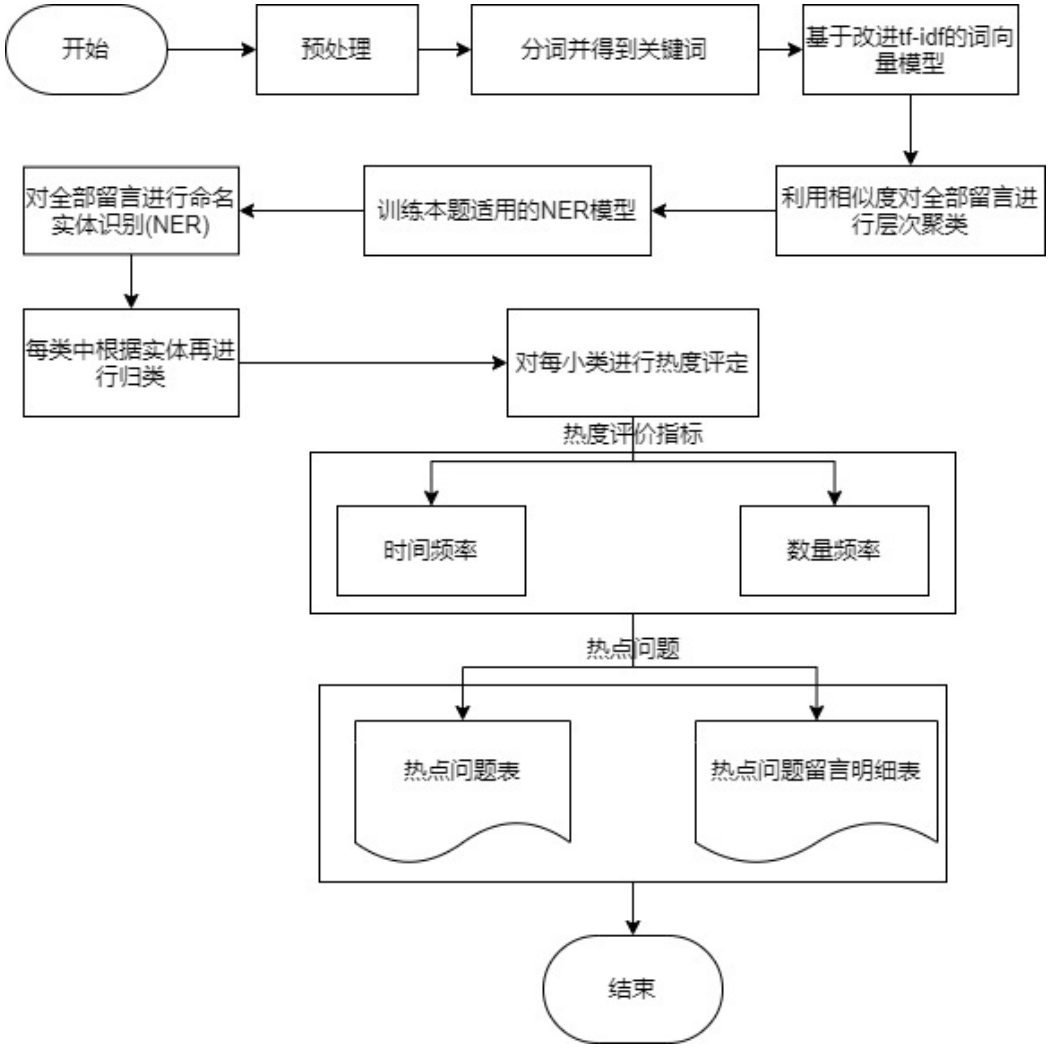

图 3 问题二的解题过程

3 实验过程

3.1问题一

本题主要运用文本分类算法来实现，文本分类算法主要实现了两大类分类算法模型，一个是基于统计学的分类算法模型，一个是基于深度学习的分类模型。在基于统计学的分类模型上，主要实现了梯度下降算法、LR 算法和支持向量机算法。在深度学习的分类模型上，实现基于。。。。。，并将其与传统分类算法的效果做对比。

3.1.1 预处理

本题的数据预处理部分包含两个方面，一个是数据清洗，另一个是分词去除停用词。

数据清洗：对官网提供的附件 2 数据进行空值的删除以及时间格式的统一调整,得到 cleaned_data2 的数据集,总共有 9210 条数据,其中有交通运输、劳动和社会保障、卫生计生、商贸旅游、城乡建设、教育文体、环境保护这七个一级标签类别。

分词去除停用词：分词采用的是 python 自带的 jieba 分词器，使用默认的精确模式，调用其分词词典，其较适合作文本分析。停用词是认为带有很少的有助于分类任何信息的代词、介词、连词等高频词，通过网上已有的较成熟的中文停用词资源进行整理，得到最终的 stopwords.txt。

3.1.2 传统机器学习分类模型

- 特征工程

基于 BOW 的词袋模型对文本进行特征值化。首先，将分词后的所有词构成一个词袋，并对它们进行索引标识，每个单词我们用一个向量来表示，其中向量的维度根据词袋中不重复词的个数来确定，向量中每个数字是词典中每个单词在文本中出现的频率，即词频 tf。在此基础上，给每个词分配一个权重，其中在多篇文档中出现的词语赋予较小的权重，在极少文档出现的文章中赋予较大权重，这个权重记为逆文档频率。结合词频和逆文档频率得到 tf-idf 值，一个词的此值越高，则重要性越大，在 python 中的是实现是用 CountVectorizer 函数和 TfidfTransformer 函数结合使用。

- 划分训练集和测试集

利用 python 的 train_test_split 函数将预处理后的数据集随机分成 70%训练集和 30%测试集，其中，训练集 6447 条，测试集 2763 条，将训练集保存为 machine_learning_train_data 数据集，然后分别对分别对训练集和测试集进行第一步的特征工程，将训练集各标签涵盖的样本量占比绘制成饼图如下：

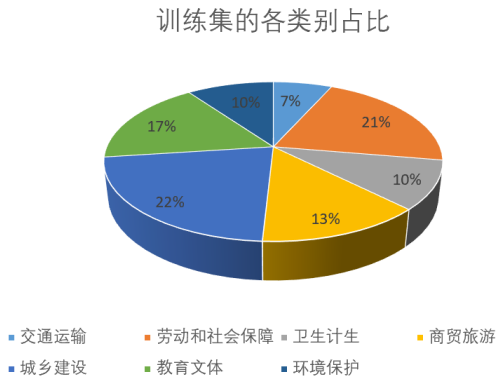

图 4 训练集各类占比情况

根据上图，我们可以看到训练集的各标签类别所包含的样本量存在一定的不平衡性，环境保护、交通运输、卫生计生这三个类别的数据较少，下面在计算模型评价效果时将考虑数据不平衡的计算情况。

- 训练模型

利用前面处理过的训练集来构建三个传统机器学习分类算法，分别是随机梯度下降算法（即 SGD 算法）、逻辑斯蒂回归算法（即 LR 算法）和支持向量机算法（SVM 算法）。

➢ SGD 算法

SGD 算法主要应用在大规模稀疏数据问题上，经常用于文本分类。本次调用 python 的 sklearn 包中的 SGDClassifier 函数来进行多分类，它以”one-vs-all(OVA)”的方式通过结合多个二分类来完成。对于 K 个类中的每个类来说，一个二分类器可以通过它和其它 K-1 个类来进行学习得到。在测试时，我们会为每个分类器要计算置信度（例如：到超平面的有符号距离）并选择最高置信度的类。其主要设置的参数有损失函数 loss 和对应惩罚项的 penalty 参数，结合网上的一些 SGD 模型训练，确定初步 SGD 模型的参数设置为 loss=”hinge”,alpha=1e-3,max_iter=5,random_state=42。

➢ Logistic Regression 算法

LR 的算法原理为，首先定义计算损失函数，投入训练集进行参数求解和迭代训练，然后用测试集投放到训练好的模型进行预测，归判所属类别。我们调用 python 的 LogisticRegression 函数，其进行多分类的原理是一对一（ovo）的多分类方法，k 个类别的样本就要设计 k(k-1)/2 个分类器。LogisticRegression 函数的参数选择为惩罚项的 penalty 的值设置为”l2”,即假设模型参数满足高斯分布，一定程度上保证模型不会过拟合，另外通过资料将 dual 通常设置为 False。由于数据不会高度失衡，所以暂时不用 class_weight=balanced,根据调试，发现当正则化系数 λ 的倒数 C=4 时，模型的准确率和得分最高.所以最终设定 penalty=”l2” C=4,dual=False 这三个参数。

➢ SVM 算法

SVM 算法首先根据训练集构建优化函数，用 SMO 算法求解分界参数，并将测试机投入，计算分类值。本此采用的是 python 的 sklearn 包的 svm.SVC 函数，其采用的多分类方法为一对一（ovo）的多分类方法，其做法是在任意两类样本之间设计一个 SVM，因此 k 个类别的样本就要设计 k(k-1)/2 个 SVM。当对一个

未知样本进行分类时，最后得票最多的类别即为该未知样本的类别。svm.SVC 函数的参数 C 越大，相当于惩罚松弛变量，希望松弛变量接近 0，即对误分类的惩罚增大，趋向于对训练集全分对的情况，这样对训练集测试时准确率很高，但泛化能力弱。C 值小，对误分类的惩罚减小，允许容错，将他们当成噪声点，泛化能力较强，对于 SVM 来说，惩罚系数 C 是很重要的参数，最终决定当 C=1 时，效果较好。经过测试，核函数选用线性核函数，其他参数保持默认方法，最终得到的预测效果相对较好。

3.1.3 深度学习分类模型

- TextCNN

整个模型由四部分构成：输入层、卷积层、池化层、全连接层。

Step1:输入层：Text-CNN 模型的输入层输入的是句子的分词向量的拼接而成的维度大小为 batch*seq_len 矩阵。

Step2:嵌入层：Text-CNN 模型的输入的矩阵，经过词嵌入层，即将每一个词嵌入到更高维的空间。

Step3:卷积层：在 TextCNN 模型中一般使用多个不同尺寸的卷积核。卷积核的高度，即窗口值，它是一个超参数，需要在任务中尝试，一般选取 2-8 之间的值，本次模型使用的是 2 层卷积核

Step4:池化层：在 Text-CNN 模型的池化层中使用了 Max-pool（最大值池化），即减少了模型参数，又保证了在不定长的卷基层的输出上获得一个定长的全连接层的输入。

Step5:全连接层：全连接层的作用就是分类器

优化调整过程：经过多次的调参优化，获得相对较优的模型，参数，epoc=100，embedding=300，dropout=0.5，lr=0.001，kerne_num=2

- LSTM

LSTM 模型就是在每个小单元中增加了三个 sigmoid 函数，实现门控功能，控制数据的流入流出，分别称为遗忘门（forget gate），输入门（input gate）和输出门（output gate）。

- GRU

GRU 在 LSTM 的基础上主要做出了两点改变 ，GRU 只含有两个门控结构，且在超参数全部调优的情况下，二者性能相当，但是 GRU 结构更为简单，训练样本较少，易实现。

GRU 的更新门的主要功能是决定有多少过去的信息可以继续传递到未来，其得出的数值在[0, 1]之间

GRU 的重置门的主要功能是决定有多少历史信息不能继续传递到下一时刻。其操作于更新门差不多，只是用法不同。

3.2问题二

3.2.1 预处理

本文将留言按时间排序好后，对留言详情进行分词，提取出关键词。因留言主题是留言详情的核心内容，因此在计算留言详情关键词词频时，在留言主题出现过的词记为特征项，对特征项要修改词频权重为 3，非特征项的词频权重为 1，然后计算 idf 值，最后构造词向量模型。

$$tf^*(keyword) = \begin{cases} tf, if\ keyword\ in\ theme \\ 3tf, if\ keyword\ not\ in\ theme \end{cases}$$

3.2.2 Agglomerative Cluster

Agglomerative Cluster 是一种常用的层次聚类算法，其原理是：最初将每个对象看成一个簇，然后将这些簇根据某种规则被一步步合并，就这样不断合并直到达到预设的簇类个数。本文需要将同类事件聚类，因此这里使用相似度作为合并规则。

3.2.3 NER

留言主题已经包含了事件包含的人群地点，因此本文仅对留言主题使用 NER 模型。

本文通过以下步骤利用 Stanford corenlp 的模型训练出符合本题的 Theme-NER 模型：

1) 随机抽取附件 3 小部分留言数据的留言主题，对其标注“LOC”、“PER”、“FAC”和“ORA”（分别代表地点、人物、设施和组织）作为数据集，同时加入了人工选出的原始 NER 模型未识别的街道名；

2) 将数据集中的大部分作为训练集训练 NER 模型，并将小部分作为测试集，检查模型识别效果；

3) 倘若对测试集识别效果不错，再次从附件 3 中随机抽取新的数据（需要注意避开和训练集中相似的实体），使用模型对其进行标注，检查模型对未识别过的实体的识别效果，若对新数据识别效果一般，则手动修改新数据的识别结果，然后将新数据和测试集数据并入训练集，取新的数据进行标注作为新的测试集，再次进行步骤 2；

4) 倘若对测试集识别效果一般，则直接将测试集并入训练集，再取新的数据进行标注作为新的测试集，再次进行步骤 2.

当模型基本能将实体识别出来时停止训练，本文在模型达到准确率 75%时停止训练，准确率不高是因为其中有些实体被误判类别，考虑到后续归类只需要把实体都提取出来，本文就此停止训练。使用 Theme-NER 模型对所有的留言主题进行命名实体识别，并提取出每个留言主题对应的实体在每类中进行归类。

3.2.4 热度评价指标

本文设定的热度得分由留言频率决定，针对某类热点问题的留言频率分为时间频率和数量频率。

1) 时间频率

时间频率是指在该类热点问题出现的时间段内，出现过该类热点问题的天数与该段时间总天数的比值，时间频率越大，则该类问题热度越高。

$$score_{T_i} = \frac{N_{t_i}}{N_i}$$

$score_{T_i}$为第 i 类热点问题的时间频率的热度得分，其中N_i为第 i 类热点问题留言出现过的时间段，N_{t_i}为第 i 类热点留言出现过的天数总和。

2) 数量频率

数量频率是指在该类热点问题出现的时间段内，该类热点问题的留言数量，与该段时间内留言总数的比值,数量频率越大，则该类问题热度越高。

$$score_{C_i} = \frac{\sum_{j=1}^{N_{t_i}} df_{ij} \,/\, df_j}{N_{t_i}}$$

$score_{C_i}$为第 i 类热点问题的数量频率的热度得分，df_{ij}为第 i 类热点问题在第 j 天的留言数量，df_j为第 j 天的总留言数量。

综合上面 2 个指标，我们得出某类热点问题的热度得分为

$$score = score_{T_i} \times score_{C_i}$$

3.3问题三

3.3.1 预处理

本题的数据预处理分为两个步骤。首先，进行数据清洗，对官网提供的附件 2 数据进行空值的删除以及时间格式的统一调整，得到 data4 数据集。其次，用 python 的 jieba 包实现分词以及去除停用词，停用词用问题一所用的停用词，然后进行词性筛选，选择'n', 'nz', 'v', 'vd', 'vn', 'l', 'a', 'd'这些词性留下来。

3.3.2 特征筛选

用 tf-idf 方法获取答复文本 top10 关键词，其具体步骤为：首先，构建词频

矩阵，将文本中的词语转换成词频矩阵；其次，统计每个词的 tf-idf 权值；接着，获取词袋模型中的关键词，然后获取 tf-idf 矩阵，a[i][j]表示 j 词在 i 篇答复文本中的 tf-idf 权重。

3.3.3 答复评价模型

对留言的答复做相关性的评价，要点在于如何衡量留言与答复之间语义上的相关。一般的做法是将留言、答复两个文本进行分词，向量化，使用欧氏距离、余弦相似度或者 jacard 距离来计算两个文本之间的相似性。这些距离计算方法在向量空间上效果显著，但是对于文本语义的相关性，语义除了要在向量空间上的相关，仍需在上下文之间相关。对于这次的相关性评价，我们采用了基于概率检索模型提出的算法，BM25。BM25 是一种用来评价搜索词和文档之间相关性的算法，对于留言答复的相关性评价，有很大的潜力。

BM25 的思想：对 Query 进行语素进行解析，生成语素 qi（其中 Query 每一个分词作为一个语素），然后，对每个搜索结果 D，计算每个语素 qi 与 D 的相关性得分，最后将 qi 相对于 D 的相关性得分进行加权求和，从而得到 Query 与 D 的相关性得分。

本次解题思路基于 BM25 的思想，对每条留言的答复，对相对应的留言文本进行相关性计算，其中，使用全部的留言关键词来 BM25 模型建立文档词库，通过 IDF（Inverse Document Frequency）逆文档率来计算语素与文档 D 的相关性权重。

BM25 的一般公式为：

$$Score(Q,d) = \sum_{i}^{n} Wi \cdot R(qi,d)$$

其中，Q 为 Query，qi 为 Q 解析后的一个语素，d 表示一个搜索文档即答复对应的留言，Wi，R（qi，d）表示 qi 与文档 d 的相关性权重，即语素的 IDF。

最终将得到每条答复评论与原来评论的相似度得分，然后我们设定分数阈值 15，10 来将评论划分为优、良、差，其中优的相关性、完整性以及可解释性最强。

3.3.4 模型评价

模型简单直观，用量化的分数来衡量答复的质量。

模型的不足之处就是用分数来衡量可解释性与完整性的相对较，后期还应该进行更深层次的语义分析。

4 实验结果

4.1问题一

以下是 SGD 模型预测结果：

```
                precision    recall  f1-score   support

        交通运输       0.94      0.65      0.77       182
      劳动和社会保障       0.89      0.96      0.92       616
        卫生计生       0.95      0.87      0.91       243
        商贸旅游       0.92      0.77      0.84       355
        城乡建设       0.81      0.91      0.86       586
        教育文体       0.91      0.92      0.91       481
        环境保护       0.89      0.92      0.90       300

    accuracy                           0.88      2763
   macro avg       0.90      0.86      0.87      2763
weighted avg       0.89      0.88      0.88      2763
```

图 5 SGD 模型预测结果

下面为混淆矩阵

```
[[118   8   1   6  38   5   6]
 [  0 591   7   3   8   6   1]
 [  0  20 211   4   3   4   1]
 [  6  13   1 272  39  16   8]
 [  2  14   2   5 536   9  18]
 [  0  19   1   4  16 441   0]
 [  0   0   0   1  21   3 275]]
```

图 6 SGD 模型混淆矩阵

分析各个类的判别准确数量。该初步 SGD 模型的准确率如下

表 1SGD 模型分类结果

	准确率	回召率	得分
未对数据赋权重	0.90	0.86	0.87
对数据赋权重	0.89	0.88	0.88

下面利用网格搜索对 SGD 算法进行超参数优化，调用 python 的 GridSearchCV 函数，优化参数是将 CountVectorizier()的 n-gram 取值为(1,1)或(1,2)，TfidfTransformer()中的 ues_idf 取值 True 或者 False, SGDClassifier()中的惩罚系数取值为 1e-2,1e-3,1e-4,1e-5 这其中一个，并采用十折交叉验证。得到最优模型如下所示：

```
Best: 0.896540
using {'clf__alpha': 0.0001, 'tfidf__use_idf': True, 'vect__ngram_range': (1, 2)}
```

图 7 SGD 优化模型

其中测试集的预测效果如下

```
                precision    recall  f1-score   support

        交通运输       0.94      0.74      0.83       182
      劳动和社会保障       0.93      0.96      0.94       616
        卫生计生       0.96      0.89      0.93       243
        商贸旅游       0.92      0.85      0.89       355
        城乡建设       0.84      0.93      0.88       586
        教育文体       0.95      0.93      0.94       481
        环境保护       0.91      0.94      0.93       300

    accuracy                           0.91      2763
   macro avg       0.92      0.89      0.90      2763
weighted avg       0.91      0.91      0.91      2763
```

图 8 SGD 优化模型预测结果

```
[[134   3   0   6  34   2   3]
 [  0 591   6   2  12   4   1]
 [  1  14 216   5   3   3   1]
 [  4   5   1 303  24  11   7]
 [  3  11   1   7 544   4  16]
 [  0  14   0   5  15 447   0]
 [  0   0   0   1  14   2 283]]
```

图 9 SGD 优化模型的混淆矩阵

表 2 SGD 优化模型分类效果

	准确率	召回率	F1 得分
未对数据赋权重	0.92	0.89	0.90
对数据赋权重	0.91	0.91	0.91

我们可以看到优化后准确率、回召率以及得分都相应提高，对数据赋权重后的得分提高了 3%。

	precision	recall	f1-score	support
交通运输	0.93	0.69	0.79	182
劳动和社会保障	0.92	0.96	0.94	616
卫生计生	0.97	0.87	0.92	243
商贸旅游	0.88	0.85	0.87	355
城乡建设	0.81	0.93	0.87	586
教育文体	0.95	0.91	0.93	481
环境保护	0.94	0.91	0.92	300
accuracy			0.90	2763
macro avg	0.91	0.87	0.89	2763
weighted avg	0.90	0.90	0.90	2763

图 10 LR 模型预测结果

下面为对应的混淆矩阵

```
[[126   2   0  10  42   1   1]
 [  0 592   5   3  12   4   0]
 [  1  16 212   7   4   3   0]
 [  5   5   1 303  27   9   5]
 [  4  12   1  10 544   3  12]
 [  0  18   0  10  17 436   0]
 [  0   1   0   1  23   2 273]]
```

图 11 LR 模型的混淆矩阵

LR 模型得到模型的各类召回率和总的召回率，运行结果如下表所示：

表 3 LR 模型分类效果

	精准率	回召率	F1 得分
未调整数据权重	0.91	0.87	0.89
调整数据权重	0.90	0.90	0.90

根据以上模型，我们训练的结果如下：

	precision	recall	f1-score	support
交通运输	0.90	0.70	0.79	182
劳动和社会保障	0.92	0.95	0.94	616
卫生计生	0.96	0.88	0.91	243
商贸旅游	0.87	0.85	0.86	355
城乡建设	0.81	0.92	0.87	586
教育文体	0.96	0.92	0.94	481
环境保护	0.93	0.91	0.92	300
accuracy			0.90	2763
macro avg	0.91	0.88	0.89	2763
weighted avg	0.90	0.90	0.90	2763

图 12 SVM 模型的预测结果

```
[[128   2   0   9  43   0   0]
 [  0 583   8   6  16   3   0]
 [  1  18 213   6   3   2   0]
 [  6   4   1 303  26   9   6]
 [  6   8   1  13 541   3  14]
 [  1  15   0   9  13 443   0]
 [  1   1   0   2  22   0 274]]
```

图 13 SVM 模型的混淆矩阵

表 4 SVM 模型的分类效果

	准确率	回召率	得分
未调整数据权重	0.91	0.88	0.89
调整数据权重	0.90	0.90	0.90

模型评价：

由于本题的数据有 9210 条，数据量相对较少，所以三个传统机器学习分类模型的效果都比较好，得分基本达到 90%。

但是 LR 算法对于参数的计算采用精确解析的方式，对于小样本数据来说模型好，但是对于大量数据来说存在计算时间长的问题，这种方法的问题难以扩展到数量巨大的样本上，SVM 散发对大样本的适用性也不好。SGDClassifier 采用随机梯度下降算法估计模型参数，计算时间短但是产出模型性能略低。SGD 的优点是高效和容易实现，而缺点也很明显：SGD 需要许多超参数：比如正则项参数、迭代数；同时 SGD 对于特征归一化（feature scaling）是敏感的。

表 5 textcnn 参数表

参数	参数说明	数值
embedding_dim	词嵌入的空间维度	300
dropout	在训练模型时刻意隐藏部分网络节点，以达到更好的训练效果	0.5
class_num	类别数，即分类结果	7
epoch	训练次数	100
lr	模型学习率	0.001
vocab_size	词袋的大小	33602
kernel_num	卷积核高度	2

LSTM：

首先，采用 200 维的 embed 层，经过多次调整，发现正态分布随机初始化 embed 层的 weights 较好。

其次，建立三层 lstm 双向传播层，设置 dropout 为 0.5，后接一层全连接层，并正态分布随机初始化 fc 层的 weights，0 填充 bias

最后在连接层后再加一层 log_softmax 来平滑数据，可考虑在 lstm 层与全连接层之间，填加一层 NORM 层来规范化数据，防止过拟合。

表 6 LSTM **参数表**

参数	参数说明	训练结果数值
lr	学习率	0.01
epoch	训练次数	100
dropout	训练时隐藏部分节点	0.5
embedding_dim	词嵌入空间维度	200
hidden_size	隐藏层状态的维度	128
lstm_layers	RNN 模型层数	2
class_num	分类数目	7
bidirectional	是否采用双向网络	True

表 7 GRU **参数表**

参数	参数说明	训练结果数值
lr	学习率	0.01
epoch	训练次数	100
dropout	训练时隐藏部分节点	0.5
embedding_dim	词嵌入空间维度	300
hidden_size	隐藏层状态的维度	128
num_layers	RNN 模型层数	2
class_num	分类数目	7
bidirectional	是否采用双向网络	True

所有分类模型对比：

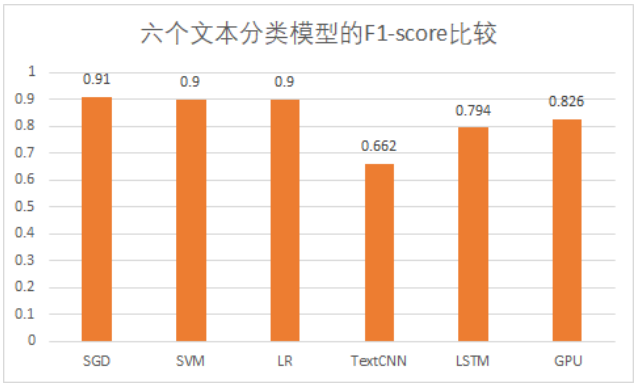

图 14 **模型对比**

4. 2问题二

表 8 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	0.069919	2019/07/07 至 2019/09/01	A 市伊景园滨河苑	投诉 A 市伊景园滨河苑定向限价商品房违规涨价
2	2	0.029578	2019/08/12 至 2019/08/26	A 市经开区泉星公园	对 A 市经开区泉星公园项目规划再进一步优化的建议
3	3	0.027027	2019/03/25 至 2019/05/01	A7 县星沙中贸城	A7 县星沙中贸城欺诈业主、拖欠业主资金不退还
4	4	0.022599	2019/11/15 至 2019/12/26	A 市丽发小区建搅拌站	A 市丽发小区建搅拌站，噪音污染严重
5	5	0.011728	2019/08/18 至 2019/12/04	A 市魅力之城商铺	A 市魅力之城商铺无排烟管道，小区内到处油烟味

热点问题明细表请见作品附件。

5 参考文献

[1]朱正. 政府通告文本分类系统的设计与实现[D].东南大学,2018.
[2]文本表示之词袋模型 https://zhuanlan.zhihu.com/p/53302305
[3]机器学习---SGDClassifier 梯度下降分类方法
https://blog.csdn.net/WxyangID/article/details/80365779
[1]迟呈英,李红.基于改进 TF* PDF 算法的网络新闻热点话题检测和跟踪[J].计算机应用与软件,2013,30(12):311-314.
[2]刘星星,何婷婷,龚海军,陈龙.网络热点事件发现系统的设计[J].中文信息学报,2008,22(06):80-85.
[3] How Not To Sort By Average Rating https://www.evanmiller.org/how-not-to-sort-by-average-rating.html